ECONOMIC LOAD DISPATCH USING EVOLUTIONARY ALGORITHMS

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Power Systems & Electric Drives

Thapar University, Patiala

By:
SIMARJIT KAUR
(Regn. No. 80641023)

Under the supervision of:
Dr. Sanjay K. Jain

Assistant Professor (EIED)

JUNE 2008

ELECTRICAL & INSTRUMENTATION ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, **"Economic Load Dispatch using Evolutionary Algorithms"**, in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Power Systems & Electric Drives* submitted in Electrical & Instrumentation Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of Dr. Sanjay K. Jain, Assistant Professor, EIED. The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.

(Simarjit Kaur)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Dr. Sanjay K. Jain)
Assistant Professor, EIED
Thapar University
Patiala

Countersigned by :

(Dr. Smarajit Ghosh)
Professor & Head
Electrical & Instrumentation Engg. Department
Thapar University
Patiala

(Dr. R.K.Sharma)
Dean(Academic Affairs)
Thapar University,
Patiala.

ACKNOWLEDGEMENT

I wish to express my special thanks and deepest regards to my Supervisor, Dr. Sanjay K. Jain, Assistant Professor, Electrical & Instrumentation Engineering Department, for providing me invaluable guidance, suggestions, and support which have helped me to submit this Thesis report in time.

I take this opportunity to express my gratitude and sincere thanks to Dr.Smarajit Ghosh, Professor & Head, and the faculty members of Electrical & Instrumentation Engineering Department particularly Mrs. Manbir Kaur, Mr. Parag Nijhawan, Mr. Yogesh K. Chauhan for their valuable advice and suggestion during the thesis work.

I wish to thank all my classmates for their time to time suggestions and cooperation. Finally it is the support, encouragement and good wishes of my parents and other family members without which I would not have been able to complete my M.E

I thank and owe my deepest regards to all who have helped me directly or indirectly.

SIMARJIT KAUR
(80641023)

ABSTRACT

Ensuring a smooth electrical energy to the consumer has been identified as the main role of electric supply utility. The power utility needs to ensure that the electrical power is generated with minimum cost. Hence, for economic operation of the system, the total demand must be appropriately shared among the generating units with an objective to minimize the total generation cost for the system. Thus, Economic load dispatch (ELD) is one of the important problems of power system operation and control. This work proposes evolutionary optimization techniques namely Genetic Algorithm and Evolutionary Programming to solve ELD in the electric power system, which are generic population, based probabilistic search optimization algorithms and can be applied to real world problem.

In this thesis, the two main types of EAs, which are genetic algorithm (GA) and evolutionary programming (EP), are respectively applied to solve an ELD problems. Also, a Classical Lagrange Multiplier Method is used to solve the same problem. And at the last the comparison between the three methods has been presented. The EAs provides the generation level such that the total losses are reduced and the generation cost is coming out to be lower than the cost resulted with Lagrange Multiplier method.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER -1

INTRODUCTION

In this chapter, the overview of the problem and literature review is summarized in brief. The objectives of the work and the outline of the thesis is discussed.

1.1. Overview

Generating efficiency has been considered as one of the most important issues in most power plants. This is due to the fact that, so many intensive competitions have experienced in the electrical supply industries. The remote location of power plant from the load centre has been identified as one of the reasons which caused high cost. The increase in fuel cost these days has also contributed to this phenomenon. Therefore, economic load dispatch is implemented in order to determine the Output (generating) of each generator so that the total generation cost will be minimized.

The generator's output has to be varied within limits so as to meet a particular load demand and losses with minimum fuel cost **[1].** Thus, ELD is one of the important topics to be considered in power system engineering.

Solving economic dispatch problems normally involved optimization process. So, new techniques are continuously being developed that result in a more efficient system operation. As a good global optimal tool, evolutionary algorithm (EA) is widely used in the recent past in many engineering fields. EAs differ from traditional optimization techniques in that they involve a search from a "population" of solutions, not from a single point.

In this thesis, genetic algorithm and evolutionary programming are used for optimization of economic load dispatch and the results are compared comprehensively with Classical Lagrangian Method (LMM).

1.2. Literature Review

Despite extensive research focusing on thermal power dispatch problem, much of the efforts still today have involved so as to obtain the most optimal dispatch at most optimal cost.

The conventional economic dispatch discussed by Kirchmayer **[42]**, takes into account, incremental production cost equations as deterministic. Several inaccuracies and uncertainties lead to deviation from optimal operation. With rising fuel costs, there is growing interest to account for these deviations, since the effect of inaccuracies results in an increase in the overall cost.

Dhillon et. al **[43]** formulated the problem as multiobjective one. They considered objectives such as operating cost, minimal emission and minimum transmission losses in thermal power dispatch systems, considering uncertainties and inaccuracies in system data. The validity and effectiveness of the method was demonstrated by analysing a 6-generator case.

A general survey of the present status of economic dispatch had been done by Happ in 1974 **[48,49]** which reviewed the progress of economic dispatch going as far back as the early 1920's, when engineers were concerned with the problem of economic allocation of generation or the proper division of the load among the generating units available. Prior to 1930, various methods are in use such as: (a) the base load method where the next most efficient unit is loaded to its maximum capability, then the second most efficient unit is loaded, etc. (b)"best point loading," where units are successively loaded to their lowest heat rate point, beginning with the most efficient unit and working down to the least efficient unit. Different methods for ELD are discussed in **[51],** described as follows:1) Weighted least squares, 2) Gauss-Newton method, and many more.

The ELD is a non linear programming problem and is used to determine optimal outputs of generators. The power generation limits are set with an objective to minimize the total fuel cost while system is operating within limits. Since its introduction as network constraints economic dispatch by Carpentier **[39]** and its definition as by Dommel and Tinney **[40]**, the ELD problem has been the object of intensive research. Carpentier **[39]** introduced a generalized, nonlinear programming formulation of the economic load dispatch. It was later named optimal power flow (OPF) by Dommel and Tinney **[40]**.

Upto the present time, several research methods have been reported. The principle methods employed for ELD problem are Newton-based method and the linear programming method which have emerged as the dominant optimization methods for solving ELD. The earlier Newton based approaches were based on the method of Dommel and Tinney **[40]** where the first partial derivatives of the equations. These use a search direction in the iterative procedure to find a solution.

Several approaches have been discussed to overcome the drawbacks of classical ELD problem. Some of these methods have been based on successive linear programming and successive quadratic programming described by different authors in literature. Different methods for Power System Operations has been discussed by Miller and Malinnowski **[1].**

According to Palanichamy and K.shrikrishna**[2]** discussed Simple algorithm for economic power dispatch for optimizing the problem while satisfying a set of system operating constraints, including constraints dictated by the electric network. ELD has been widely used in power system operation and planning discussed by Wood and Woolenberg in **[34].**

These days, genetic algorithm (GA) **[20, 22 to 25]** and evolutionary programming techniques **[26 to 28]** have been suggested to overcome difficulties of classical methods. GA, invented by Holland **[20]** in the early 1970's, is a stochastic global search method that mimics the metaphor of natural biological evolution. GA's operate on a population of candidate's solution encoded to string called chromosomes in order to obtain optimality. Each chromosome exchanges information by using string operators borrowed from natural genetic to produce the better solution. Although GA seems to be a good method to solve optimization problem, sometimes the solution obtained from GA is only a near global optimum solution **[22]**. GA and EP comes under the category of EA. In EP, invented by Fogel**[44, 45, and 52]**, the state of the machine is mutated**.** Therefore, EP has been become an optimal tool and was used in many practical problems.

Saddukaran **[25]** used refined genetic algorithm for the solution of multi objective problem. In this method, the author have used genetic operators like reproduction, crossover, mutation and one additional operator Elitism which stores the fittest string from each population and so the process is faster as well as stores the best solutions to the problem.

Randy and Sue **[37]** discussed the solution methodology of GA. The examples help in choosing GA parameters and also broaden the view of problems being solved.

Deb **[10]** proposed many approaches and methods to solve optimization problems. The solution methods are broadly grouped under two major categories: traditional and non traditional methods. Also, two distinct types of optimization algorithms are discussed. (1) Deterministic, with specific rules for moving from one solution to the other. (2) Probabilistic (newer as compare to deterministic and are easy to understand) which are stochastic in nature.

1.3. Objective of work

The objectives of the thesis work are summarised as follows –

1. To find solution of economic load dispatch problem so that the total fuel cost is minimized while satisfying the power generation limits.
2. Use global search techniques like GA/EP to find the optimal settings.
3. Investigate the effectiveness of these methods for ELD problem while neglecting the transmission losses and considering the transmission losses.
4. Compare the results obtained from the above methods with the results obtained from classical method.

1.4. Organization of Thesis

The thesis is organised into five chapters. The organisation of chapters is as follows:

The **chapter-1** summarized the overview of the problem, brief literature review, scope of work and organization of the thesis.

The **chapter-2** highlights the topic ELD and also gives the overview of various classical methods and economic considerations etc. are also discussed in brief. Also, ELD solution using LMM is discussed in this chapter.

The **chapter-3** explores the associated structure of GA and EP. The problem formulation using GA and EP is discussed and the algorithm for the same is presented in this chapter.

The **chapter-4** outlines the simulated results with losses and without losses using EP and GA for cases having different generators. Comparisons of the results from these techniques are done.

The **chapter-5** presents the conclusions drawn and also outlines the possible avenues for future work.

CHAPTER-2

ECONOMIC LOAD DISPATCH

In this chapter, the Basic concepts of economic load dispatch (ELD), problem formulation, classical solution methods and the solution of ELD using Lagrange Multiplier method are summarized in brief.

2.1. Basic Concept

Electrical energy can not 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 may consist of main 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 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. The efficient and optimum economic operation and planning of electric power generation systems have always occupied an important position in the electric power industry. General area of economic load dispatch can be categorized into four main categories:

- Optimal power flow.
- Economic dispatch in relation to AGC.
- Dynamic dispatch.
- Economic dispatch with non-conventional generation sources.

In a typical power system, multiple generators are implemented to provide enough total output to satisfy a given total consumer demand. Each of these generating stations can, and usually does, have a unique cost-per-hour characteristic for its output operating range. A station has incremental operating costs for fuel and maintenance; and fixed costs associated with the station itself that can be quite considerable in the case of a nuclear power plant. For example, things get even more complicated when utilities try to account for transmission line losses, and the seasonal changes associated with hydroelectric plants. In all of these cases, however, the basic objective is to operate the system as inexpensively as possible. Thus, load dispatching should be done appropriately.

2.1.1. Load Dispatching:

The operation of a modern power system has become very complex. It is necessary to maintain frequency and voltage within limits in addition to ensuring reliability of power supply and for maintaining the frequency and voltage within limits it is essential to match the generation of active and reactive power with the load demand. For ensuring reliability of power system it is necessary to put additional generation capacity into the system in the event of outage of generating equipment at some station. Over and above it is also necessary to ensure the cost of electric supply to the minimum. The total interconnected network is controlled by the load dispatch centre. The load dispatch centre allocates the MW generation to each grid depending upon the prevailing MW demand in that area. Each load dispatch centre 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 centre is to keep the exchange of power between various zones and system frequency at desired values.

2.1.2. Why generation scheduling is necessary:

In a practical power system, the power plants are not located at the same distance from the centre of loads and there fuel costs are different. Also under normal operating, the generation capacity is more than the total load demand and losses. Thus, there are many options for scheduling generation. In an interconnected power system, the objective is to find the real and reactive power scheduling of each power plant in such a way so as to minimize the operating cost. This means that the generators real and reactive powers are allowed to vary within certain limits so as to meet a particular load demand with minimum fuel cost. This is called the “Economic load dispatch” (ELD) problem.

The objective functions, also known as cost functions may present economic cost system security or other objectives. The transmission loss formula can be derived and the economic dispatch of generation based on the loss formula can also be obtained. The Loss coefficients are known as B-coefficients.

A major challenge for all power utilities is not only to satisfy the consumer demand for power, but to do so at minimal cost. Any given power system can be comprised of multiple generating stations having number of generators and the cost of operating these generators does not usually correlate proportionally with their outputs; therefore the challenge for power utilities is to try to balance the total load among generators that are running as efficiently as possible.

