

CHAPTER 1

INTRODUCTION

1.1 ECONOMIC LOAD DISPATCH

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.

To solve the economic load dispatch problem many techniques were proposed such as classical techniques, linear programming (LP), nonlinear programming (NLP), Quadratic Programming (QP), swarm optimization, evolutionary programming, genetic algorithm, etc.

The main aim of electric power utilities is to provide high-quality, reliable power supply to the consumers at the lowest possible cost while operating to meet the limits and constraints imposed on the generating units. This formulates the economic load dispatch (ELD) problem for finding the optimal combination of the output power of all the online generating units that minimizes the total fuel cost, while satisfying an equality constraint and a set of inequality constraints. As the cost of power generation is exorbitant, an optimum dispatch results in economy.

In recent years, with an increasing awareness of the environmental pollution caused by thermal power plants, limiting the emission of pollutants is becoming a crucial issue in economic power dispatch. The conventional economic power dispatch cannot meet the environmental protection requirements, since it only considers minimizing the total fuel cost. The multi-objective generation dispatch in electric power systems treats economic and emission impact as competing objectives, which requires some reasonable trade off among objectives to reach an optimal solution. This formulates the combined economic emission dispatch (CEED) problem with an objective to dispatch the electric power considering both economic and environmental concerns.

1.2 COMBINED ECONOMIC EMISSION DISPATCH:

The basic objective of economic dispatch (ED) of electric power generation is to schedule the committed generating unit outputs so as to meet the load demand at minimum operating cost while satisfying all unit and system equality and inequality constraints. This makes the ED problem a large-scale highly nonlinear constrained optimization problem. In addition, the increasing public awareness of the environmental protection and the passage of the Clean Air Act Amendments of 1990 have forced the utilities to modify their design or operational strategies to reduce pollution and atmospheric emissions of the thermal power plants. Fossil fuel based power generators are blamed for being a major contributor to air pollution. Their primary gaseous pollutants are carbon oxides (CO_x), sulfur oxides (SO_x) and nitrogen oxides (NO_x). Several strategies to reduce the atmospheric emissions have been proposed and discussed. These include installation of pollutant cleaning equipment such as gas scrubbers and electrostatic precipitators, switching to low emission fuels, replacement of the aged fuel-burners and generator units with cleaner and more efficient ones, and emission dispatching. The first three options require installation of new equipment and/or modification of the existing ones that involve considerable capital outlay and, hence, they can be considered as long-term options. The emission dispatching option is an attractive short-term alternative in which the emission in addition to the fuel cost objective are to be minimized. Thus, the ED problem can be handled as a multi-objective optimization problem with non-commensurable and contradictory objectives. In recent years, this option has received much attention since it requires only small modification of the basic ED to include emissions.

For that work, CEED which is a multi-objective problem has begun on the distribution of power. Several researchers have considered emissions with fuel cost in the objective function. They converted the multi-objective CEED problem into a single objective function using a price penalty factor. In the traditional economic dispatch problem, the cost function for each generator has been approximately represented by a single quadratic function. Unfortunately, the characteristics of generating units are highly nonlinear inherently, because of the constraints power system and emission. A number of methods have been presented to solve CEED problems such as neural networks, Fuzzy logic, evolutionary computation methods, recursive method, γ -iteration method, particle swarm optimization, differential evolution, simplified recursive method and genetic algorithms. Genetic algorithm is one of the modern

heuristic algorithms, which can be used to solve nonlinear and non-continuous optimization problems. In genetic algorithms method, the genetic operators such as crossover and mutation have significant impact on its performance. Michalewicz described a crossover method which is called arithmetic in 1994. An arithmetic crossover operator linearly combines two parent chromosome vectors to produce two new children. The main drawback of this method is the use of random crossover constant. This coefficient can be obtained from the similarity of parents chromosomes.

Economic load dispatch (ELD) is a method to schedule power generators' output with respect to the load demands and to operate a power system economically, so as to minimize the operation cost of the power system. The input-output characteristics of modern generators are nonlinear by nature because of the valve-point loadings and rate limits. Thus the characteristics of ELD problems are multimodal, discontinuous and highly nonlinear. The Combined economic-emission dispatch (CEED) problem which accounts for minimization of both cost and emission is a multiple, conflicting objective function problem. In this dissertation 'Economic-Emission Load Dispatch using Optimization Tool' solved the economic-emission load dispatch problem through genetic algorithms in MATLAB. The application and validity of the proposed method are demonstrated for a sample system having six generators and eleven generators. In reality, the CEED problem is a multiple objective problem with conflicting objectives because minimum pollution is conflicting with minimum cost of generation.

1.3 LITERATURE SURVEY

Different techniques have been reported in the literature pertaining to environmental/economic dispatch (EED) problem. The problem has been reduced to a single objective problem by treating the emission as a constraint with a permissible limit. This formulation, however, has a severe difficulty in getting the trade-off relations between cost and emission. Alternatively, minimizing the emission has been handled as another objective in addition to usual cost objective. A linear programming based optimization procedures in which the objectives are considered one at a time was presented in Ref. [37]. Unfortunately, the EED problem is a highly nonlinear and a multimodal optimization problem. Therefore, conventional optimization methods that make use of derivatives and gradients, in general, not able to locate or identify the global optimum. On the other hand, many mathematical assumptions such as analytic and differential objective functions

have to be given to simplify the problem. Furthermore, this approach does not give any information regarding the trade-offs involved.

In other research direction, the multi-objective EED problem was converted to a single objective problem by linear combination of different objectives as a weighted sum. The important aspect of this weighted sum method is that a set of non-inferior (or Pareto-optimal) solutions [33] can be obtained by varying the weights. Unfortunately, this requires multiple runs as many times as the number of desired Pareto-optimal solutions. Furthermore, this method cannot be used to find Pareto-optimal solutions in problems having a non-convex Pareto-optimal front. In addition, there is no rational basis of determining adequate weights and the objective function so formed may lose significance due to combining non-commensurable objectives. The most obvious weaknesses of this approach are that it is time-consuming and tends to find weakly non-dominated solutions.

Goal programming method [34] was also proposed for multi-objective EED problem. In this method, a target or a goal to be achieved for each objective is assigned and the objective function will then try to minimize the distance from the targets to the objectives. Although the method is computationally efficient, it will yield an inferior solution rather than a non-inferior one if the goal point is chosen in the feasible domain. Hence, the main drawback of this method is that it requires a prior knowledge about the shape of the problem search space.

The recent direction is to handle both objectives simultaneously as competing objectives instead of simplifying the multi-objective problem to a single objective problem. A fuzzy multi-objective optimization technique for EED problem was proposed [30]. However, the solutions produced are sub-optimal and the algorithm does not provide a systematic framework for directing the search towards Pareto-optimal front. An evolutionary algorithm based approach evaluating the economic impacts of environmental dispatching and fuel switching was presented in Ref. [13,28]. The important aspect of this approach is that it produces several alternatives along the Pareto-optimal front. However, some of non-dominated solutions may be lost during the search process while some of dominated solutions may be misclassified as non-dominated ones due to the selection process adopted. In addition, no effort has been done to prevent the algorithm from its bias towards some regions. A multi-objective stochastic search technique for the multi-objective EED problem was presented in Ref. [32]. This technique hybridizes genetic algorithms (GA) and simulated annealing in the sense that the selection process of GA is

enhanced by local heuristic search for better search capabilities. However, the technique is computationally involved and time-consuming.

The classical lambda iteration method has been used to solve the ELD problem. This method utilizes an equal incremental cost criterion for systems without transmission losses and the penalty factors using B-matrix for systems with transmission losses. Other methods such as gradient, Newton, linear programming and interior point have also been applied to solve the ELD problems.

Zwe-Lee Gaing [35] has proposed a particle swarm optimization (PSO) method for solving the economic dispatch (ED) problem in power systems. This method made use of PSO for its global search capability to allocate optimum loading of each generator. The test results of three different systems have been compared with that of GA-based approach.

Gent and Lamont have started the early work on minimum emission dispatch [22]. Optimal power-dispatch problem, considering practical constraints such as transformer taps, voltage, and line-flow constraints has been solved by Fletcher's quadratic programming method [23]. Nanda, Hari , and Kothari made an attempt to explore the feasibility of developing a classical technique based on coordination equations to solve economic emission load dispatch with line flow constraints [24]. Chen and Chen proposed a fast Newton-Raphson method for a multi-objective power dispatch problem with line flow constraints [25].