The economic load dispatch (ELD) problem assumes that the amount of power to be supplied by a given set of units is constants for a given interval of time and attempts to minimize cost of supplying this energy subject to constraints of the generating units. Therefore, it is concerned with the minimization of total cost incurred in the system and constraints over the entire dispatch period **[6].**

Therefore, the main aim in the economic dispatch problem is to minimize the total cost of generating real power (production cost) at various stations while satisfying the loads and the losses in the transmission links. For simplicity we consider the presence of thermal plants only in this thesis.

2.1.3. Economics of Power Generation of Thermal Plant:

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 (2.1).

$$C_i\,(P_i\,(t)) = \sum(a_i\,P_i^2 + b_i\,P_i + c_i\,) \qquad (i=1,2,...,N_g) \qquad (2.1)$$

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 guaranteed 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.

Cost of fuel used for economics of power generation is specified by the input-output curve of a generating unit. This is explained below:

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. 2.1(a). Converting the ordinate of heat rate curve from BTU/hr to Rs/hr. results in the **fuel cost curve** shown in fig. 2.1(b).

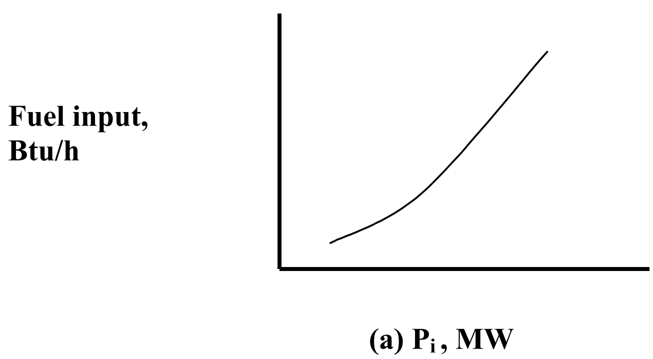

Fig. 2.1(a). Heat- rate curve

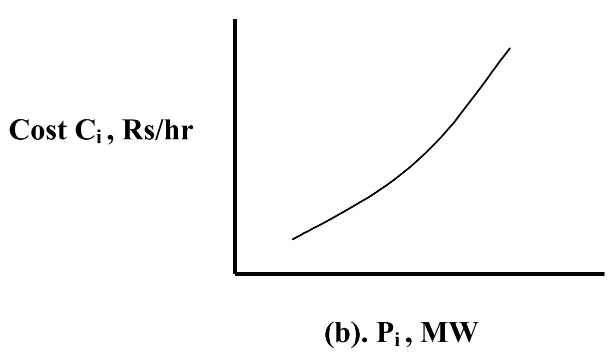

Fig. 2.1(b). Fuel-rate curve

In all practical cases, the fuel cost of generator *i* can be represented as a quadratic function of real power generation

$$C_i = a_i + b_i\, P_i + c_i\, P_i^2 \qquad (2.2)$$

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. 2.2.

$$dC_i/dP_i = 2a_i\, P_i + b_i \qquad (2.3)$$

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 labour, 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.

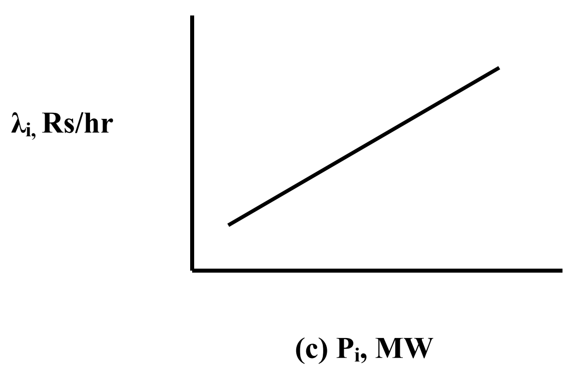

Fig. 2.1(c) Incremental fuel-cost curve

2.1.4. Transmission Losses:

When transmission distances are very small and load density is very high, transmission losses may be neglected and the optimal despatch 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

$$P_L=\sum\sum P_i B_{ij} P_j \qquad (i, j=1, 2,\ldots\ldots,n_g) \qquad (2.4)$$

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

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

$$P_L = \sum\sum P_i B_{ij} P_j + \sum B_{0i} P_i + B_{00} \qquad (2.5)$$

The coefficients B_{ij} are called *loss coefficients* or B-coefficients. 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. There are various ways of arriving at a loss equation.

2.1.5 ELD Formulation

The economic dispatching problem is to minimize the overall generating cost which is the function of plant output given by:

$$C_i (P_i (t)) = \sum a_i P_i^2 + b_i P_i + c_i \qquad (i=1,2,...,N_g) \qquad (2.6)$$

Subject to the constraints that generation should equal total demand plus losses, i.e.,

$$\sum P_i = P_D + P_L \qquad (2.7)$$

Satisfying the inequality constraints, expressed as follows:

$$P_{i(max)} \leq P_i \leq P_{i(min)} \quad (i=1,2,\ldots\ldots,n_g) \qquad (2.8)$$

where $P_{i(min)}$ and $P_{i(max)}$ are the minimum and maximum generating limits, respectively, for plant *i*.

2.2 Classical methods to solve ELD

There have been many algorithms proposed for economic dispatch: Merit Order Loading, Range Elimination, Binary Section, Secant Section, Graphical/Table Look-Up, Convex Simplex, Dantzig-Wolf Decomposition, Separable Convex Linear Programming, Reduced Gradient with Linear Constraints, Steepest Descent Gradient, First Order Gradient, Merit Order Reduced Gradient etc. The close similarity of the above techniques can be shown if the solution steps are compared. Some of the methods are discussed below briefly to minimize the fuel cost:

1. Lambda iteration method:

In this method, incremental cost curve for all the units is sketched and operating point of all the units is found where all the units have minimum fuel cost and at the same time specified demand is obtained. So in this method, incremental cost (λ) is assumed to find the power outputs of all the plants. This method converges fast but is slightly difficult.

2. Gradient method:

It works on the principle of minimization of function. But disadvantage of this method is lack of guarantee that the new points found at each step will lie on same surface.

3. Analytical approach:

In this technique, a mathematical representation of the cost-per-output characteristics is utilized to determine the optimal operating points for each generating station. Adjustments are then made at each station in order to bring the output to meet consumer demand.

4. Monte Carlo:

One of many other techniques is a form of Monte Carlo simulation where randomly allocated loads are given to stations until an acceptably low operating cost is discovered. This technique can utilize tables of recorded reference data to form a more realistic model of the generators, but it is very slow since it does not track its own progress, and is very inefficient since it simply guesses at solutions rather than optimizing its next guess by basing it on previous solutions.

2.3. ELD solution using Langrangian Multiplier Method

The algorithm used for this method requires derivative information. In many real-world problems, it is difficult to obtain the information about derivatives, either due to nature of the problem or due to the computations involved in calculating the derivatives. Despite these difficulties, gradient-based methods are popular and are often found to be effective. However, it is recommended to use these algorithms in problems where the derivative information is available or can be calculated easily. The optimality property that at a local or a global optimum the gradient is zero can be used to terminate the search process. The classical economic dispatch problem is an optimization problem that determines the power output of each online generator that will result in a least cost system operating state [56, 51].

As we know, the objective of economic load dispatch problem is to minimize cost of power generation in such a way that the cost should be minimized while satisfying various constraints. This power allocation is done considering system balance between generation and loads.

Problem Formulation:

The ELD problem can be formulated by first specifying the objective function of the problems. The objective function can be characterized by:-

Minimize $C_t = \sum C_i(P_i(t))$

$$= \sum a_i P_i^2 + b_i P_i + c_i \qquad (i=1,2,\ldots\ldots,n_g) \qquad (2.9)$$

Subjected to the constraint

When losses are neglected

$$\sum P_i = P_D \qquad (i=1,2,\ldots\ldots,n_g) \qquad (2.10)$$

When losses are included

$$\sum P_i = P_D + P_L \qquad (i=1,2,\ldots\ldots,n_g) \qquad (2.11)$$

While

$$P_{i(max)} \leq P_i \leq P_{i(min)} \qquad (i=1,2,\ldots\ldots,n_g) \qquad (2.12)$$

where

C_t = total operating cost

$C_i(P_i(t))$ = individual gen. production cost in terms of real power output Pi at time t

a_i , b_i , c_i are cost coefficients

P_D is the total load demand.

One of the most important, simple but approximate methods of expressing transmission loss as a function of generated powers is through B-coefficients.

This method uses the fact that under normal operating condition, the transmission loss is quadratic in the injected bus real powers. The general form of the loss formula using B-coefficients is: $P_L = \sum\sum P_i B_{ij} P_j$ MW

Where

P_i , P_j are real power injections at the ith, jth buses.

B_{ij} are loss coefficients which are constants under certain assumed conditions**.** The above loss formula is known as George's Formula. The above constraint optimization problem is converted into an unconstraint one**[33].** Lagrangian Multiplier method is used in which a function is minimized subject to side conditions in the form of equality constraints. Using Lagrangian Multipliers, an augmented function is defined as:

$$L = C_t + \lambda (P_D + P_L - \sum P_i) \qquad (2.13)$$

where λ is the Lagrangian Multiplier.

Necessary conditions for the optimization problem are:

$$\partial L / \partial Pi = \partial C_t / \partial Pi + \lambda(\partial P_L / \partial P_i - 1) = 0$$

Rearranging the above equation,

$$\partial C_t / \partial Pi = \lambda(1 - \partial P_L / \partial P_i)$$

This equation is known as exact coordination equation.

where $\partial C_t / \partial Pi$ is the incremental cost of the i^{th} generator (Rs/MWh)

$\partial P_L / \partial Pi$ represent the incremental transmission losses.

and

$$\partial L / \partial \lambda = P_D + P_L - \sum P_i = 0 \qquad (2.14)$$

By differentiating the transmission loss equation w.r.t. P_i , the incremental transmission loss can be obtained as:

$$\partial P_L / \partial Pi = \sum 2B_{ij}P_j \qquad (2.15)$$

and by differentiating the cost function w.r.t. P_i, the incremental cost can be obtained as

$$\partial C_t/\partial Pi = 2a_iP_i + b_i \quad (2.16)$$

when losses are considered, the resulting equations would be non-linear and must be solved iteratively.

The Flowchart for solving problem with Lagrangian Multiplier Method is shown in fig.2.2

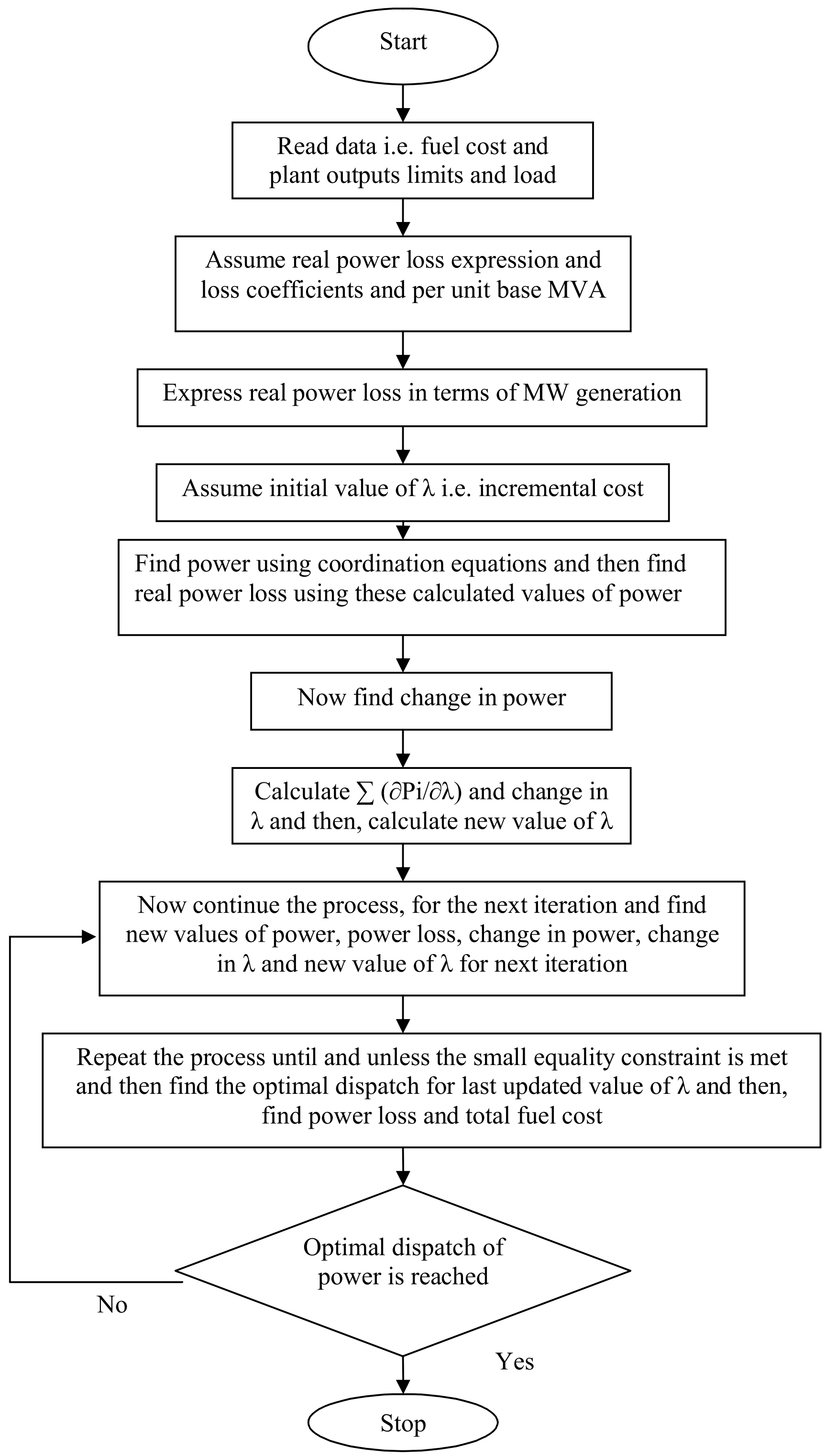

FIGURE (2.2) Flowchart for solving problem with Classical method.

CHAPTER-3

ELD Solution using Evolutionary Algorithms

In this chapter, the Evolutionary Algorithms particularly the Genetic Algorithm and Evolutionary Programming are briefly reviewed. And the formulation of ELD using GA and EP is discussed.

3.1. Evolutionary Algorithms (EAs):

Recently, EAs, which are probabilistic optimum search methods using genetics and evolution theory has been widely used. These are stochastic search methods that have been applied successfully in many search, optimization, and machine learning problems In real-world problems, EAs can implement algorithms without complex procedures and don't require lots of modification in software to handle additional constraints. Because the search process is parallel, there is a high probability of finding optimal solutions. EAs also use objective function information, not derivatives or other auxiliary knowledge **[41,45]**. A variety of evolutionary algorithms have been proposed in literature. The major ones are:

- Genetic algorithms (GA),
- Evolutionary programming (EP),
- Evolution strategies (ES),
- Classifier systems (CS), and
- Genetic programming(GP).

The General scheme of an Evolutionary Algorithm can be shown as shown in fig.(3.1)

Important features of EAs found suitable for power dispatch are:

Suitability: Inequality constraints on feature values can be implemented in the initial stage (normalization), which significantly reduces the computational complexity in handling large and complex systems.

Global Solution: The advantage of evolutionary algorithms over other methods for function optimization is that former provides a robust global solution, while the latter may get stuck at some local optimum. However, the global solution is dependent on several factors like population size, string length and weight coefficient.

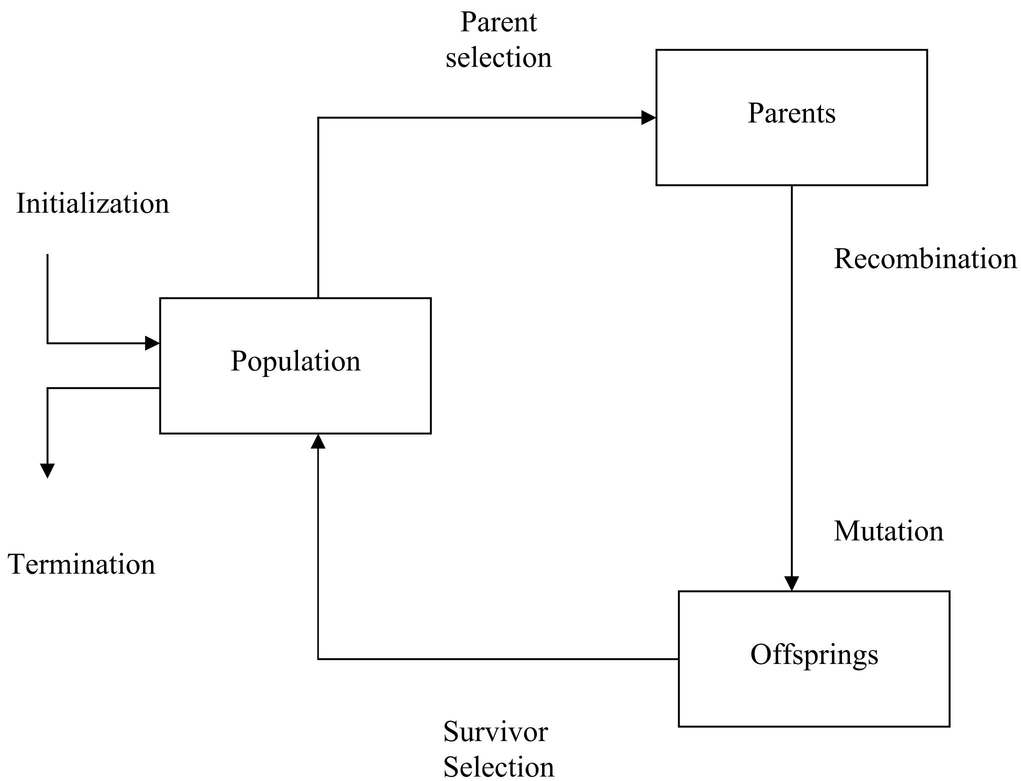

FIGURE (3.1). General scheme of an Evolutionary Algorithm

Flexibility: Different types of objective functions and relations (equality constraints) can be included into the combined objective function. Since the EAs optimizes different objective functions, in an independent manner, multi-objective function optimization can be easily achieved. EAs can be designed to operate in parallel, so that a large search space can be efficiently handled in making them suitable for on-line applications. Genetic algorithms, with suitable problem formulation, and representation, can be effectively and efficiently employed for optimization of complex power system problems. Research is in progress in exploiting their potential for other related problems in power systems.

3.2. Genetic Algorithm (GA)

A global optimization technique known as genetic algorithm has emerged as a candidate due to its flexibility and efficiency for many optimization applications. It is a stochastic searching algorithm. The method was developed by John Holland (1975). GAs are categorized as global search heuristics. These are particular class of evolutionary algorithms that use techniques inspired by genetics such as inheritance, mutation, selection and crossover **[14].**

GAs are search algorithms based on mechanics of natural selection and natural genetics. They combine Darwin's theory of 'survival of the fittest' among string structures with a structured yet randomized information exchange to form a search algorithm with some of the innovative flair of human search. In every generation, a new set of artificial creatures (strings) is created using bits and pieces of the fittest of the old; an occasional new part is tried for good measure. Even though randomized, genetic algorithms are not simple walk algorithms. They efficiently exploit historical information to speculate on new search points with expected improved performance. The individuals in the population then go through a process of evolution.

The advantages of the genetic algorithmic approach in terms of problem reduction, flexibility and solution methodology are also discussed. Some of the advantages and disadvantages with assumption are discussed which are used for solving our problem.

<u>Advantages of GA:</u>

Advantages of GA's are given below as discussed in **[7, 23].**

- 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 with numerically generated data, experimental data, or analytical encoded variables.
- Works with numerically generated data, experimental data, or analytical functions. Therefore, 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, GAs are used for a number of different application areas. In power system, the GAs have been used in following areas:

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

Disadvantages of GA

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

- Longer computation time.
- Less guaranteed convergence, particularly in case of epistatic objective function containing highly correlated parameters **[17, 18].**
- Premature convergence of GA is accompanied by a very high probability of entrapment into the local optimum **[19].**
- 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**[19].**
- Have to discretize parameter space**[17,18].**

When to use to a GA:

- Alternate solutions are too slow or overly complicated

- Need an exploratory tool to examine new approaches
- Problem is similar to one that has already been successfully solved by using a GA
- Want to hybridize with an existing solution
- Benefits of the GA technology meet key problem requirements

The process of GA follows this pattern:

1. An initial population of a random solution is created.
2. Each member of the population is assigned a fitness value based on its evaluation against the current problem.
3. Solution with highest fitness value is most likely to parent new solutions during reproduction.
4. The new solution set replaces the old, a generation is completed and the process continues at step (2).

Members of the population (chromosomes) are represented by a string of genes. Each gene represents a design variable and is symbolized by a binary number. Then, GA operators are used which are:

- reproduction,
- crossover and
- mutation.

GA allows a population composed of many individuals to evolve under specified selection rules to a state that maximizes the "fitness" (i.e., minimizes the cost function)**[34,35].** The different steps to perform GA are explained with the help of a **simplified flowchart** as shown below in fig 3.2

GA work iteratively sustaining a set (population) of representative chromosomes of possible solutions to the problem domain at hand. As an optimization method, they evaluate and manipulate these chromosomes using stochastic evolution rules called Genetic Operators. During each iterative step, known as a generation, the representative chromosomes in the current population are evaluated for their fitness as optimal solutions. By comparing these fitness values, a new population of solution chromosomes is created using the genetic operators, known as reproduction, crossover and mutation. **[26]**. There are five components that are needed to implement a Genetic Algorithm.

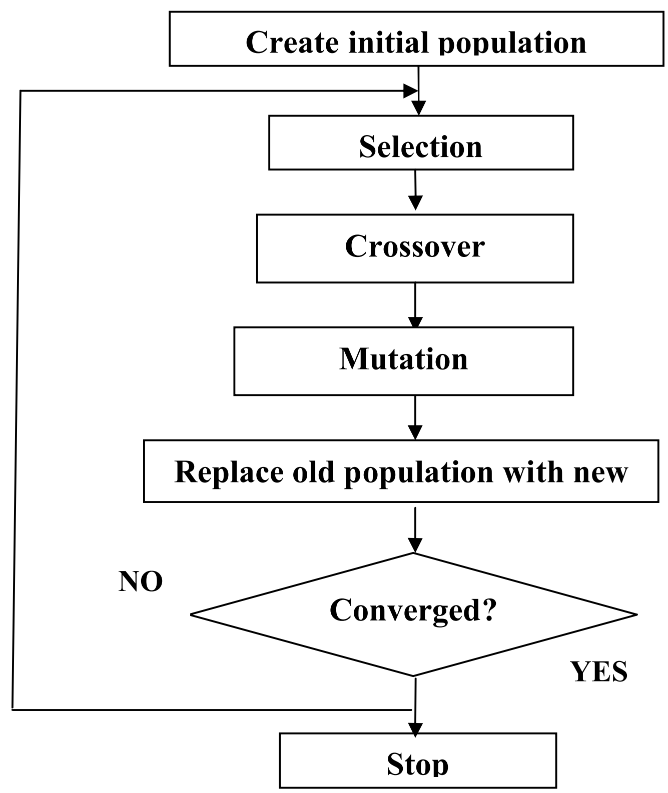

FIGURE 3.2: Simplified Flowchart of a Genetic Algorithm

These five components listed below and are detailed in the following sections:

1. Representation
2. Initialization
3. Evaluation Function
4. Genetic Operators, and
5. Genetic Parameters
6. Termination

3.2.1. Representation:

Genetic Algorithms are derived from a study of biological systems. In biological systems evolution takes place on organic devices used to encode the structure of living beings. These organic devices are known as chromosomes. A living being is only a decoded structure of the chromosomes. Natural selection is the link between chromosomes and the performance of their decoded structures. In GA, the design variables or features that characterize an individual are represented in an ordered list called a string. Each design variable corresponds to a gene and the string of genes corresponds to a chromosome.

Encoding:

The application of a genetic algorithm to a problem starts with the encoding. The encoding specifies a mapping that transforms a possible solution to the problem into a structure containing a collection of decision variables that are relevant to the problem. A particular solution to the problem can then be represented by a specific assignment of values to the decision variables. The set of all possible solutions is called the search space and a particular solution represents a point in that search space. In practice, these structures can be represented in various forms, including among others, strings, trees, and graphs. There are also a variety of possible values that can be assigned to the decision variables, including binary, k-array, and permutation values.

Therefore, in order to implement GA for finding the solution of given optimization problem, variables are first coded in some structure. The strings are coded by binary representations having 0's and 1's. The string in GA corresponds to "chromosomes" and

For power dispatch problems, firstly population of random numbers of 0's and 1's has been generated. The length of each string in this study has been assumed as 16. The population size has been taken as 20.