P. K. Hota, R. Chakrabarti, and P. K. Chattopadhyay proposed a sequential quadratic programming technique to solve CEED problem by assigning weighting factors for generation and emission cost functions [26]. A number of methods have been presented to solve CEED problems such as Hopfield neural networks [5], Fuzzy logic [30], evolutionary computation methods [13], γ -iteration method, differential evolution [36] and simplified recursive method [1].

Talaq et al. [31] have formulated an optimal power flow problem with emission constraints where the main objective was to minimize the fuel cost and the total emission over a wide time period of different intervals and system demands. The test results of standard 5-bus and IEEE-30 bus systems display a trade-off relationship between fuel cost and emission.

In Song et al. [30] Fuzzy logic controlled genetic algorithm is used to solve CEED problem Wong et al. [28] have developed an efficient and reliable evolutionary programming based algorithm for solving the environmentally constrained economic dispatch (ECED) problem. This

method made use of acceleration techniques in order to enhance the speed and robustness of the algorithm. Venkatesh et al. [13] have built an EP algorithm to solve the CEED problem with line flow constraints. The line flows in MVA have been computed directly from the Newton-Raphson method. A novel modified price penalty factor has been introduced to find the exact economic emission fuel cost with respect to the load demand. The test results of IEEE-14, 30 and -118 bus systems have been compared with that of other evolutionary computing techniques.

Abido [33] has derived a Pareto-based multi-objective evolutionary algorithm (MOEA) for solving an environmental/economic electric power dispatch problem. This fuzzy-based hierarchical clustering technique has been implemented in order to obtain the best solution. The test results of an IEEE-30 bus system have been compared with that of other traditional multi-objective optimization techniques.

In the very recent years, researchers have been showing a great deal of attention to the development of efficient, real multi-objective optimization techniques for solving various real world multi-objective problems. A fuzzy multi-objective optimization technique for CEED problem was proposed [30]. A new Strength Pareto Evolutionary Algorithm (SPEA) based approach is proposed to solve CEED problem [33]. In this SPEA method, Pareto optimal solutions are found in a single run. The comparative study proves that the SPEA is highly capable of finding Pareto optimal solutions, with diversity in a single run, over the other method like Niche Pareto GA (NPGA).

D.C. Walters and G.B. Sheble [38] have developed a genetic algorithm to solve the economic dispatch problem with valve-point effects. This algorithm has utilized payoff information of the candidate solutions to evaluate their optimality. The test results of three units system have been compared with that of dynamic programming method. Wong et al. [39] have built an incremental genetic algorithm based approach for the determination of global or near-global optimum solution. Another technique that incorporates both incremental genetic theory and simulated annealing has served to determine the economic loadings of 13 generators in a practical power system with the effects of valve-point loading and ramping characteristics. The test results have been found to yield better results when compared with that of simulated annealing based method. Chen et al. [25] have presented a GA-based method that uses the incremental cost of encoded parameter of the system for solving the ED problem taking into account the network losses, ramp

rate limits, valve-point zone and prohibited operating zone. The numerical results of the method for a large scale 40-unit system have been compared with that of lambda-iteration method.

Fung et al. [40] have formulated an integrated parallel genetic algorithm incorporating Tabu search (TS) and simulated annealing for solving the ED problem. The parallel computing platform has been based on a network of interconnected personal computers (PCs) using TCP/IP socket communication facilities. The test results of a practical power system have been obtained to compute the optimal loading of 13 generators.

Lai et al. [41] have applied PSO to solve economic dispatch (ED) of units with non-smooth input-output characteristic functions. The test results of an IEEE-30 bus system with six generating units have been compared with that of evolutionary programming (EP).

1.4 Objective and Methodology

Economic load dispatch (ELD) and economic emission dispatch (EED) have been applied to obtain optimal fuel cost and optimal emission of generating units respectively. Combined economic emission dispatch (CEED) problem is obtained by considering both the economy and emission objectives. This bi-objective CEED problem is converted into a single objective function using a price penalty factor approach. A novel modified price penalty factor is proposed to solve the CEED problem.

Combined Economic emission dispatch (CEED) problem is to schedule the committed generating unit's outputs to meet the required load demand at minimum operating cost with minimum emission simultaneously. Genetic Algorithm method uses similarity crossover for solving CEED problem in power systems. In this method, children created by using similarity measurement between mother and father chromosomes relationship. The study results show that the proposed approach is more efficient in finding higher quality solutions in CEED problems.

The main goal of electric vehicles at the lowest possible cost to consumers is to provide a reliable source. One should bear in mind that the power plant which is usually based on the lowest fuel costs and environmental pollution is not taken into consideration. But in recent years, electric industry and its contribution to environmental pollution, environmental protection and pollution caused by power plants to reduce or eliminate electric industry raises questions about it. Therefore, emissions from electric power distribution should be kept in mind. For that work, CEED which is a multi-objective problem has begun on the distribution of power. Several

researchers have considered emissions with fuel cost in the objective function. They converted the multi-objective CEED problem into a single objective function using a price penalty factor. In the traditional economic dispatch problem, the cost function for each generator has been approximately represented by a single quadratic function. Unfortunately, the characteristics of generating units are highly nonlinear inherently, because of the constraints power system and emission . A number of methods have been presented to solve CEED problems such as neural networks , Fuzzy logic , evolutionary computation methods , recursive method , γ -iteration method, particle swarm optimization, differential evolution, simplified recursive method and genetic algorithms . Genetic algorithm is one of the modern heuristic algorithms, which can be used to solve nonlinear and non-continuous optimization problems. In genetic algorithms method, the genetic operators such as crossover and mutation have significant impact on its performance . Michalewicz described a crossover method which is called arithmetic in 1994. An arithmetic crossover operator linearly combines two parent chromosome vectors to produce two new children. The main drawback of this method is the use of random crossover constant. This coefficient can be obtained from the similarity of parents chromosomes.

Similarity is fundamentally important in almost every scientific field and makes a vital mission for all concepts of formation. There is mainly an important issue in this regard: How to measure the similarity between pairs of data points [4]. In this research work, a novel GA based on solving CEED problem is proposed. Firstly, children created by using similarity measurement between parents chromosomes relationship. The study results show that the proposed approach is more efficient in finding higher quality solutions in CEED problem.

Chapter 2

MULTIOBJECTIVE OPTIMIZATION

2.1 INTRODUCTION

The optimization Process consists of three basic components: an objective function, variables, and constraints. It finds the value of the variables that minimize or maximizes the objective function while satisfying the constraints. The problem relies on many variables and therefore various combinations of values of the variables have to be explored to obtain the optimized objective function. Conflicting criteria such as cost, capacity performance and reliability are to be considered simultaneously and most suitable one is selected. This is also called a multi objective optimization problem (MOOP).

If a multi-objective problem is well formed, there should not be a single solution that simultaneously minimizes each objective to its fullest. In each case we are looking for a solution for which each objective has been optimized to the extent that if we try to optimize it any further, then the other objective(s) will suffer as a result. Finding such a solution, and quantifying how much better this solution is compared to other such solutions (there will generally be many) is the goal when setting up and solving multi-objective optimization problem.

Optimal economic dispatch in electric power systems has gained increasing importance as the cost associated with generation and transmission of electric energy keeps on increasing. The procedure involves the allocation of total generation requirements among the available generating units in the system in such a manner that the constraint imposed on different system variables are adequately satisfied and the achieved overall cost associated with it is a minimum.

2.2 Formulation of multi-objective problem

In mathematical terms, the multi-objective problem can be written as:

Minimize:

$$Z = [Z_1(x), Z_2(x), \dots, Z_n(x)]^T$$

S.t.

$$g(x) \leq 0$$

$$x \geq 0$$

$$h(x) = 0$$

Where Z is the multi objective function, Z_1, Z_2, \dots, Z_n are the individual objective functions. g and h are the inequality and equality constraints, respectively, and x is the vector of optimization or decision variables.

Multi-objective Optimal power dispatch (MOPD) studies have been carried out on six generators and eleven generator system in 2D space. The data of six generators and eleven generator systems is given in chapter-5. In 2D space, two objectives i.e. cost of generation and system emission outputs are considered.

The ideal situation where one would like to operate the power systems is one where all the objectives i.e. cost of generation and system emission output are minimum. Such a point is called the ideal point. Therefore, while considering multi-objective optimal power dispatch problem, a strategy has to be adopted by the power systems analyst or operator to achieve optimum values as per his satisfaction level and requirements. The operating point so obtained is called the Target Point (TP) or the best compromise solution.

2.3 Weighting Method

The weighting method identify the non-inferior set, in which the best compromise solution lies, also known as the parametric approach, has been the most common method used for solving multi-objective problems until recently. Multi-objective problem is converted in this method into scalar optimization as given below:

$$\text{Minimize : } \quad \sum_{i=1}^G W_i f_i(x)$$

$$\text{Subject to } \quad x \in X$$

Where W_i is the weighting coefficients.

G is the total number of objectives.

The approach yields meaningful results to the decision maker only when solved many times for different values of W_i ($i=1,2,\dots,G$). Though very little is usually known about the values of

weighting coefficient, the DM (Decision Maker) still choose them, presumably on the basis of his institution. The weighting coefficients do not reflect proportionally the relative importance of the objectives but are only factors which, when varied, locate points in the non-inferior set.

2.4 Advantages of Multi-objective Planning

The consideration of many objectives in the planning process accomplishes three major improvements in the problem solving:

- (i) Multi-objective programming and planning promotes more appropriate roles for the participants in the planning and decision making process.
- (ii) A wider range of alternatives is usually identified.
- (iii) The power system analyst's perception of a problem will be more realistic if many objectives are considered.

There are two parts of multi-objective decision making process; analysis and decision making. Analysis of a problem provides information about the problem for making decisions. Multi-objective approaches pursue an important decision making process: an explicit consideration of the relative impacts of the different objectives on the problem. These approaches emphasize the range of choice associated with a decision problem. The responsibility of the assigning relative values to the objectives remains with the decision maker. The beauty of multi-objective approaches is that these provide sufficient information and facilitate the decision making process.

Regardless of the actual nature of decision making process, multi-objective approaches can be useful in promoting the explicit consideration of the value judgments which are implicitly made in the application of single objective approaches.

Multi-objective techniques are used to generate and evaluate more than one alternative. These techniques indicate to decision makers a range of choices beyond one optimal alternative identified by single objective method. A general rule for decision making which is assumed is that more information carefully presented is better than less information. The decision to accept or reject a single optimal alternative is an uninformed decision. Informed decision making requires knowledge of full range of possibilities provided by multi-objective analysis. Multi-objective analysis allows several non-commensurable effects to be treated without artificially combining them.

Chapter -3

GENETIC ALGORITHM

3.1 INTRODUCTION

Genetic Algorithms are a way of solving problems by mimicking the same processes Mother Nature uses. They use the same combination of selection, recombination and mutation to evolve a solution to a problem. Genetic algorithms are one of the best ways to solve a problem for which little is known. They are a very general algorithm and so will work well in any search space and will be able to create a high quality solution. Genetic algorithms use the principles of selection and evolution to produce several solutions to a given problem.

Genetic algorithms tend to thrive in an environment in which there is a very large set of candidate solutions and in which the search space is uneven and has many hills and valleys.

A genetic algorithm (GA) is a search technique used in computing to find exact or approximate solutions to optimization and search problems. Genetic algorithms are categorized as global search heuristics. GA has been developed by John Holland, his colleagues and his students at the University of Michigan. The goals of their search have been two fold. i.e.

1. To abstract and rigorously explain the adaptive processes of natural systems.
2. To design artificial systems software that retains the important mechanisms of natural systems.

Genetic algorithms are a particular class of evolutionary algorithms (EA) that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover. Genetic algorithm should be used in case when alternate solutions are too slow or overly complicated.

A GA is heuristic, which means it estimates a solution which could be the exact solution, but that may be a minor concern. In fact, most real-life problems are like that: you estimate a solution rather than calculating it exactly.

For most problems there is no such formula for solving the problem because it is too complex, or if done, it just takes too long to calculate the solution exactly. An example could be space optimization - it is very difficult to find the best way to put objects of varying size into a room so they take as little space as possible. The most feasible approach then is to use a heuristic method.

Genetic algorithms are different from other heuristic methods in several ways. The most important difference is that a GA works on a population of possible solutions, while other heuristic methods use a single solution in their iterations. Another difference is that GAs are probabilistic (stochastic), not deterministic.

Each individual in the GA population represents a possible solution to the problem. The suggested solution is coded into the "genes" of the individual. One individual might have these genes: "1100101011", another has these: "0101110001". The values (0 or 1) and their position in the "gene string" tell the genetic algorithm what solution the individual represents.

Then the rules of evolution to the individuals are applied. Finding the individuals which sense to be the best suggestions to the problem and then combine these individuals into new individuals. Using this method repeatedly, the population will hopefully evolve good solutions.

A genetic algorithm (GA) is a search technique used in computing to find exact or approximate solutions to optimization and search problems. Genetic algorithms are categorized as global search heuristics. Genetic algorithms are a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover (also called recombination). Zalzal et al.[43] made the review on the current development techniques in genetic algorithm. It explains theoretical aspects of genetic algorithms and genetic algorithm applications. Theoretical topics under review include genetic algorithm techniques, genetic operator technique, niching techniques, genetic drift, and method of benchmarking genetic algorithm performances, measurement of difficulty level of a test-bed function, population genetics and developmental mechanism in genetic algorithms. According to Zalzal et al.[43] there were two types of genetic algorithm earlier one was Breeder genetic algorithm and the other was simple genetic algorithm. Breeder Genetic Algorithm (BGA) is first introduced by Mühleibein and Schlierkamp-Voosen [44]. The major difference between simple genetic algorithm and BGA is the method of selection. Generally, truncation selection is used in BGA. Genetic drift is an important phenomenon in genetic algorithm search. Once the algorithm is converged, the size of original gene pool is reduced to the size of found solution(s) gene pool. This leads to genetic drift. Two niching techniques - simple sub-population scheme and deterministic crowding are being reviewed. Many traditional optimization algorithms suffer from myopia for highly complex search spaces, leading them to less than desirable performance (both in terms of execution speed and fraction of time they need to find an optimal solution) . This

paper helps us in understanding application of genetic algorithm on multiple fault diagnosis problems. It is seen that in a MFD problem, there are regions of the search space and there is little information to direct the search (e.g., in a flat valley). Consequently, local search algorithms may exhibit less than desirable performance. To handle irregular search spaces, such heuristics should adopt a global strategy and rely heavily on intelligent randomization. Genetic algorithms follow just such a strategy. Following the model of evolution, they establish a population of individuals, where each individual corresponds to a point in the search space. An objective function is applied to each individual to rate their fitness. Using well conceived operators, a next generation is formed based upon the survival of the fittest. Therefore, the evolution of individuals from generation to generation tends to result in fitter individuals, solutions, in the search space.

3.2 GENETIC ALGORITHMS AND TRADITIONAL SEARCH METHODS

A population of points is used for starting the procedure instead of a single design point. If the number of design variables is n , usually the size of the population is taken as $2n$ to $4n$. Since several points are used as candidate solutions, GAs is less likely to get trapped at a local optimum. GAs uses only the value of objective function. The derivatives are not used in search procedure.

In GAs the design variables are represented as strings of binary variables that correspond to the chromosomes in natural genetics. Thus the search method is naturally applicable for solving discrete and integer programming problems. For continuous design variables, the string length can be varied to achieve any desired resolution.

The objective function value corresponding to a design vector plays the role of fitness in natural genetics. In every new generation, a new set of strings is produced by using randomized parents selection and crossover from the old generation (old set of strings). Although randomized, GAs is not simple random search techniques. They efficiently explore the new combinations with the available knowledge to find a new generation with better fitness or objective function value.

3.3 EVOLUTIONARY COMPUTATION

The GA is a stochastic global search method that mimics the metaphor of natural biological evolution. GAs operates on a population of potential solutions applying the principle of survival of the fittest to produce (hopefully) better and better approximations to a solution. At each generation, a new set of approximations is created by the process of selecting individuals according to their level of fitness in the problem domain and breeding them together using operators borrowed from natural genetics. This process leads to the evolution of populations of individuals which are better suited to their environment than the individuals that they were created from.