Decoding:

Decoding is the process of conversion of the binary structure of the chromosomes into decimal equivalents of the feature values. Usually this process is done after de-catenation of the entire chromosome to individual chromosomes. The decoded feature values are used to compute the problem characteristics like the objective function, fitness values, constraint violation and system statistical characteristics like variance, standard deviation and rate of convergence. The stages of selection, crossover, mutation etc are repeated till some termination condition is reached. There are several ways of selecting the termination conditions, which can be either the convergence of the total objective function or the satisfaction of the equality constraint or both. Since the genetic algorithm determines the above features independently, the satisfaction of both the conditions has to be considered for total absolute convergence. However, in situations of constraint violation, independent satisfaction of the above conditions have to be considered and in the order of occurrence to decide the feasibility of the solution.

String representation:

GA works on a population of strings consisting of a generation. A string consists of sub-strings, each representing a problem variable. In the present ELD problem, the problem variables correspond to the power generations of the units. Each string represents a possible solution and is made of sub-strings, each corresponding to a generating unit. The length of each sub-string is decided based on the maximum/minimum limits on the power generation of the unit it represents and the solution accuracy desired. The string length, which depends upon the length of each sub-string, is chosen based on a trade-off between solution accuracy and solution time. Longer strings may provide better accuracy, but result in higher solution time.

3.2.2. Initialization:

Initially many individual solutions are randomly generated to form an initial population. The population size depends on the nature of the problem, but typically contains several hundreds or thousands of possible solutions. Traditionally, the population is generated randomly, covering the entire range of possible solutions (the search space). Occasionally, the solutions may be "seeded" in areas where optimal solutions are likely to be found **[9, 10, 11].**

3.2.3. Evaluation Function:

The evaluation function is a procedure for establishing the fitness of each chromosome in the population and is very much application orientated. Since Genetic Algorithms proceed in the direction of evolving the fittest chromosomes, and the performance is highly sensitive to the fitness values. In the case of optimization routines, the fitness is the value of the objective function to be optimized. Penalty functions can also be incorporated into the objective function, in order to achieve a constrained problem **[26].**

Fitness function:

The Genetic algorithm is based on Darwin's principle that "The candidates, which can survive, will live, others would die". This principal is used to find fitness value of the process for solving maximization problems. Minimization problems are usually transferred into maximization problems using some suitable transformations. Fitness value $f(x)$ is derived from the objective function and is used in successive genetic operations. The fitness function for maximization problem can be used the same as objective function $F(X)$.

Coming up with an encoding is the first thing in genetic algorithm user has to do. The next step is to specify a function that can assign a score to any possible solution or structure. The score is a numerical value that indicates how well the particular solution solves the problem. Using a biological metaphor, the score is the fitness of the individual solution. It represents how well the individual adapts to the environment. In case of optimization, the environment is the search space. The task of the GAs is to discover solutions that have fitness values among the set of all possible solutions.

In general, a fitness function F(x) is first derived from the objective function and used in successive genetic operations. Certain genetic operators require that the fitness function be non-negative. For maximization problems, the fitness function be considered to be the same as objective function or

$$F(X) = f(X) \tag{3.1}$$

For minimization problems, the fitness function is an equivalent maximization problem chosen such that the optimum point remains unchanged. The following fitness function is often used in minimization problems:

$$F(x) = 1/(1+f(X)) \tag{3.2}$$

This information does not alter the location of the minimum, but converts a minimization problem to an equivalent maximization problem. The fitness function value of a string is known as the string's fitness.

The operation of GAs begins with a population of random strings representing design or decision variables. Thereafter, each string is evaluated to find the fitness value. The population is then operated by three operators- reproduction, crossover, and mutation to create a new population of points. The new population is further evaluated and tested for termination. If the termination criteria is not met, the population is iteratively operated by the above three operators and evaluated. This procedure is continued until the termination criterion is met. One cycle of these operations and the subsequent evaluation procedure is known as a generation in GA's terminology.

Implementation of power dispatch problem in GA is realized within the fitness function written in eqn.(3.2)

3.2.4. Genetic Operators:

Genetic operators are a set of random transition rules employed by a Genetic Algorithm. These operators are applied to a randomly chosen set of chromosomes during each generation, to produce a new and improved population from the old one. A simple GA consists of three basic operators:

- reproduction,
- crossover,
- mutation.

Reproduction:

The Reproduction is the straightforward copying of an individual to the next generation, otherwise known as Darwinian or asexual reproduction. Reproduction is usually first operator applied on a population. Reproduction selects good strings in a population and forms a mating pool. That is why the reproduction operator is sometimes known as the selection operator. There exist a number of reproduction operators in GA literature, but essential idea in all of them is that the above average strings are picked from the current population and their multiple copies are inserted in the mating pool in a probabilistic manner. The commonly used reproduction operator is the proportionate reproduction operator where a string is selected for the mating pool with a probability proportional to its fitness. Thus, the ith string in the population is selected with a probability proportional to fitness F_i. Since the population size is usually kept fixed in a simple GA, the sum of the probability of each string being selected for the mating pool must be one. Therefore, the probability for selecting the ith string is:

$$P_i = F_i / \sum F_i \qquad (3.3)$$

where i=1,2,…,n and where n is the population size.

One way to implement this selection scheme is to imagine a roulette-wheel with its circumference marked for each string proportionate to the string's fitness. The roulette-wheel is spun n times, each time selecting an instance of the string chosen by a roulette-wheel (RW) pointer. Since the circumference of the wheel is marked according to a string's fitness, this roulette-wheel mechanism is expected to make f_i / f_{av} copies of the ith string in the mating pool. The average fitness of the population is calculated as:

$$f_{av} = \left(\sum_{i=1}^{n} f_i \right) \times \frac{1}{n} \qquad (3.4)$$

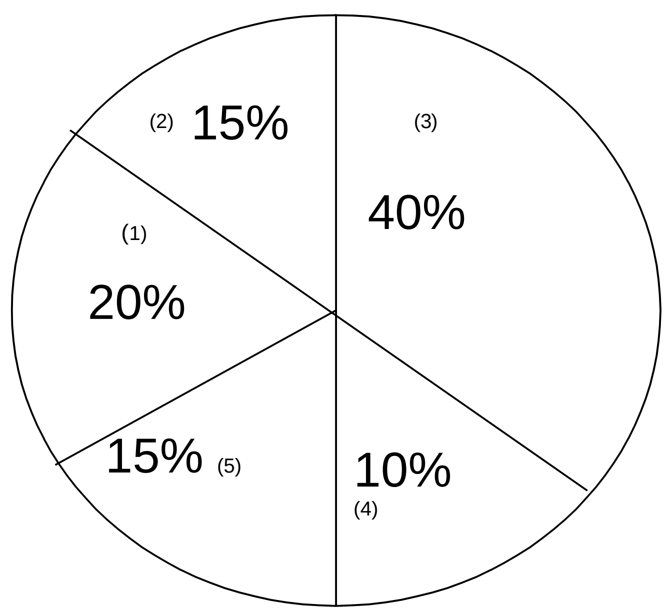

FIGURE (3.3) Roulette Wheel selection

Since, the third individual has a higher fitness value than any other, it is expected that the RW selection will choose the third individual more than any other individual. This roulette-wheel selection scheme can be simulated easily. Using the fitness value F_i of all the strings, the probability of selecting a string p_i can be calculated. Thereafter the cumulative probability (P_i) of each string being copied can be calculated by adding the individual probabilities from the top of the list. Thus, the bottom-most string in the population should have a cumulative probability (P_n) equal to one. The roulette-wheel concept can be simulated by realizing that the i^{th} string in the population represents the cumulative probability values from P_{i-1} to P_i. The first string represents the cumulative value from zero to P_1. Thus, the cumulative probability of any string lies between 0 to 1. In order to choose n strings, n random numbers between zeros to one are created at random. Thus, a string that represents the chosen random number in the cumulative range (calculated from the fitness value) for the string is copied to the mating pool. This way, the string with a higher fitness value will represent a larger range in the cumulative probability values and therefore has a higher probability of being copied into the mating pool. On the other hand, a string with a smaller fitness value will represent a smaller range in the cumulative probability values and has a smaller probability of being copied into the mating pool.

Crossover:

The basic operator for producing new chromosome in the genetic algorithm is crossover. In the crossover operator, information is exchanged among strings of the mating pool to create new strings. In other words, crossover produces new individuals that have some parts of both parent's genetic materials. It consists of taking two individuals A and B and randomly selecting a crossover point in each. The two individuals are then split at these points. The choice of crossover point is not always uniform.

It is expected from the crossover operator that good substrings from the parent strings will be combined to form a better child offspring. At the molecular level what occurs is that a pair of Chromosomes bump into one another, exchange chunks of genetic information and drift apart. This is the recombination operation, which GA generally refers to as crossover because of the way that genetic material crosses over from one chromosome to another. The crossover operation happens in an environment, where the selection of who gets to mate is a function of the fitness of the individuals. How good the individual is at competing in its environment.

Some Genetic Algorithms use a simple function of the fitness measure to select individuals (probabilistically) to undergo genetic operations such as crossover or asexual reproduction (the propagation of genetic material unaltered). This is fitness-proportionate selection. Other implementations use a model in which certain randomly selected individuals in a subgroup compete and the fittest is selected. This is called tournament selection and is the forms of selection we see in nature .The two processes that most contribute to evolution are crossover and fitness based selection/reproduction.

There are three forms of crossover: ***(1) one point crossover,***
(2) multipoint crossover, and
(3) uniform crossover.

During the implementation of GA, one point crossover is used.

One 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 (exchanged) between the two strings at the crossover site. Suppose site **3** is selected at random. It means starting from the 4^{th} bit and onwards, bits of strings will be swapped to produce offspring which are given below:

Parent 1: x_1 = {010 **1101011**}

Parent 2: x_2 = {100 **0011100**}

Offspring 1: x_1 = {010 **0011100**}

Offspring 2: x_2 = {100 **1101011**}

Mutation:

Mutation also plays a role in this process, although how important its role is, depends upon the conditions. It is also known as background operator .It plays dominant role in the evolutionary process. It cannot be stressed too strongly that the genetic Algorithm is not a random search for a solution to a problem for highly fit individual. It consists of randomly selecting a mutation point.

The genetic algorithm uses stochastic processes, but the result is distinctly non-random. Genetic Algorithms are used for a number of different applications areas. An example of this would be multidimensional optimization problems in which the character string of the Chromosome can be used to encode the values for the different parameters being optimized.

Mutation is an important operator, as newly created individuals have no new inheritance information, this process results in contraction of the population at one single point, which is wished one. Mutation operator changes 1 to 0 at only one place in the whole string with a small probability and vice versa.

e.g. Child 1 1011***0***0

Let mutation is done at location 5 the new child will be

New child 1011***1***0

In general, the mutation probability is fixed through out the whole process. However a small mutation probability results in small premature convergence but the search with large fixed mutation probability will not converge a lot so this operator is seldom used in the process.

3.2.5. Genetic Parameters:

Genetic parameters are a means of manipulating the performance of a Genetic Algorithm. There are many possible implementations of Genetic Algorithms involving variations such as additional genetic operators, variable sized populations and so forth. Listed below are some of the basic genetic parameters:

(i) Population Size (N)

(ii) Crossover rate (C)

(iii) Mutation rate (M)

(i). Population Size (N): Population size affects the efficiency and performance of the algorithm. Using a small population size may result in a poor performance from the algorithm. This is due to the process not covering the entire problem space. A larger population on the other hand, would cover more space and prevent premature convergence to local minima. At the same time, a large population needs more evaluations per generation and may slow down the convergence rate.

(ii). Crossover rate (C): The crossover rate is the parameter that affects the rate at which the process of crossover is applied. In each new population, the number of strings that undergo the process of crossover can be depicted by a chosen probability. This probability is known as the crossover rate. A higher crossover rate introduces new strings more quickly into the population. If the crossover rate is too high, high-performance strings are eliminated faster than selection can produce improvements. A low crossover rate may cause stagnations due to the lower exploration rate, and convergence problems may occur.

(iii). Mutation rate (M): Mutation rate is the probability with which each bit position of each chromosome in the new population undergoes a random change after the selection process. It is basically a secondary search operator which increases the diversity of the population. A low mutation rate helps to prevent any bit position from getting trapped at a single value, whereas a high mutation rate can result in essentially random search.