The most common type of genetic algorithm works like this: a population is created with a group of individuals created randomly. The individuals in the population are then evaluated. The evaluation function is provided by the programmer and gives the individuals a score based on how well they perform at the given task. Two individuals are then selected based on their fitness, the higher the fitness, the higher and the chance of being selected. These individuals then "reproduce" to create one or more offspring, after which the offspring are mutated randomly. This continues until a suitable solution has been found or a certain number of generations have passed, depending on the needs of the programmer.

The genetic algorithm uses three main types of rules at each step to create the next generation from the current population:

Selection rules select the individuals, called parents that contribute to the population at the next generation.

Crossover rules combine two parents to form children for the next generation.

Mutation rules apply random changes to individual parents to form children.

Following are the steps of genetic algorithm:

1. Produce an initial population of individuals.
2. Evaluate the fitness of all the individuals.
3. Determine if the solution is found; if yes go to step 8.
4. Select fitter individuals for reproduction.
5. Produce new Individuals by crossover and mutation.
6. Evaluate fitness of new individuals.

7. Generate a new population by choosing some new “fit” Individuals.
8. Stop.

This flowchart illustrates the basic steps GA:

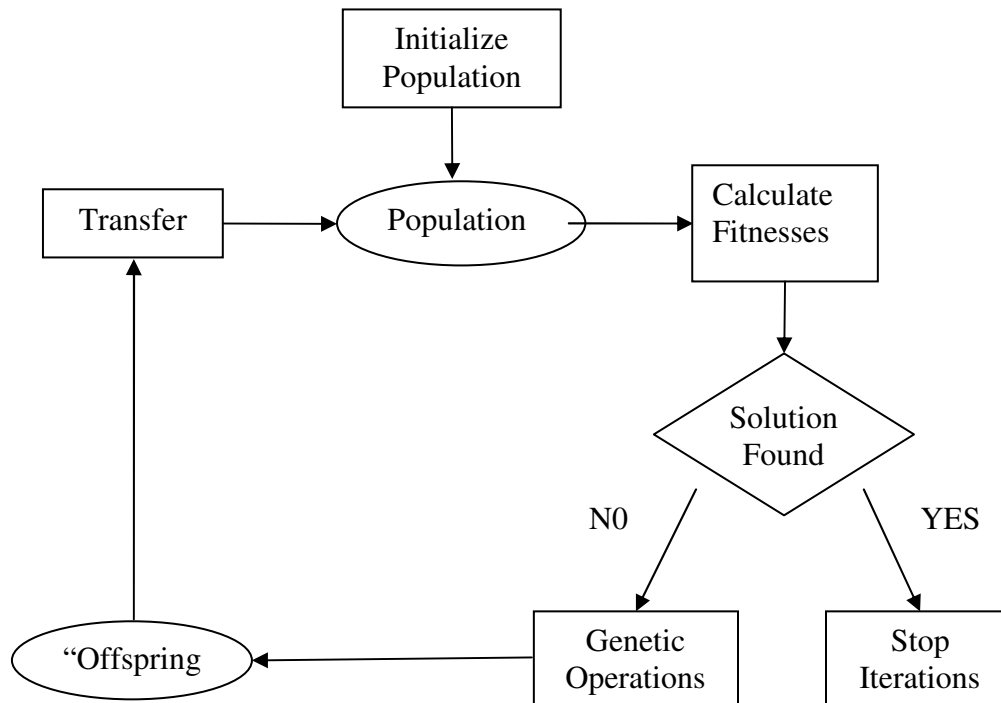


Fig: 3.1 Flow chart of Genetic algorithm

3.4 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, Multi point 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 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 in Example-1.

Example-1 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 shown in Example-2. In a multipoint crossover, again there are two cases. One is even no. of cross sites and other is odd no of sites. For even no. of sites the string is treated as a string and cross sites are selected around the circle uniformly at random.

Example-2 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: $X_2 = \{ 1 0 0 1 1 0 1 1 0 0 \}$

Sites are selected around the circle uniformly at random if the number of cross sites is odd, then a different cross point is always assumed at the string beginning.

3.5 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 GA's 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.

Example-3 Mutation operation

Offspring X_1 : 1 1 1 1 0 1 0

New offspring X_2 : 1 1 0 1 0 1 0

3.6 Advantages of GA

Advantages of GA's are given below as discussed in [5, 27].

- Simple to understand and to implement, and early give a good near solution
- Optimizes with continuous or discrete variables.
- Doesn't require derivative information.
- Simultaneously searches from a wide sampling of the cost surface.
- Deals with a large number of variables.
- Is well suited for parallel computers.
- Optimizes variables with extremely complex cost surfaces (they can jump out of a local minimum).
- Provides a list of optimum variables, not just a single solution.
- Can encode the variables so that the optimization is done with the encoded variables.
- Works on a wide range of problems.
- For each problem of optimization in GAs, there are number of possible encodings.

These advantages are intriguing and produce stunning results where traditional optimization approaches fail miserably. Due to various advantages as discussed above, GA is used for a number of different application areas. In power system, the GA has been used in following areas:

- Loss reduction using Active Filter
- Power system restoration planning
- Controllers
- Optimal load dispatch
- Voltage stability

3.7 Disadvantages of GA

In spite of its successful implementation, GA does possess some weaknesses leading to

- Longer computation time.
- Less guaranteed convergence, particularly in case of epistemic objective function containing highly correlated parameters.
- Premature convergence of GA is accompanied by a very high probability of entrapment into the local optimum [9].
- GAs tends to fail with the more difficult problems and need good problem knowledge to be tuned.
- Need much more function evaluations than linearized methods.
- No guaranteed convergence even to local minimum.

Chapter 4

OPTIMIZATION USING GENETIC ALGORITHM

4.1 INTRODUCTION

Being a population-based approach, GA is well suited to solve multi-objective optimization problems. A generic single objective GA can be modified to find a set of multiple non-dominated solutions in a single run. The ability of GA to simultaneously search different regions of a solution space makes it possible to find a diverse set of solutions for difficult problems with non-convex, discontinuous, and multi-modal solutions spaces. The crossover operator of GA may exploit structures of good solutions with respect to different objectives to create new non dominated solutions in unexplored parts of the Pareto front. In addition, most multi-objective GA do not require the user to prioritize, scale, or weight objectives. Therefore, GA has been the most popular heuristic approach to multi-objective design and optimization problems.

4.2 TOOL (GA TOOL) IN MATLAB

Genetic algorithm software extends the optimization capabilities in MATLAB optimization toolbox. GA tool use these algorithms for problems that are difficult to solve with traditional optimization techniques, including problems that are not well defined or are difficult to model. GA is also used when computation of the objective function is discontinuous, highly nonlinear, stochastic, or has unreliable or undefined derivatives.

The Genetic Algorithm Toolbox is closely integrated with MATLAB R2008b and the Optimization toolbox. We can use the genetic algorithm and pattern search to find adept Starting points and then use the Optimization Toolbox solvers or MATLAB R2008b routines to further refine optimization. Solvers are available for both constrained and unconstrained optimization problems.

GA Toolbox complements other optimization methods, helps to find best fitness value and minimum point of the objective function. GA tool varies on various optional parameters like

population, selection, fitness scaling, crossover, mutation, stopping criteria, plot function and output function, display to command window for finding the best fitness value.

It's important to understand that the functioning of such an algorithm does not guarantee success. It has been shown that the genetic algorithm finds the best fitness.

4.3 EXPLORING THE TOOLS OF GENETIC ALGORITHM IN MATLAB

R2008B

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

4.3.1 Calling the Function at the Command Line

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

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

Where

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

`nvars` is the number of independent variables for the fitness function.

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

The output results are given by

`x` — Point at which the final value is attained

`fval` — Final value of the fitness function at `x`

Return results directly to the MATLAB® workspace.