3.26. Termination:

This generational process is repeated until a termination condition has been reached. Common terminating conditions are:

1. A solution is found that satisfies minimum criteria
2. Fixed number of generations reached
3. Allocated budget (computation time/money) reached
4. The highest ranking solution's fitness is reaching or has reached a plateau such that successive iterations no longer produce better results
5. Manual inspection
6. Combinations of the above.

The termination determines the convergence of the optimization process to achieve the optimal solution. The convergence criterion is given in the equation. If the convergence criterion is not achieved, the whole process will repeat **[9].**

$$\text{Fitness}_{max} - \text{Fitness}_{min} \leq 0.0001 \qquad (3.5)$$

3.3. Evolutionary Programming

Evolutionary programming is based on Finite State Machine (FSM) model in its early stage, which is proposed by L. **J.** Fogel in 1960's when he studied the artificial intelligence. In **1990's,** the evolutionary programming was developed by D. **B.** Fogel and was made to solve the optimal problems in real space. The evolutionary programming bas been become an optimal tool and was used in many practical problems. EP is a artficial intelligence method in which an optimisation algorithm is the main engine for the process of three steps, namely, natural selection, mutation and competition. According to the problem, each step could be modified and configured in order to achieve the optimum result. Each possible solution to the problem is called an individual. In order to use EP, the mathematical model should be capable of dealing with the data type and structure of individuals. The EP algorithm has four main stages:

Initialisation*:* The initial population consists of individuals (sections) and is created randomly. The fitness score f_i of each P_i, is obtained by a fitness function.

Statistics*:* The maximum fitness, minimum fitness, sum of fitness and average fitness of this generation are calculated.

Mutation*;* Each *Pl* is mutated in order to generate a new population.

Competition*:* Each individual P_i in the combined population has to compete with some other individuals to get its chance to be transcribed to the next generation.

The steps for EP are same as GA except that EP uses only mutation operator. It does not use the crossover operator. Rest all other steps are same. So, we are only formulated GA because the formulation for EP is also same except one step of crossover operator which is missing. Hence, EP is mutation operator based Process i.e. one step less as compare to GA**.**

3.4. ELD Formulation using GA

The ELD problem can be formulated by first specifying the objective function of the problems. The objective function can be characterized by:-

$$\text{Minimize } C_t = \sum C_i(P_i(t))$$

$$= \sum a_i P_i^2 + b_i P_i + c_i \qquad (i=1,2,\ldots\ldots,n_g) \qquad (3.6)$$

Subjected to the constraint

When losses are neglected

$$\sum P_i = P_D \qquad (i=1,2,\ldots\ldots,n_g) \qquad (3.7)$$

When losses are included

$$\sum P_i = P_D + P_L \qquad (i=1,2,\ldots\ldots,n_g) \qquad (3.8)$$

While

$$P_{i(max)} \leq P_i \leq P_{i(min)} \quad (i=1,2,\ldots\ldots,n_g) \qquad (3.9)$$

where

C_t = total operating cost

$C_i(P_i(t))$ = individual gen. production cost in terms of real power output Pi at time t

a_i , b_i , c_i are cost coefficients

P_D is the total load demand.

One of the most important, simple but approximate methods of expressing transmission loss as a function of generated powers is through B-coefficients.

This method uses the fact that under normal operating condition, the transmission loss is quadratic in the injected bus real powers. The general form of the loss formula using B-coefficients is:

$$P_L = \sum\sum P_i B_{ij} P_j \text{ MW}$$

Where

P_i , P_j are real power injections at the ith, jth buses.

B_{ij} are loss coefficients which are constants under certain assumed conditions**.** The above loss formula is known as George's Formula. The above constraint optimization problem is converted into an unconstraint one**[33].** Lagrangian

Multiplier method is used in which a function is minimized subject to side conditions in the form of equality constraints. Using Lagrangian Multipliers, an augmented function is defined as:

$$L = C_t + \lambda (P_D + P_L - \sum P_i) \qquad (3.10)$$

where λ is the Lagrangian Multiplier.

Necessary conditions for the optimization problem are:

$$\partial L / \partial Pi = \partial C_t / \partial Pi + \lambda(\partial P_L / \partial P_i - 1) = 0 \qquad (3.11)$$

Rearranging the above equation,

$$\partial C_t / \partial Pi = \lambda(1 - \partial P_L / \partial P_i) \qquad (3.12)$$

This equation is known as exact coordination equation.

where $\partial C_t / \partial Pi$ is the incremental cost of the i^{th} generator (Rs/MWh)

$\partial P_L / \partial Pi$ represent the incremental transmission losses.

and

$$\partial L / \partial \lambda = P_D + P_L - \sum P_i = 0 \qquad (3.13)$$

By differentiating the transmission loss equation w.r.t. P_i , the incremental transmission loss can be obtained as:

$$\partial P_L / \partial Pi = \sum 2B_{ij}P_j \qquad (3.14)$$

and by differentiating the cost function w.r.t. P_i, the incremental cost can be obtained as

$$\partial C_t / \partial Pi = 2a_iP_i + b_i \qquad (3.15)$$

To find the solution, we obtain

$$2a_iP_i + b_i = \lambda(1 - \sum 2B_{ij}P_j) \qquad (3.16)$$

Rearranging the above equation to get P_i i.e.

$$2a_iP_i + b_i = \lambda(1 - 2B_{ii}P_i - \sum 2B_{ij}P_j) \qquad (3.17)$$

or

$$2(a_i + \lambda B_{ii}) P_i + \lambda \sum 2B_{ij}P_j = \lambda - b_i \qquad (3.18)$$

where i=1,2 and 3

or

where λ is incremental cost. Now, the incremental cost is searched using GA.

The GA can be implemented by searching the generation of power plants P_i within the generation limits given by above equation (3.8). The ELD problem is discussed in chapter-2, where transmission losses to be considered are also discussed.

There are further two methods using EA to solve the ELD problem:

- First is using real power method, in which search is directly made on power of each unit.
- Second method is to search lambda to minimize the fuel cost.

In this thesis, second method of lambda is used to minimize the fuel cost.

While solving the problem using GA, the length of each GA string in the population has been taken as **16** and the population size has been assumed as 20. Therefore, population we have taken is **20 x 16** in binary. The crossover and mutation probabilities are taken **0.5** and **0.1** selection rate is **50%.** The crossover points are chosen randomly. The population considered represents the power generations separately for every generator; therefore population of every variable would be generated.

In order to simulate the proposed GA to solve ELD, a set of steps were produced. These set of steps and its implementation are shown below, and outlined how each of the different sections of GA were simulated to produce the results and the simulated results are also shown.

Step 1). Initialization or creating the initial population:

The initial population is generated in the form of random numbers. The binary numbers are produced by using *rand* command. The binary representation used for random numbers is shown below as binary strings. Let 20 binary numbers were generated of population size 16 i.e. population of 20 x 16.

Step 2). Find incremental cost λ:

The minimum and maximum limits for lambda are considered as 10 and 12.5 and lambda is calculated using formula:

$$\lambda = \lambda min + ((\lambda max - \lambda min) / (2^l - 1)) * Y, \qquad (3.19)$$

where Y represents the decimal number and *l* represents string length.

Decoding:

Decoding of string can be done using equation $Y^j = \sum 2^{i-1} b_i^j$ (j=1,2,…,L ; i=1,2,…,*l*)

Let example, the first member of population is

0	1	1	0	1	1	1	1	0	0	0	1	1	1	1	1

$Y^1 = 0 \times 2^0 + 1 \times 2^1 + 1 \times 2^2 + 0 \times 2^3 + 1 \times 2^4 + 1 \times 2^5 + 1 \times 2^6 + 1 \times 2^7 + 0 \times 2^8 +$

$0 \times 2^9 + 0 \times 2^{10} + 1 \times 2^{11} + 1 \times 2^{12} + 1 \times 2^{13} + 1 \times 2^{14} + 1 \times 2^{15}$

$= 1 + 2^1 + 2^2 + 2^4 + 2^5 + 2^6 + 2^7 + 2^{11} + 2^{12} + 2^{13} + 2^{14} + 2^{15} = 63735$

Using this, λ^1 can be calculated as:

$$\lambda = \lambda^{min} + ((\lambda^{max} - \lambda^{min}) / (2^l - 1)) * Y^1$$

$$= 10.0 + ((12.5 - 10.0) / (2^{16} - 1)) * 63735$$

$$= 12.34 \text{ Rs/MWh}$$

Step 3). Find P_i's:

P_i's are calculated using incremental cost. Before calculating P_i's, power generation limits has to be set and the calculated values of P_i's should be within these limits such that these should satisfied the load demand. And total power is sum of all three powers TP = P1 + P2 + P3, where P_1, P_2,P_3 can be calculated using following equation:

$$\begin{bmatrix} 2(a_1 + \lambda B_{11}) & 2\lambda B_{12} & 2\lambda B_{13} \\ 2\lambda B_{21} & 2(a_2 + \lambda B_{22}) & 2\lambda B_{23} \\ 2\lambda B_{31} & 2\lambda B_{32} & 2(a_3 + \lambda B_{33}) \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} = \begin{bmatrix} \lambda - b_1 \\ \lambda - b_2 \\ \lambda - b_3 \end{bmatrix}$$

The above equation can be re-written as:

$$\sum A_{ik} P_k = C_i$$

where

$A_{ii} = 2(a_i + \lambda B_{ii})$

$A_{ik} = 2 \lambda B_{ik}$

$C_i = \lambda - b_i$

Step 4). Find cost, transmission losses and error:

Cost can be simply calculated using equation: $C = a_i P_i^2 + b_i P_i + c_i$

And total cost is the sum of all 3 costs C1, C2, C3.

And transmission losses can be calculated using:

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

Error can be calculated using equation:

$$\epsilon = | P_D + P_L - \sum P_i |$$

Step 5). Fitness Process:

We have to find the population with maximum fitness because the chromosome with highest fitness will be more suitable to survive and will give more better results.

The formula used to calculate the fitness function is given below:

$$F = 1 / ((1+ \text{error}) / PD))$$

$$\text{where error} = | PD + PL - \sum P_i |$$

Step 6). Reproduction Process:

6(a). Set selection rate and number of mating in a pool.

6(b). Define total fitness as a sum of values obtained by using above step for all chromosomes which are selected.

6(c). Select a percentage of the roulette wheel each chromosome which is equal to the ratio of its fitness value to the total fitness value. i.e. find probability which can be written as:

Probability = fitness / ∑ Fitnesses

6(d). Calculate cumulative sum (CS) to normalize the values on the Roulette wheel between 0.0 and 1.0.

Step 7). Crossover Process:

7(a). Choose a pair of random numbers between 0 and 1 to select one mother(ma) and one father(pa) chromosomes so as to produce new offsprings.

7(b). pairing the chromosomes from different location,for different locations, crossover point has to be selected which can be selected randomly, so a Generate offsrings by applying crossover.

Furthermore, crossover can be applied over single point, two-point and multipoint. Following example represents the single point crossover.

Parent 1: x_1 = {010 **1101011**}

Parent 2: x_2 = {100 **0011100**}

Offspring 1: x_1 = {010 **0011100**}

Offspring 2: x_2 = {100 **1101011**}

After crossover, the offsprings which are generated contains traits from mother and father chromosomes. This will add to the new information to the offsprings.

After applying crossover, again whole process is repeated and with new calculated values of P_i's(offsprings), again power losses, cost, fitness etc is calculated and checked whether the new solutions provide the better results or not.

Step 8). Mutation Process:

Perform the mutation. In this we have, we have to flip one bit. If its 0, then, flip it equal to 1 and vice-versa. But before that, mutation rate is to be defined.

Random mutations alter a certain percentage of the bits in the list of chromosomes. Mutation is the second way a GA explores a cost surface. It can introduce traits not in the original population and keeps the GA from converging too fast before sampling the entire cost surface. Mutation points are randomly selected from the total number of bits in the population matrix. Increasing the number of mutations increases the algorithm's freedom to search outside the current region of variable space.

Example of mutation process can be depicted as follows:

The following pairs were randomly selected:

mrow=[5 7 6 3 8 4 9]

mcol= [6 12 7 1 3 8 5]

The first random pair is (5, 6). Thus the bit in row 5 and column 6 of the population matrix is mutated from a 1 to a ***0***:

00101*1*00000001

↓

00101*0*00000001

Step 9). The next generation or updating Population:

After the mutations take place, the costs associated with the offspring and mutated chromosomes are calculated .The process described is iterated, replacing the old population or (initial population) with the new improved population generated after passing through the whole process of selection, crossover and mutation again.