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

4.3.2 Running the Problem in GA TOOL

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

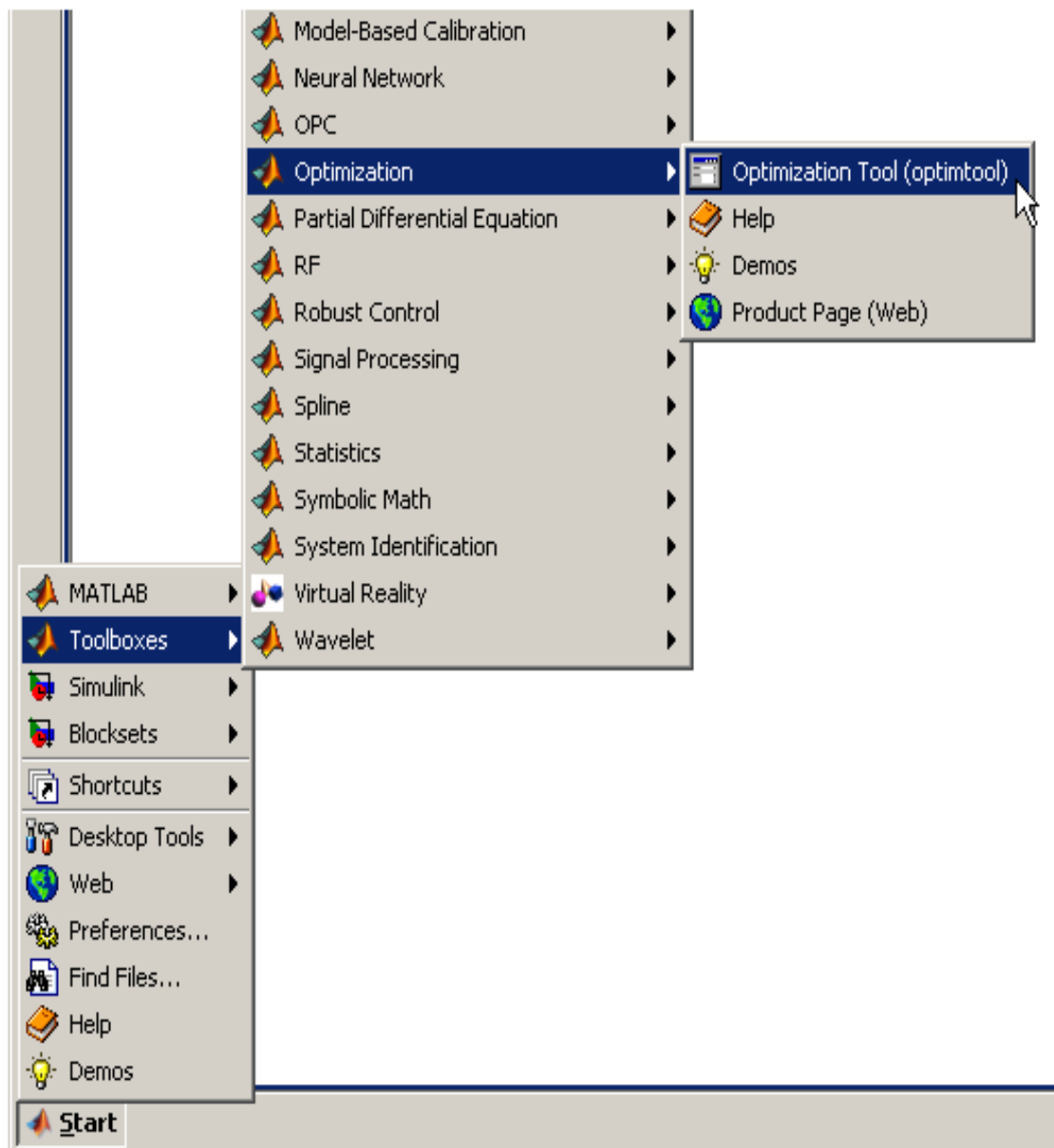
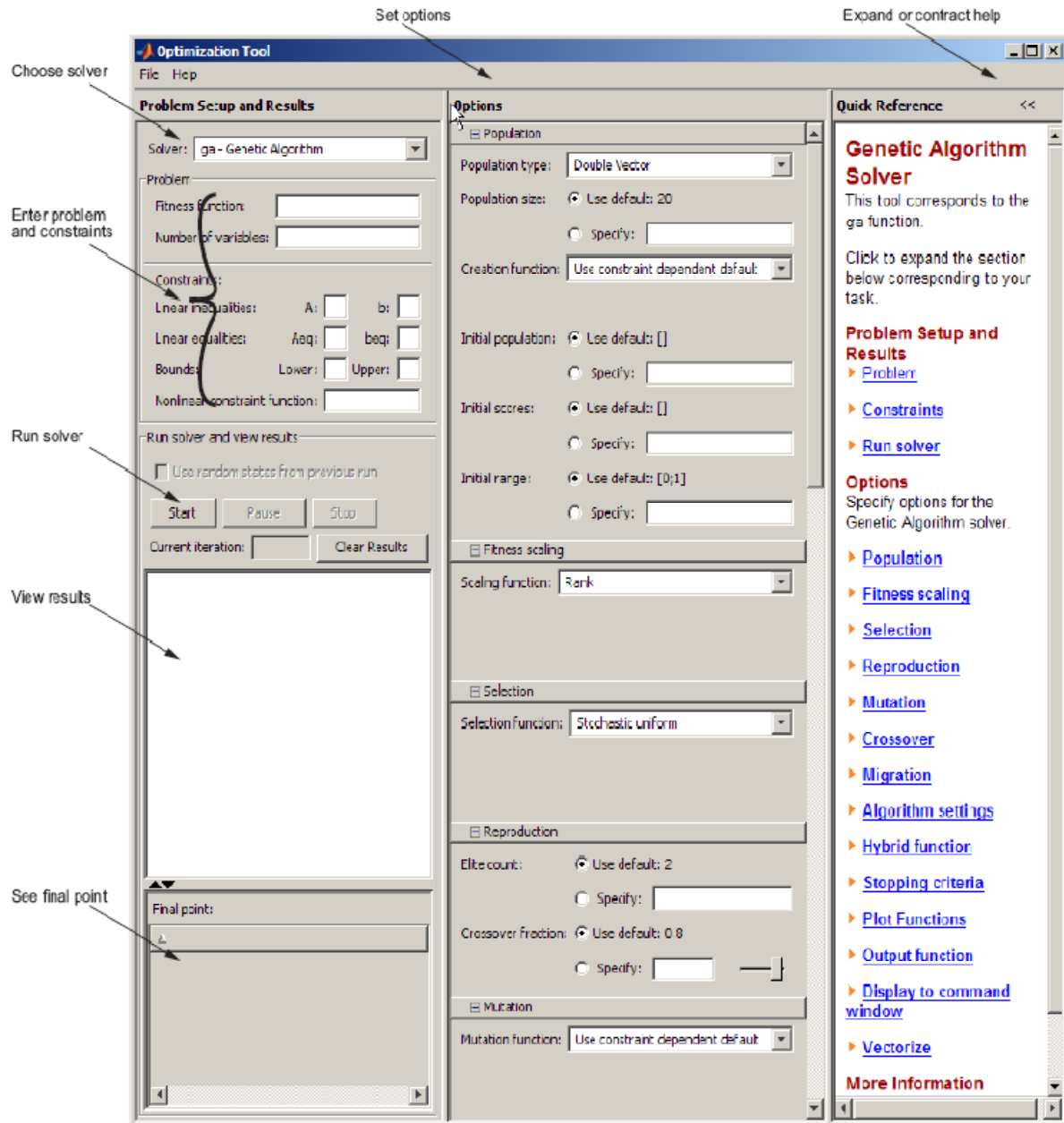


Fig 4.1 Launching Optimization Tool

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



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

Fig: 4.2 GA tool box in MATLAB

4.4 DIFFERENT PARAMETERS IN GA TOOL

- Populations
- Fitness scaling
- Selection
- Reproduction
- Mutation
- Crossover
- Stopping criteria

Display to the command window.

4.4.1 Population

Population options specify options for the population of the genetic algorithm. Population type specifies the type of the input to the fitness function. It is easy to set Population type to be double vector, or Bit string, or Custom. If select Custom, then it is necessary to write your own creation, mutation, and crossover functions that work with your population type, and specify these functions in the fields Creation function, Mutation function, and Crossover function, respectively. (* MATLAB R2008b uses the default population type as double vector, whereas all the standard textbooks use Bit string as the population type).