Step 10). Convergence:

The number of generations that evolve depends on whether an acceptable solution is reached or a set number of iterations is exceeded. After a while all the chromosomes and associated costs would become the same if it were not for mutations. At this point, the algorithm should be stopped.

Flowchart of GA structure used for ELD

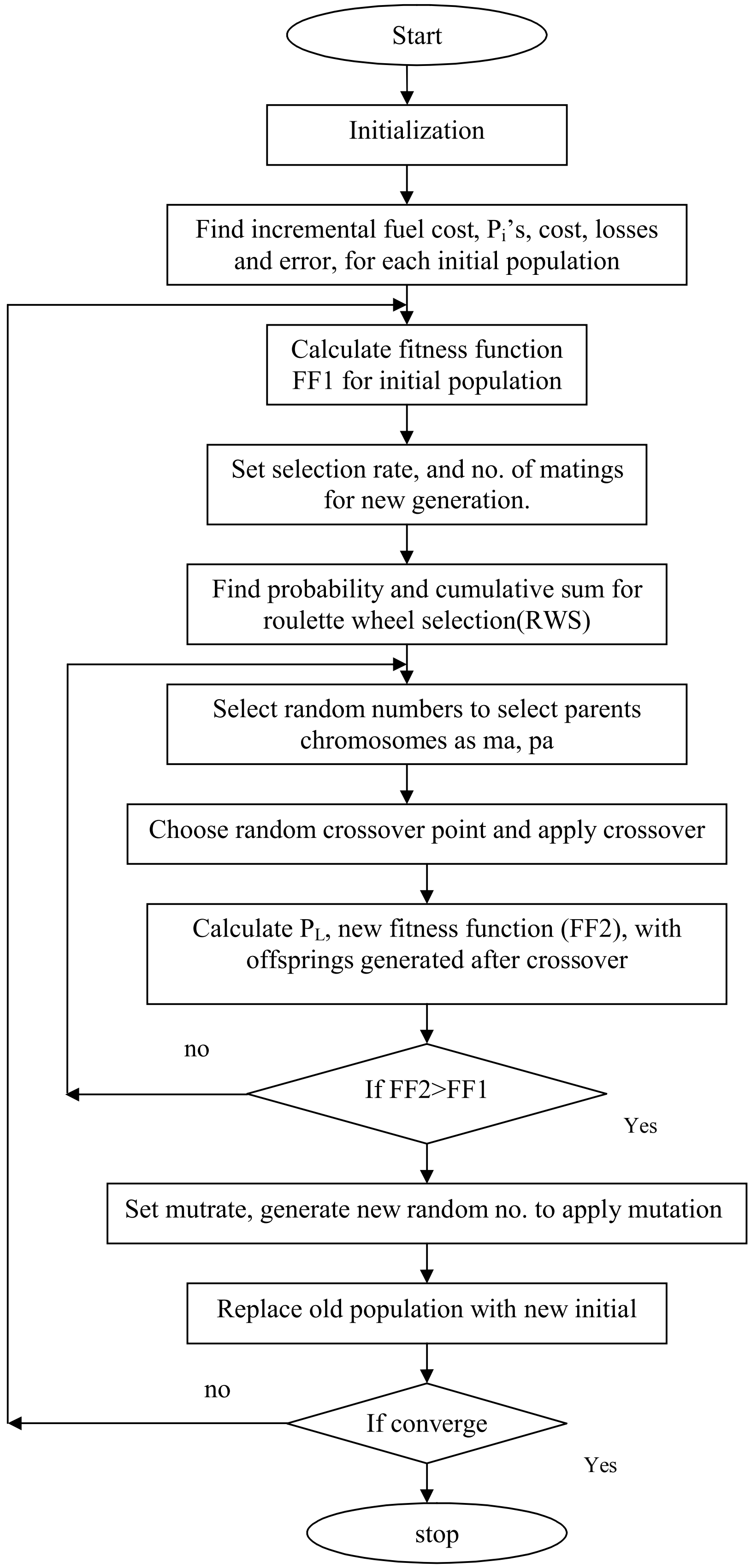

FIGURE 3.4. FLOWCHART OF GA USED FOR SOLVING ELD

3.5 ELD FORMULATION USING EP

EP is population based search method like GA. Only mutation operator is used to produce offsprings.

Therefore, the steps we performed above to solve problem with GA are similar for EP except that in case of EP, we are not applying crossover operator. EP is only based on mutation operator.

The detail of GA is summarised in above section i.e.3.4. Here, only the steps are summarised for reference.

To implement EP to ELD, following steps are used. In these steps, it can be seen that there would be only one difference between GA and EP and that is EP has only one operator i.e. mutation. No crossover operator is used in case of EP.

Following are the phases for EP:

Initialisation*:* The initial population consists of individuals (sections) and is created randomly. The fitness score f_i of each P_i, is obtained by a fitness function.

Statistics*:* The maximum fitness, minimum fitness, sum of fitness and average fitness of this generation are calculated.

Mutation*;* Each *Pl* is mutated in order to generate a new population.

Competition*:* Each individual P_i in the combined population has to compete with some other individuals to get its chance to be transcribed to the next generation.

Following steps are used for ELD formulation using EP:

Step 1). Initialization or creating the initial population.

Step 2). Find incremental cost λ

Step 3). Find P_i's

Step 4). Find cost, transmission losses and error:

Step 5). Fitness Process:

Step 6). Reproduction Process

Step 8). Mutation Process

Step 9). The next generation or updating Population

Step 10). Convergence:

CHAPTER-4

RESULTS AND DISCUSSIONS

In this section, the results of ELD after the implementation of proposed EA (GA and EP) are discussed and compared with the classical method. The algorithms are implemented in MATLAB to solve ELD problem. The main objective is to minimize the cost of generation of thermal plants using GA, EP and Classical LMM method. The performance is evaluated with and without losses for two set generator data, which are referred as Case I and Case II

Case I: Three generator test system **[57]**

Case II: Six generator test system **[43]**

4.1 CASE I - 3 GENERATOR TEST SYSTEM

The specifications of three generator test system are detailed in Table 4.1. The coefficients of fuel cost are given below in Tables 4.1(a). Transmission loss coefficients are given in Table 4.1(b). The power demand is considered to be 150MW. And Maximum and minimum power limits are given in Table 4.1(c).

The results corresponding to LMM, GA and EP are detailed in section 4.1.1, 4.1.2 and 4.1.3 respectively.

Unit no.	a_i	b_i	c_i
1	**0.008**	**7.0**	**200**
2	**0.009**	**6.3**	**180**
3	**0.007**	**6.8**	**140**

(a)

0.000179	**0.000278**	**0.000218**

(b)

Limits	Maximum	Minimum
P1	**85**	**10**
P2	**80**	**10**
P3	**70**	**10**

(c)

TABLE-4.1. Specifications of 3-generator test system

(a) Cost coefficients

(b) Loss coefficients

(c). Power generation limits

4.1.1 Optimum Solution Using LMM for Case-I

Developed program returns the system lambda, generated power P, the total cost and total losses. The simulated results for LMM method are shown below. Here we considered two cases LMM with losses and LMM without losses.

LMM	λ (Rs/MWh)	P1(MW)	P2(MW)	P3(MW)	TP(MW)	PL(MW)	Cost(Rs/hr)
with losses	7.678935	35.0907	64.1317	52.4797	151.719	1.6991	1592.62
without losses	7.675789	34.07	63.11	52.83	150.01	0	1590.79

Table 4.2. Results using LMM for three generator test system

For three generation test system, the system **λ** and the total cost of operation are lower when losses are neglected. Cost comes to Rs **1592.62/hr** when we considered losses and Rs **1590.79/hr** when considered without losses.

4.1.2 Optimum Solution Using GA for Case-I

The description of the results obtained by utilizing GA when losses are considered is detailed herewith. The length of each string in the population has been taken as 16 and the population size has been assumed as 20. Therefore, population we have taken is 20 x 16 in binary. The crossover and mutation probabilities are taken 0.5 and 0.1, selection rate is 50%. The crossover points are chosen randomly. The generated random population is shown in Table 4.3

In order to simulate the proposed GA to solve ELD, a set of steps were produced along with the results of first iteration are shown below

Initialization or creating the initial population:

The initial population is generated in the form of random numbers. The binary numbers are produced by using *rand* command. The binary representation used for random numbers is shown below as binary strings in the table 4.3. Let 20 binary numbers were generated of population size 16 i.e. population of **20 x 16**.

S.no	Binary string															
1	1	0	1	0	1	1	0	1	1	0	0	0	0	0	1	0
2	0	1	0	1	1	1	1	1	1	0	0	0	0	1	0	1
3	0	0	1	1	0	1	1	0	0	0	0	1	1	0	0	0
4	1	0	1	0	0	1	0	0	0	1	1	0	1	0	0	1
5	1	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0
6	0	0	1	0	1	1	0	1	0	0	0	0	0	1	1	0
7	0	1	0	1	1	0	1	1	1	1	1	1	1	1	1	0
8	1	1	0	0	0	0	1	1	0	0	1	1	0	1	0	1
9	0	1	0	1	0	0	1	1	1	1	0	0	0	0	1	0
10	1	1	0	1	0	1	0	0	1	1	0	0	1	0	0	0
11	0	0	1	1	0	1	1	0	0	0	0	0	0	1	0	1
12	0	1	0	1	1	1	1	1	1	0	0	1	1	0	0	0
13	1	1	0	1	1	1	1	1	1	0	0	0	0	0	1	0
14	0	1	0	1	0	0	1	1	1	1	0	0	0	1	0	1
15	0	1	0	0	1	1	0	1	0	0	0	0	0	1	0	1
16	0	1	0	1	1	1	1	1	1	0	0	0	0	0	1	0
17	0	1	0	1	1	1	1	1	1	0	0	0	0	1	1	0
18	0	0	1	0	1	1	0	1	0	0	0	0	0	1	0	1
19	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1
20	0	1	0	1	1	1	1	1	1	0	0	0	0	1	0	1

Table 4.3 Binary random number strings

Now, next step was to calculate incremental cost λ.

Find incremental cost λ:

The minimum and maximum limits for lambda are considered as 10 and 12.5 and lambda is calculated using formula:

$$\lambda=\lambda min + ((\lambda max - \lambda min) / (2^{l} - 1))* Y,$$

Where Y represents the decimal number and *l* represents string length.

Decoding:

Decoding of string can be done using equation $Y^{j} = \sum 2^{i-1} b_i^j$ (j=1,2,…,L ; i=1,2,…,*l*)

Let example, the first member of population is

1	0	1	0	1	1	0	1	1	0	0	0	0	0	1	0

$Y^1 = 1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 0 \times 2^3 + 1 \times 2^4 + 1 \times 2^5 + 0 \times 2^6 + 1 \times 2^7 + 1 \times 2^8 + 0 \times 2^9 + 0 \times 2^{10} + 0 \times 2^{11} + 0 \times 2^{12} + 0 \times 2^{13} + 1 \times 2^{14} + 0 \times 2^{15}$

$= 16821$

Using this, λ^1 can be calculated as:

$$\lambda = \lambda^{min} + ((\lambda^{max} - \lambda^{min}) / (2^{l} - 1))*Y^1$$

$$=6 + ((10-6) / (2^{16}-1))* 16821$$

$$= 7.8 \text{ Rs/MWh}$$

Find P_i's:

Now after calculating incremental cost, power generations P_i's are calculated for their diagonal location as:

$$\sum A^{j}_{ik}*P^{j}_{k} = C^{j}_{i} \text{ where (i=1, 2, 3; j=1)}$$

where k varies from 1 to 3.

Now after calculating $P_i^{'}s$ we have to check their constraints that whether the calculated $P_i^{'}s$ are in limits or not. If the values are in limits then next step would be performed else we have to set $P_i^{'}s$.