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

4.4.2 Fitness Scaling

The scaling function converts raw fitness scores returned by the fitness function to values in a range that is suitable for the selection function. Rank scales the raw scores based on the rank of each individual, rather than its score. The rank of an individual is its position in the sorted scores. The rank of the fittest individual is 1, the next fittest is 2, and so on. Rank fitness scaling removes the effect of the spread of the raw scores.

4.4.3 Selection

The selection function chooses parents for the next generation based on their scaled values from the fitness scaling function. It is easy to specify the function that performs the selection in the Selection function field.

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

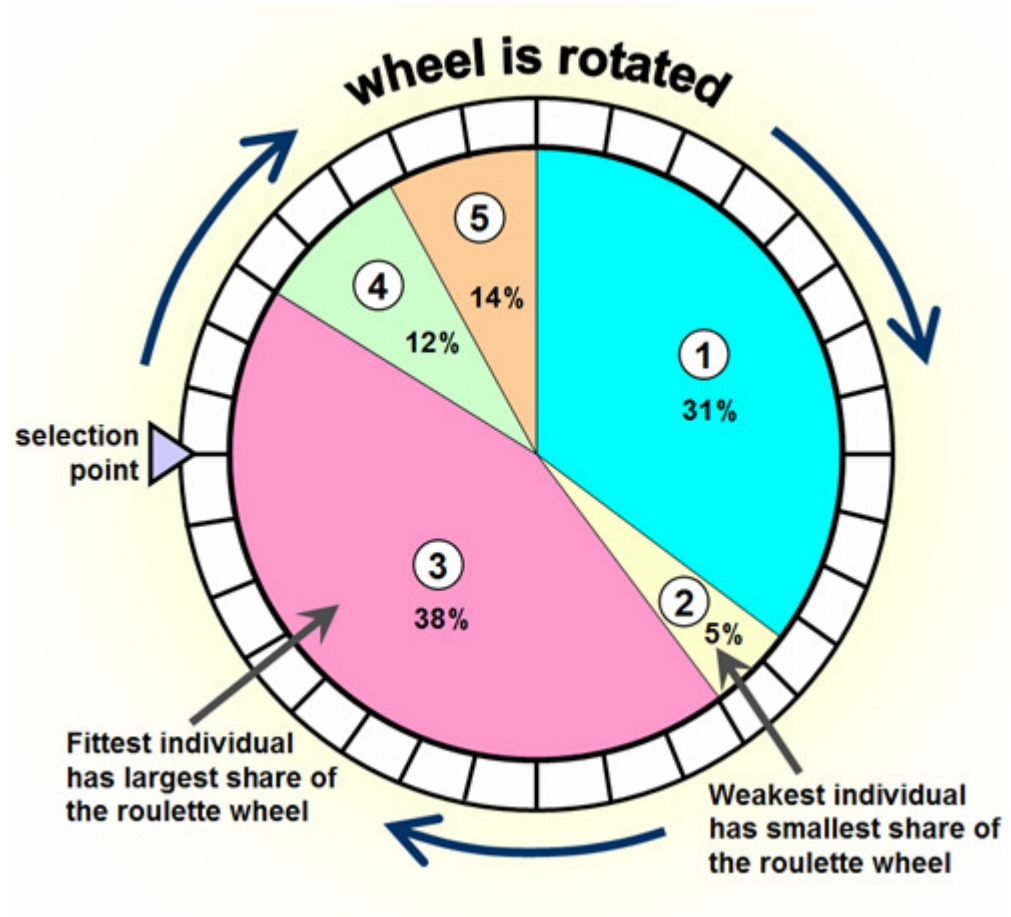


Fig 4.3 Roulette wheel

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

4.4.4 Reproduction

Reproduction options determine how the genetic algorithm creates children at each new generation. Elite count specifies the number of individuals that are guaranteed to survive to the next generation. Set Elite count to be a positive integer less than or equal to Population size.

Crossover fraction specifies the fraction of the next generation, other than elite individuals, that are produced by crossover.

4.4.5 Mutation

Mutation functions make small random changes in the individuals in the population, which provide genetic diversity and enable the Genetic Algorithm to search a broader space. It is easy to specify the function that performs the mutation in the Mutation function field. The default option in Mutation function field is Gaussian. Gaussian is normally used for unconstrained problems. For constrained problems adapt feasible option is used.

4.4.6 Stopping Criteria

Stopping criteria determines what causes the algorithm to terminate.

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

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

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

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

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

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

CHAPTER-5

RESULTS AND DISCUSSION

5.1 INTRODUCTION

The proposed method has been applied to two power systems on six-generator system and eleven-generator power systems. CEED problem is formulated in which cost function and emission function is minimized. Penalty factor h_i is varied and for various loads (500,600,700,800,900,1000,1100MW for six-generator system and 1000,1250,1500 ,1750, 2000,2250 ,2500MW for eleven -generator system respectively) and minimum value of fuel cost and emission output is obtained. The fuel cost, emission coefficients and generation limits of six-generator system are taken from [1,10,45,46] and are given in Table 1. Table 5.3.1 gives the best optimal power output of generators for CEED problem using proposed method with system demands rising from 500 - 1100 MW for six generator system. The fuel cost, emission coefficients and generation limits of eleven-generator system are taken from [1,10,45,46] are given in Table 2. Table 5.3.2 gives the best optimal power output of generators for CEED problem using proposed method with system demands rising from 1000 - 2500 MW for eleven-generator system.

Table 1. Fuel cost, emission coefficients and generation limits of six-generator system.

Generator	a_i (\$/MW ² h)	b_i (\$/MW ^h)	c_i (\$/h)	d_i (kg/MW ² h)	e_i (kg/MW ² h)	f_i (kg/h)	$P_{i\min}$ (MW)	$P_{i\max}$ (MW)
1	0.1525	38.540	756.800	0.0042	0.3300	13.860	10	125
2	0.1060	46.160	451.325	0.0042	0.3300	13.860	10	150
3	0.0280	40.400	1050.000	0.0068	-0.5455	40.267	35	225
4	0.0355	38.310	1243.530	0.0068	-0.5455	40.267	35	210
5	0.0211	36.328	1658.570	0.0046	-0.5112	42.900	130	325
6	0.0180	38.270	1356.660	0.0046	-0.5112	42.900	125	315

Table 2. Fuel cost, emission coefficients and generation limits of eleven-generator system.

Generator	a_i (\$/MW ² h)	b_i (\$/MWh)	c_i (\$/h)	d_i (kg/MW ² h)	e_i (kg/MW ² h)	f_i (kg/h)	$P_{i\min}$ (MW)	$P_{i\max}$ (MW)
1	0.00762	1.92699	387.85	0.00419	-0.67767	33.93	20	250
2	0.00838	2.11969	441.62	0.00461	-0.69044	24.62	20	210
3	0.00523	2.19196	422.57	0.00419	-0.67767	33.93	20	250
4	0.00140	2.01983	552.50	0.00683	-0.54551	27.14	60	300
5	0.00154	2.22181	557.75	0.00751	-0.40060	24.15	20	210
6	0.00177	1.91528	562.18	0.00683	-0.54551	27.14	60	300
7	0.00195	2.10681	568.39	0.00751	-0.40006	24.15	20	215
8	0.00106	1.99138	682.93	0.00355	-0.51116	30.45	100	455
9	0.00117	1.99802	741.22	0.00417	-0.56228	25.59	100	455
10	0.00089	2.12352	617.83	0.00355	-0.41116	30.45	110	460
11	0.00098	2.10487	674.61	0.00417	-0.56228	25.59	110	465

5.2 Formulation of CEED problem

Two aspects of the optimal power dispatch problem considered in 2D space are:

- 1) To minimize the cost of generation.
- 2) To minimize the system emission output.

The list of symbols used in this section is as follows:

F_T = Total generation cost of the system.

FC = Total fuel cost of generators.

NC = Total emission of generators.

n = Number of generators connected in the network.

h_i = Price penalty factor of unit i .

P_i = Power generation of unit i .

$P_{i\min}$ = Minimum generation of unit i .

$P_{i\max}$ = Maximum generation of unit i .

P_{load} = Total load of the system.

The bi-objective combined economic emission dispatch problem is converted into single optimization problem by introducing price penalty factor h_i as follows:

$$\text{Minimize: } F_T = FC + h_i * EC$$

The price penalty factor blends the emission with fuel cost and is the total operating cost in U.S dollars per hour. The price penalty factor h_i is the ratio between the maximum fuel cost and maximum emission of corresponding generator:

$$h_i = \frac{FC(P_{i\max})}{EC(P_{i\max})} \quad ; \quad i=1,2,\dots,n$$

The following steps are used to find the price penalty factor for a particular load demand.

- 1) Find the ratio between maximum fuel cost and maximum emission of each generator.
- 2) Arrange the values of price penalty factor in ascending order.
- 3) Add the maximum capacity of each unit ($P_{i\max}$) one at a time, starting from the smallest h_i unit, until

$$\Sigma (P_{i\max}) \geq P_d$$

- 4) At this stage h_i associated with the last unit in the process is the price penalty factor h_i for the given load.

The procedure gives the approximate value of price penalty factor computation for the corresponding load. The CEED problem is to find the optimal combination of power generation that minimizes the total fuel cost while satisfying the total demand and power system constraints.

The CEED can be formulated as:

$$\text{CEED} = \begin{cases} \text{Problem} & F_T = \text{Minf}(FC, EC); \\ \text{subject to} & P_{\text{load}} - \sum_{i=1}^n P_i = 0 \\ & P_{i\min} \leq P_i \leq P_{i\max} \end{cases}$$

The total fuel cost (FC) for power generation units should be as a quadratic polynomial

$$FC = \sum_{i=1}^n (a_i P_i^2 + b_i P_i + c_i)$$

where a_i , b_i and c_i are the cost coefficients of the i^{th} generating unit.

Total emission of generation (EC) can be expressed as:

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

where d_i , e_i and f_i are emission coefficients of the i^{th} generating unit.

The bi-objective combined economic emission dispatch problem is converted into single optimization problem by introducing a price penalty factor h_i as follows

$$\text{Min}F_T = \sum_{i=1}^n ((a_i P_i^2 + b_i P_i + c_i) + h_i (d_i P_i^2 + e_i P_i + f_i))$$

Note that when $h_i = 0$, the combined problem reduces to conventional fuel cost dispatch problem. Likewise, the problem becomes a pure emission dispatch when $h_i = \infty$.

The price penalty factor blends the emission with fuel cost and it is the ratio between the maximum fuel cost and maximum emission of the corresponding generator :

$$h_i = \frac{a_i P_{i\text{max}}^2 + b_i P_{i\text{max}} + c_i}{d_i P_{i\text{max}}^2 + e_i P_{i\text{max}} + f_i}$$

Values of h_i obtained from above procedure is arranged in tabular form for all given loads for six generator and eleven generator system.

Table for six generator system:

Load(MW)	500	600	700	800	900	1000	1100
h_i	44.1497	44.9229	44.9229	44.9229	48.1142	48.1142	68.8289

Table for eleven generator system:

Load(MW)	1000	1250	1500	1750	2000	2250	2500
h_i	2.8016	2.989	2.989	3.0092	3.39	3.39	3.896

The values of h_i is taken from above table for specified load and MATLAB OPTIMTOOL program is execute by varying the value of h_i , starting with the value obtained from above procedure. Best value of price penalty factor is such that for which minimum value of fuel cost and emission output is obtained.

The m-files for constraint function and objective function are written below:-

Constraint function for six generator system:

```
function [a,ceq]=constraint_6bus(x)
a=[-x(1)+10;x(1)-125;-x(2)+10;x(2)-150;-x(3)+35;x(3)-225;-x(4)+35;x(4)-210;-
x(5)+130;x(5)-325;-x(6)+125;x(6)-315];
ceq=(x(1)+x(2)+x(3)+x(4)+x(5)+x(6)-1100);
```

Objective function for six generator system:

```
x(1)=input('power in 1st gen=');x(2)=input('power in 2nd gen='); x(3)=
input('power in 3ndgen=');x(4)=input('power in 4th gen=');x(5)=input('power
in5thgen=');x(6)=input('powerin6thgen=');cost=((0.1525*x(1)*x(1))+(38.540*x
(1))+756.800)+((0.1060*x(2)*x(2))+(46.160*x(2))+451.325)+((0.0280*x(3)*x(3)
)+(40.400*x(3))+1050.00)+((0.0355*x(4)*x(4))+(38.310*x(4))+1243.53)+((0.021
1*x(5)*x(5))+(36.328*x(5))+1658.57)+((0.0180*x(6)*x(6))+(38.270*x(6))+1356.
66);
emission=((0.0042*x(1)*x(1))+(0.3300000*x(1))+13.860)+((0.0042*x(2)*x(2))
+(0.3300000*x(2))+13.860)+((0.0068*x(3)*x(3))+(0.5455)*x(3))+40.267)+
((0.0068*x(4)*x(4))+((-0.5455*x(4))+40.267)+((0.0046*x(5)*x(5))+((-0.5112)
*x(5))+42.900)+ ((0.0046*x(6) *x(6))+((-0.5112)*x(6))+42.900);
disp([cost,emission])
```

Constraint function for eleven generator system:

```
function [a,ceq]=constraint_11bus(x)
a=[-x(1)+20;x(1)-250;-x(2)+20;x(2)-210;-x(3)+20;x(3)-250;-x(4)+60;x(4)-
300;-x(5)+20;x(5)-210;-x(6)+60;x(6)-300;-x(7)+20;x(7)-215;-x(8)+100;x(8)-
455;-x(9)+100;x(9)-455;-x(10)+110;x(10)-460;-x(11)+110;x(11)-465];

ceq=(x(1)+x(2)+x(3)+x(4)+x(5)+x(6)+x(7)+x(8)+x(9)+x(10)+x(11)-1750);
```

Objective function for eleven generator system:

```
x(1)=input('power in 1st gen=');x(2)=input('power in 2nd
gen=');x(3)=input('power in 3rd gen=');x(4)=input('power in 4th
gen=');x(5)=input('power in 5th gen='); x(6)=input('power in 6th
gen=');x(7)=input('power in 7nd gen='); x(8)=input('power in 8th gen=')
;x(9)=input('power in 9th gen='); x(10)=input('power in 10th gen=');
x(11)=input('power in 11thgen=');
```

```
cost=((.00762*x(1)*x(1))+(1.92699*x(1))+387.85)+((.00838*x(2)*x(2))+(2.11969*
x(2))+441.62)+((.00523*x(3)*x(3))+(2.19196*x(3))+422.57)+((.00140*x(4)*x(4))+
(2.01983*x(4))+552.50)+((.00154*x(5)*x(5))+(2.22181*x(5))+557.75)+((.00177*x(
6)*x(6))+(1.91528*x(6))+562.18)+((.00195*x(7)*x(7))+(2.10681*x(7))+568.39)+((
.00106*x(8)*x(8))+(1.99138*x(8))+682.93)+((.00117*x(9)*x(9))+(1.99802*x(9))+7
41.22)+((.00089*x(10)*x(10))+(2.12352*x(10))+617.83)+((.00098*x(11)*x(11))+(2
.10487*x(11))+674.61);emission=((.00419*x(1)*x(1))+((-
.67767)*x(1))+33.93)+((.00461*x(2)*x(2))+((-0.69044)*x(2))+24.62) + ((.00419
*x(3)*x(3))+((-0.67767)*x(1))+33.93)+((.00683*x(4)*x(4))+((-0.54551)
*x(4))+27.14)+((.00751*x(5)*x(5))+((-0.40060)*x(5)) +24.15)+ ((.00683*x(6)
*x(6))+((-0.54551)*x(6))+27.14)+((.00751*x(7)*x(7))+((-0.40006)*x(7)) +24.15)
+((.00355*x(8)*x(8))+((-0.51116)*x(8))+30.45)+((.00417*x(9)*x(9))+((-0.56228)
*x(9))+25.59)+((.00355*x(10)*x(10))+((-0.41116)*x(10))+30.45) +((.00417*
x(11)*x(11))+((-0.56228)*x(11))+25.59);disp([ cost , emission])
```

This MATLAB Program is executed for different values of h_i until minimum value of fuel cost and emission output is obtained. Tested values of h_i is shown in tabular form for different loads for six generator and eleven generator system.