In our first iteration the values for three generators were:

P1 = **45.9** P2=**74.10** P3=**62.50**

Find cost, transmission losses and error:

Cost can be simply calculated with the help of $P_i^{'}s$ generated in previous step. The formula for cost calculation is :

$$F = \sum(a_i (P_i^2) + b_i P_i + c_i \quad = \mathbf{1425.4\ Rs/hr}$$

$$(i = 1,2,3)$$

And transmission losses can be calculated using:

$$P_L = \sum\sum P_i B_{ij} P_j \quad = \mathbf{1.8\ MW}$$

$$(i,k= 1,2,3)$$

Error can be calculated using equation:

$$\epsilon = | P_D + P_L - \sum P_i | \quad = \mathbf{31.7}$$

Fitness Process:

We have to find the population with maximum fitness because the chromosome with highest fitness will be more suitable to survive and will give more better results.

The formula used to calculate the fitness function is given below:

$$F= 1 / ((1+ error) / PD)) \quad = \mathbf{0.8257}$$

$$\text{Where error} = | PD + PL - \sum P_i |$$

Reproduction Process:

6(a). Set selection rate and number of mating in a pool.

6(b). Define total fitness as a sum of values obtained by using above step for all chromosomes which are selected.

6(c). Select a percentage of the roulette wheel each chromosome which is equal to the ratio of its fitness value to the total fitness value. i.e. find probability which can be written as:

$$\text{Probability} = \text{fitness} / \sum \text{Fitnesses}$$

6(d). Calculate cumulative sum (CS) to normalize the values on the Roulette wheel between 0.0 and 1.0.

Crossover Process:

7(a). Choose a pair of random numbers between 0 and 1 to select one mother(ma) and one father(pa) chromosomes so as to produce new offsprings.

7(b). pairing the chromosomes from different location,for different locations crossover point has to be selected which can be selected randomly, so as Generate offsrings by applying crossover.

Furthermore, crossover can be applied over single point, two-point and multipoint. Following example represents the single point crossover.

Parent 1:

0	1	1	0	1	1	1	1	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**1**

Parent 2:

0	1	0	1	1	1	1	1	**1**	**0**	**0**	**0**	**0**	**1**	**0**	**1**

Offspring 1

0	1	1	0	1	1	1	1	**1**	**0**	**0**	**0**	**0**	**1**	**0**	**1**

Offspring 2

0	1	0	1	1	1	1	1	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**1**

After crossover, the offsprings which are generated contains traits from mother and father chromosomes. This will add to the new information to the offsprings.

After applying crossover, again whole process is repeated and with new calculated values of P_i's(offsprings), again power losses, cost, fitness etc is calculated and checked whether the new solutions provide the better results or not.

Mutation Process:

Perform the mutation. In this we have, we have to flip one bit. If its 0, then, flip it equal to 1 and vice-versa. But before that, mutation rate is to be defined.

Random mutations alter a certain percentage of the bits in the list of chromosomes. Mutation is the second way a GA explores a cost surface. It can introduce traits not in the original population and keeps the GA from converging too fast before sampling the entire cost surface. Mutation points are randomly selected from the total number of bits in the population matrix. Increasing the number of mutations increases the algorithm's freedom to search outside the current region of variable space.

Example of mutation process can be depicted as follows:

The following pairs were randomly selected:

mrow=[5 7 6 3 8 4 9]

mcol= [6 12 7 1 3 8 5]

The first random pair is (5, 6). Thus the bit in row 5 and column 6 of the population matrix is mutated from a 1 to a ***0***:

1	1	0	0	0	*0*	0	0	0	0	1	0	0	1	0	0

1	1	0	0	0	*1*	0	0	0	0	1	0	0	1	0	0

The next generation or updating Population:

After the mutations take place, the fitness associated with the offsprings (after crossover and mutation) are calculated. The process described is iterated, replacing the old population or (initial population) with the new improved population generated after passing through the whole process of selection, crossover and mutation again. The results for first iteration are shown below in table(4.4)

Sno	λ	FF	Err	P1	P2	P3	TP	PL
1	7.3305	0.75397	48.947	18.369	50.871	32.427	101.67	0.61371
2	7.8323	0.82567	31.672	45.907	75.067	62.481	183.45	1.7829
3	6.3489	0.55546	120.05	10	10	10	30	0.0451
4	6.6242	0.56841	113.89	10	16.175	10	36.175	0.069992
5	6.7019	0.57678	110.06	10	20.028	10	40.028	0.091475
6	8.5332	0.64946	80.96	83.687	80	70	233.69	2.727
7	6.1563	0.55546	120.05	10	10	10	30	0.0451
8	7.2429	0.70347	63.229	13.519	46.609	27.121	87.249	0.47783
9	9.8793	0.64588	82.243	85	80	70	235	2.7571
10	8.7203	0.64588	82.243	85	80	70	235	2.7571
11	6.6242	0.56846	113.87	10	16.175	10	36.175	0.0451
12	7.8323	0.81841	33.283	45.907	75.067	62.481	183.45	0.1715
13	7.3305	0.75531	48.594	18.369	50.871	32.427	101.67	0.26079
14	6.3489	0.54994	122.76	10	10	10	30	2.7571
15	8.8742	0.64546	82.391	85	80	70	235	2.6094
16	7.8792	0.79303	39.147	48.461	77.312	65.262	191.04	1.8878
17	8.3783	0.67372	72.645	75.402	80	70	225.4	2.7571
18	7.8792	0.786	40.84	48.461	77.312	65.262	191.04	0.19522
19	6.7019	0.57093	112.73	10	20.028	10	40.028	2.7571
20	7.3305	0.74829	50.456	18.369	50.871	32.427	101.67	2.1225

Table 4.4 Results of first iteration

The optimal result for first iteration is shown in table 4.5

St no	Itr	FF	λ	err	P1	P2	P3	TP	Losses	Cost
2	1	.82567	7.8323	31.672	45.907	75.067	62.481	183.45	1.7829	1425.4

Table 4.5 Optimal Result for first iteration

Step 10). **Convergence:**

The number of generations that evolve depends on whether an acceptable solution is reached or a set number of iterations is exceeded. After a while all the chromosomes and associated costs would become the same if it were not for mutations. At this point, the algorithm should be stopped.

The results for our final iteration is shown in table 4.6 and the optimal result for three generator with losses are shown in table 4.7.

RESULTS OF GA WITH LOSSES

S. no.	λ	TP	Loss	Error	Fitness
1	7.8732	190.066	1.907	38.159	0.7971
2	6.1406	30	0.0451	120.0451	0.5554
3	6.5458	32.279	0.0529	117.7739	0.5601
4	8.0805	209.3745	2.2539	57.1206	0.7242
5	7.6372	151.8131	1.2503	0.5629	0.9963
6	8.3826	225.6333	2.5525	73.0808	0.6724
7	6.3303	30	0.0451	120.0451	0.5554
8	6.6777	38.8293	0.0843	111.255	0.5742
9	7.9067	195.4692	2.0117	43.4575	0.7753
10	7.8858	192.0975	1.9461	40.1515	0.7888
11	7.6372	151.8131	1.2503	0.5629	0.9962
12	7.6372	151.8131	1.2503	0.5629	0.9962
13	6.1551	30	0.0451	120.0451	0.5555
14	6.5312	31.5554	0.0503	118.4948	0.5587
15	8.0938	210.0921	2.2656	57.8265	0.7218
16	7.6333	151.1801	1.2406	0.0606	0.9996
17	8.0414	207.2614	2.2204	55.041	0.7316
18	6.702	40.0371	0.0915	110.0544	0.5768
19	6.6958	39.7272	0.0896	110.3624	0.5761
20	7.8886	192.5468	1.9547	40.5921	0.7870

Table 4.6 Results for final iteration

Optimal result:

Str.loc	Iterno	fitness	λ	Err	P1	P2	P3	TP	Losses	cost
16	21	0.9999	7.6333	0.0606	35.03	65.51	50.63	151.1801	1.24	1588.66

Table 4.7 Results for final iteration

The same problem of six generators is also solved by neglecting losses in the system and its final results are shown below in table 4.8.

Results of GA without losses

S. no.	λ	TP	Loss	Error	Fitness
1	7.323	114.3771	0	35.6229	0.8081
2	7.281	106.3175	0	43.6825	0.7745
3	7.442	136.8934	0	13.1066	0.9196
4	6.657	39.83210	0	110.1679	0.5765
5	7.374	124.0631	0	25.9369	0.8526
6	7.249	100.3179	0	49.6821	0.7512
7	6.419	30.0000	0	120.000	0.5555
8	7.798	199.8684	0	49.8684	0.7505
9	7.436	135.7152	0	14.2848	0.9130
10	7.509	149.7671	0	0.2329	0.9984
11	7.279	106.1079	0	43.8921	0.7736
12	7.436	135.7152	0	14.2848	0.9130
13	6.416	30.0000	0	120.000	0.5555
14	7.279	106.1079	0	43.8921	0.7736
15	7.799	199.9065	0	49.9065	0.7503
16	7.825	201.5469	0	51.5469	0.7442
17	7.511	150.010	0	0.0016	0.9999
18	7.322	114.1458	0	35.8542	0.8071
19	6.259	30.0000	0	120.000	0.5555
20	7.798	199.866	0	49.866	0.7502

Table 4.8 Results of GA without losses Case I

The optimal results of GA without losses are shown below in Table 4.9

Optimal result:

Str.loc	Iterno.	Fitness	λ	Err	P1	P2	P3	TP	Losses	cost
17	67	0.9999	7.511	0.0016	31.9367	67.277	50.7847	149.9984	0	1579.6872

TABLE 4.9 Optimal result of GA without losses Case I

GA	λ	P1	P2	P3	TP	TL	TC
With losses	7.6333	35.03	65.51	50.63	151.1801	1.24	1588.66
Without losses	7.511	31.9367	67.277	50.7847	149.9984	0	1579.6872

TABLE 4.10 Results of GA, (with and without losses)

4.1.3 Optimum Solution Using EP for Case-I

Some problem is solved using EP. The algorithm used for solving te problem with EP is same as that of GA exept that GA uses two operators: (i) crossover, and (ii) mutation whereas EP uses only one operator. Due to this there is difference between the results of GA and EP. But both are better than classical method.

The results of EP considering losses and neglecting losses are shown in table(4.11)

EP	λ	P1	P2	P3	TP	TL	TC
With losses	7.6257	35.1638	64.6283	50.7728	150.5649	1.2465	1590.5605
Without losses	7.5121	32.0082	67.3406	50.8665	150.1153	0	1581.116

TABLE 4.11 Results of EP (with and without losses)

4.2. CASE-II SIX GENERATOR TEST SYSTEM

The applicability of the proposed technique has also been tested on a sample system which consists of six generator system. The coefficients of fuel cost are given below in Tables 4.12(a).Transmission loss coefficients are given in Table 4.12(b). The power demand is considered to be 450MW. And Maximum and minimum power limits are given in Table 4.12(c)

Unit no.	a_i	b_i	c_i
1	0.005	2.0	100
2	0.010	2.0	200
3	0.020	2.0	300
4	0.003	1.95	80
5	0.015	1.45	100
6	0.010	0.95	120

(a)

Each coefficient is multiplied by 10^{-2} i.e. $B_{ij}*10^{-2}$

0.0200	0.0010	0.0015	0.0005	0.0000	0.0030
0.0010	0.0300	-0.0020	0.0001	0.0012	0.0010
0.0015	-0.0020	0.0100	0.0010	0.0010	0.0008
0.0005	0.0001	0.0010	0.0150	0.0006	0.0050
0.0000	0.0012	0.0010	0.0006	0.0250	0.0020
0.0030	0.0010	0.0008	0.0050	0.0020	0.0210

(b)

Limits	Maximum	Minimum
P1	85	10
P2	80	10
P3	70	10
P4	250	50
P5	150	5
P6	100	15

(c)

TABLE-4.12. Specifications for 6-generator test system

(a) Cost coefficients

(b) Loss coefficients

(c) Power generation limits

The simulated results are presented for various cases using 6-generator sample test system using LMM, with losses and without cases:

4.2.1 Optimum Solution Using LMM for Case-II

In table 4.13 results of LMM are shown from which it can be seen that cost with losses is more than as compared to cost without losses.

LMM	λ	P1	P2	P3	P4	P5	P6	TP	TL	TC
With losses	5.562	80.92	75.64	68.64	88.446	71.34	65.80	451.78	1.78	1974.02
w/o losses	4.486	69.393	73.4	61.6508	84.901	70.401	90.2	449.95	0	1972.47

TABLE-4.13 Results using LMM for six generator test system

4.2.2 Optimum Solution Using GA for Case-II

The results of the above said problem has also been obtained by utilizing GA. The length of each GA string in the population has been taken as 16 and the population size has been assumed as 20. Therefore, population we have taken is 20 x 16 in binary. The crossover and mutation probabilities are taken 0.5 and 0.8 selection rate is 50%. The crossover points are chosen randomly. The results we obtain from genetic algorithm are shown below in table 4.14. These parameters and other constraints are shown above in table 4.12 (a,b,c)**.** The population considered represents the power generations separately for every generator therefore population of three variables would be generated. Because we have six generators, so there will be P1, P2, P3, P4, P5, P6 as population.