Tables for six generator system for different loads:

Table-5.2.1.for 500 MW load

h	p1	p2	p3	p4	p5	p6	Fuel cost (\$/h)	emission (kg/h)
44.1497	31.355414	23.28492	88.33277	89.47235	135.0272	132.5274	27184	257
40.1497	30.96541	22.33529	88.33166	89.57159	135.7552	133.0409	27174	257
35.14	30.396301	20.94585	88.30872	89.70353	136.8447	133.8009	27160	258
30.155	29.717015	19.27997	88.2451	89.83799	138.1912	134.7288	27144	258
25.15	28.882349	17.22521	88.1055	89.96638	139.9197	135.9008	27124	259
22.15	28.285752	15.75148	87.95941	90.03103	141.2099	136.7625	27111	259
20.15	27.839051	14.64326	87.81951	90.06056	142.2134	137.4243	27101	260
19.15	27.597453	14.04497	87.73351	90.07065	142.7649	137.7885	27096	260
18.15	27.3441	13.41311	87.63312	90.07388	143.3623	138.1735	27090	260
17.15	27.076146	12.7469	87.51755	90.07246	144.0005	138.5864	27085	261

Table-5.2.2 for 600 MW load

h_i	p1	p2	p3	p4	p5	p6	Fuel cost (\$/h)	Emission (kg/h)
44.9229	45.31759	39.53938	102.5452	103.3545	155.7085	153.5348	31812	331
40.905	44.58691	38.33545	102.6797	103.5562	156.5985	154.2433	31793	331
35.9	43.53622	36.58274	102.8551	103.8338	157.9106	155.2815	31766	332
30.9	42.2897	34.47094	103.0321	104.1417	159.5206	156.545	31735	333
25.9	40.78441	31.87557	103.1933	104.4756	161.5501	158.121	31699	334
22.9	39.72049	30.0122	103.2676	104.6822	163.0453	159.2722	31675	335
20.9	38.92926	28.6092	103.2957	104.8197	164.1969	160.1492	31658	336
18.9	38.05999	27.05377	103.2999	104.9503	165.5002	161.1359	31639	337
17.9	37.59152	26.2115	103.2899	105.0113	166.2206	161.6752	31629	338
17	37.15104	25.41104	103.27	105.0619	166.9108	162.1952	31620	338

Table-5.2.3. for 700 MW load

h_i	p1	p2	p3	p4	p5	p6	Fuel cost (\$/h)	emission (kg/h)
44.9229	59.20977	55.62387	116.7582	117.2547	176.52	174.6335	36610	422
40.9	58.12935	54.14455	117.0267	117.5602	177.5895	175.5497	36578	423
35.9	56.58264	51.99251	117.3959	117.9846	179.157	176.8873	36534	424
30.9	54.75233	49.39171	117.8064	118.4641	181.072	178.5135	36482	426
25.5	52.3544	45.89941	118.2905	119.047	183.6884	180.7204	36418	428
22.5	50.77645	43.54987	118.5686	119.3971	185.4837	182.2243	36378	430
20	49.29344	41.30401	118.7968	119.6957	187.2306	183.6795	36341	431
19	48.65055	40.31928	118.8833	119.8159	188.0072	184.3238	36326	432
18.5	48.31628	39.8061	118.9265	119.8751	188.4141	184.6618	36318	433
18	47.97594	39.27926	118.9658	119.9349	188.8351	185.0089	36310	433
17.8	47.83704	39.06416	118.9815	119.9586	189.0076	185.151	36307	433
17.65	47.73195	38.90127	118.9937	119.9757	189.138	185.2593	36305	433

Table-5.2.4. for 800 MW load

h_i	p1	p2	p3	p4	p5	p6	Fuel cost (\$/h)	emission (kg/h)
44.9229	73.1021	71.70828	130.9712	131.1551	197.3313	195.732	41582	531
38.9	70.89466	68.98908	131.5901	131.7836	199.2682	197.4744	41507	533
35.5	69.45169	67.1765	131.9854	132.1843	200.5643	198.6379	41460	534
30	66.73293	63.68622	132.7057	132.9136	203.0755	200.8861	41373	537
25	63.73022	59.71806	133.4554	133.6711	205.9627	203.4626	41282	540
22	61.61244	56.84883	133.948	134.168	208.0788	205.344	41220	543
20	60.03811	54.67847	134.2915	134.5128	209.6985	206.7806	41176	545
19	59.19539	53.50307	134.4656	134.6885	210.5835	207.564	41153	546
18.8	59.02171	53.2604	134.5006	134.7239	210.7671	207.7263	41148	546

18.7	58.93459	53.13771	134.5179	134.7418	210.8599	207.8082	41146	546
18.65	58.89078	53.0763	134.5267	134.7504	210.9063	207.8495	41145	546
18.63	58.87334	53.05159	134.5303	134.7538	210.925	207.8661	41144	546
18.6	58.84676	53.01475	134.5357	134.759	210.9534	207.8904	41144	546
18.57	58.82041	52.97755	134.5405	134.7648	210.9811	207.9156	41143	546
18.56	58.81169	52.96523	134.5425	134.7663	210.9908	207.9235	41143	546
18.53	58.78501	52.92806	134.5476	134.7719	211.0186	207.9487	41142	546

Table-5.2.5. for 900 MW load

h_i	p1	p2	p3	p4	p5	p6	Fuel cost (\$/h)	emission (kg/h)
48.1142	88.26437	89.22081	144.7979	144.6852	217.1388	215.893	46776	656
45	87.02648	87.82887	145.1744	145.0463	218.1171	216.8068	46727	657
40	84.78845	85.27273	145.8491	145.6902	219.9157	218.4839	46641	659
35	82.17459	82.21865	146.6262	146.4262	222.0665	220.4879	46543	661
30	79.07956	78.50735	147.5267	147.2708	224.6881	222.9275	46431	665
25	75.35526	73.90515	148.5735	148.2384	227.9577	225.9699	46302	670
20	70.78274	68.05276	149.7865	149.3296	232.1664	229.882	46153	676
19	69.73985	66.68699	150.048	149.5611	233.1591	230.8049	46121	678
18.7	69.41723	66.26228	150.128	149.6309	233.4694	231.0922	46111	678

Table-5.2.6. for 1000 MW load

h_i	p1	p2	p3	p4	p5	p6	Fuel cost(\$/h)	emission (kg/h)
48.1142	102.4072	105.4961	158.917	158.5115	237.8228	236.8455	52111	798
45	100.9249	103.9184	159.3853	158.9442	238.9253	237.9018	52045	800
40	98.2482	101.0166	160.2285	159.7189	240.9485	239.8393	51929	803
35	95.1234	97.54753	161.2058	160.606	243.3635	242.1537	51796	806
30	91.42596	93.32901	162.3479	161.6277	246.2995	244.9699	51645	811
25	86.98051	88.09236	163.6924	162.805	249.9523	248.4774	51472	817

20	81.52785	81.42712	165.2802	164.1476	254.6344	252.9828	51273	826
19.5	80.91296	80.65969	165.4547	164.2894	255.177	253.5062	51252	827
19.4	80.78869	80.50369	165.4889	164.3189	255.2878	253.612	51248	827

Table-5.2.7. for 1100 MW load

h_i	p1	p2	p3	p4	p5	p6	Fuel cost(\$/h)	Emission (kg/h)
61.8289	122.6791	127.8592	171.0424	170.5207	254.2517	253.6469	57935	953
55	119.8882	125.125	171.9488	171.3513	256.1695	255.5173	57791	955
50	117.5244	122.7608	172.7176	172.0506	257.8201	257.1265	57671	957
45	114.8235	120.0075	173.5957	172.8432	259.7335	258.9965	57536	960
40	111.7079	116.7604	174.6079	173.7475	261.9813	261.1949	57385	964
35	108.0724	112.8761	175.7856	174.7859	264.6606	263.8195	57213	968
30	103.7729	108.1499	177.1693	175.9845	267.9118	267.0115	57018	974
25	98.60538	102.2796	178.811	177.372	271.947	270.985	56793	983
20	92.27258	94.80116	180.7754	178.9645	277.1023	276.0841	56536	994