In order to simulate the proposed GA to solve ELD, a set of steps were produced. These set of steps and its implementation are same as discussed above, the simulated results are also shown in table(4.14).

GA	Lamb	P1	P2	P3	P4	P5	P6	TP	TL	TC
With losses	4.6698	72.48	60.52	64.432	75.432	94.055	85	451.51	1.5173	1970.02
w/o losses	4.466	79.393	63.4	61.6508	89.901	75.401	80.2	449.95	0	1969.47

TABLE-4.14. Results using GA

4.2.3 Optimum Solution Using EP for Case-II

The simulated results for EP for six generators with losses and without losses are shown in table 4.15

EP	Lamb	P1	P2	P3	P4	P5	P6	TP	TL	TC
With losses	5.534	**75.45**	**60.55**	**66.92**	**98.07**	**94.74**	**55.87**	**451.61**	**1.61**	**1972.28**
Without losses	**4.3989**	**68.97**	**65.82**	**59.9714**	**101.2214**	**73.7214**	**80.20**	**449.9142**	**0**	**1970.67**

TABLE-4.15 Results using EP

CHAPTER-5

CONCLUSIONS AND FUTURE SCOPE

In this work, the formulation and implementation of solution methods to obtain the optimum solution of Economic Load Dispatch problem using Genetic Algorithm and Evolutionary Programming is carried out.

The effectiveness of the developed program is tested for three generator and six generator test system for both considering the losses and neglecting the losses. The results obtained from these methods are also compared with the Classical Lagrange Multiplier method.

It is found that GA and EP are giving better results than LMM. i.e. EAs have proved itself better than the Classical Method.

The scope for further work in this field is identified as -

1. The GA or EP can be implemented to the coordinated dispatch of spinning reserves and bulk power.
2. Multiobjective optimization while considering environmental economic dispatch in which both the cost and emission level can be minimized.
3. Real value coding can be attempted for GA or EP and can be tested to large system.

REFERENCES

1. R.H. Miller, and J. H. Malinnowski, "*Power System Operation*", McGraw-Hill, Inc., 1994.

2. Palanichamy C. and K. Shrikrishna (1991), " *Simple algorithm for economic power dispatch"* electrical power syst. Res, Vol.21 pp 174-153.

3. W. Ongsakul and N. Ruangpayoongsak "*Constrained Dynamic Economic dispatch by Simulated Annealing/Genetic Algorithms*", IEEE Transaction on Power Systems, pp. 207, 2001.

4. R. Wayne Barcelo and Parvis Rastgoufard. "*Dynamic Economic Dispatch Using the Extended Security Constrained Economic Dispatch Algorithm*", IEEE Transaction on Power Systems, pp. 961, 1996.

5. Pathom Attaviriyanupap, Hiroyuko Kita, Eiichi Tanaka, and Jun Hasegawa, "*A Hybrid EP and SQP for Dynamic Economic Dispatch with Nonsmooth Fuel Function*", IEEE Transaction on Power Systems pp. 411 May 2002.

6. R.C. Eberhart, Y. Shi, "*Comparison between genetic algorithms and particle swarm optimization*", in: Proc. IEEE Int. Conf. Evol. Comput., May, 1998, pp.611-616.

7. Goldberg,D. E. 1989. "*Genetic Algorithms in Search, Optimization, and Machine Learning.Reading*", MA: Addison-Wesley

8. Happ H.H. (1974),"*Optimal power dispatch"*IEEE trans. power apperat system, Vol. PAS- 93, pp. 820-830.

9. J.G. Damousis, A.G. Unkirlzis, P.S. Dokopoulos, "*Network- constrained economic dispatch using real-coded genetic algorithm*", IEEE Trans. Power Syst. 8 (November (4)) (1995) 198-205.

10. Deb K. (1995) "*Optimization for engineering design*", prentice hall of India.

11. Goldberg D.B. (1989)"*Genetic algorithm in search optimization and machine learning*", addisen Wesley.

12. A. Bakirtzis, V. Petridis, and S. Kazarlis, "*Genetic Algorithm Solution to the Economic Dispatch Problem*", IEE Proc.-Gener. Transm. Distrib., Vol. 141, No.4, July 1994, pp 377-382.

13. Pathom Attaviriyanupap, Hiroyuko Kita, Eiichi Tanaka, "*Power System Analysis and DesignJun Hasegawa: A Hybrid EP and SQP for Dynamic Economic Dispatch with Non-smooth Fuel Function*", IEEE Transaction on Power Systems, pp. 411 May 2002.

14. WONG, K. P., and WONG, Y. W., 1994, "*Genetic and genetic simulated annealing approaches to economic dispatch*". IEE Proceedings C., 685–692.

15. Bakirtzis A. G., Biskas P.N., Zoumas C. E. & V. Petridis(2002)," *Optimal Power Flow by Enhanced Genetic Algorithm*", IEEE Trans., PWRS-17 (2), pp. 229-236.

16. Megahed I. (1997)"*A Modified Method for Solving the Economic Dispatching Problem*," IEEEE Transactions on Power Apparatus and Systems, Vol. PAS-96 (I), pp. 124-133.

17. WONG, K. P., and WONG, Y. W., 1994, "*Genetic and genetic simulated annealing approaches to economic dispatch*". IEE Proceedings C., 685–692.

18. Holland, J. H. 1992. *Genetic algorithms*. Sci. Am. 267:66–72.

19. Haupt,R.L.,and S.E.Haupt.2000.*Practical Genetic Algorithms*, 1st ed.New York:Wiley.

20. M. Salami, and G. Cain, "*Multiple Genetic Algorithm Processor for The Economic Power Dispatch Problem*", Proceeding of the Genetic Algorithm in Engineering Systems: Innovations and Applications, Conference Publication, No. 414, IEE, 1995, pp 188-193.

21. M. E. El-Hawary and S. Y. Mansour(1982), "*Performance Evaluation of Parameter Estimation Algorithms for Economic Operation of Power Systems*," IEEE Transactions on Power Apparatus and Systems, Vol. PAS-I01 (3), pp. 574-582.

22. Orero S.O. & M.R. Irving (1996), "*Economic Dispatch of Generators with Prohibited Operating Zones: A Genetic Algorithm Approach*", IEE Proc.-Gener. Transm. Distrib., 143 (6) pp. 529-534.

23. M. Salami, and G. Cain, "*Multiple Genetic Algorithm Processor for The Economic Power Dispatch Problem*", Proceeding of the Genetic Algorithm in Engineering Systems: Innovations and Applications, Conference Publication, No. 414, IEE, 1995, pp 188-193.

24. Ruangpayoongsak N., Ongsakul W. & Runggeratigul S. (2002), "*Constrained Economic Dispatch by Combined Genetic and Simulated Annealing Algorithm*", Electric Power Components and Systems, 30, pp. 917-931.

25. Saddukaran M.R.(2004)"*Applications of refined Genetic algorithm to combined Economic and Emission dispatch"*, Journal institution of engineers(India), Vol 85,pp.115-119.

26. Angeline, P. J. 1995. Evolution revolution: An introduction to the special track on genetic and evolutionary programming. IEEE Exp. Intell. Syst. Appl. 10:6–10.

27. D.B. Fogel, Evolutionary Computation: "*Toward a New Philosophy of Machine Intelligence*", second ed., IEEE Press, Piscataway, NJ, 2000.

28. X. Yao, Y. Liu, and G. Lin, (1999) "*Evolutionary Programming made Faster*," IEEE Trans Evolutionary Computation. Vol 3, no 2, 1999, pp. 82-102.

29. Angeline, P. J. 1995. Evolution revolution: An introduction to the special track on genetic and evolutionary programming. IEEE Exp. Intell. Syst. Appl. 10:6–10.

30. Dhillon J.S. and Kothari D.P. (2000),"*The surrogate worth trade off approach for multiobjective thermal power dispatch problem*", Electric power system search ,Vol 56(02) pp. 103-110.

31. Abido M.A. (2003), "*Economic Power Dispatch Using Multiobjective Evolutionary Algorithms"*, IEEE Transactions on Power Systems 18 (4), pp. 1529-1537.

32. Asgarpoor S., Panarelli S.K. (1995), *"Expected Cost Penalty due to Deviation from Economic Dispatch for interconnected power systems*", IEEE Transactions on Power Systems, Vol. 10, Issue 1, pp.441-447.

33. Kothari D.P. and Dhillon J.S. (2004*)," power system optimization, prentice hall Of India*".

34. Wood Allen.J. & Woolenber Bruce.F. (1996)," *power generation, operation and Control*",johnwilley & sons.

35. ALSAC, O.—BRIGHT, J.—PARIS, M.—STOTT: *Further Developments in LP-Based Optimal Power Flow*, IEEE Transaction of Power Systems 5 No. 3 (August 1990), 697–711.

36. BJORNDAL, M.—JORNSTEN, K. : *Zonal Pricing in a Deregulated Electricity Market*, The Energy Journal 22 No. 1 (2001).

37. "Practical Genetic Algorithms" / Randy L. Haupt, Sue Ellen Haupt. – 2nd ed, P. cm,1998

38. J.E. Lansbeny, L. Wozniak, Adaptive hydrogenerator governor tuning with a genetic algorithm, IEEE Paper 93 WM 136-2 EC, presented at the IEEERES 1993 Winter Meeting.

39. Carpentir J., *Contribution à l'étude du dispatching économique*, Bulletin de la Société Française des Electriciens, vol. 3, August 1962.

40. DOMMEL, H. W. "*Optimal Power Dispatch*", IEEE Transactions on Power Apparatus and Systems PAS93 No. 3 (May/june 1974), 820–830.

41. D. E. Goldberg, "Genetic Algorithms in Search, Optimization & Machine *Learning",* Addison Wesley, 1989.

42. L.K. Kirchmayer, 'Economic operation of power system.'John Wiley,eastern Ltd,New York, 1958

43. J S Dhillon,S C Parti and D P Kothari," Multiobjective stochastic optimal thermal power dispatch, ibid, pp. 136-140, 1987.

44. D.B. Fogel, and K. Chellapilla, "Revisiting Evolutionary Programming", Proc. Of Applications and Science of Computional Intelligence. Orlando, pp. 2-11, 1998.

45. D.B. Fogel, and L.J. Fogel, "An Introduction to Evolutionary Programming", European Conf. on AE'95, Springer, Berlin, pp. 21-33, 1996.

46. G. B. Sheble and K. Brittig, "Refined genetic algorithm-economic dispatch example," *IEEE Trans. Power Syst.*, vol. 10, pp. 117–124, Feb. 1995.

47. D.B. Fogel, "Applying Evolutionary Programming to selected TSP", Cybernetics and Systems, Vol 24, pp. 27-36, 1993.

48. HAPP, H.H.:'Optimal power dispatch', IEEE Trans., 1974, PAS-93, pp. 820-830.

49. HAPP, H.H.:'Optimal power dispatch-a comprehensive survey', IEEE Trans., 1977, PAS-96, pp. 841-854.

50. RAMANATHAN, R.:'Fast economic dispatch based on penalty factor from Newton's method', IEEE Trans., 1985, PAS-104, pp. 1624-1629.

51. WOOD, A.J., and WOLLENBURG, B.F,:'Power generation, operation & control',John Wiley, New York, 1984.

52. D.B. Fogel, Evolutionary Computation: "*Toward a New Philosophy of Machine Intelligence*", second ed., IEEE Press, Piscataway, NJ, 2000.

53. Heitkoetter, J.; Beasley, D.; The Hitch-Hiker's Guide to Evolutionary Computation: A list of Frequently Asked Questions (FAQ), USENET: comp.ai.genetic. Available via anonymous FTP from rtfm.mit.edu:/pub/usenet/news.answers/ai-faq/genetic/; 1996; About 100 pp;

54. X. Yin, N. Getmay, "*Investigations on solving the load flow problem by genetic algorithm*", EPSR 22 (1991) 151-163.

55. P.H. Chen, H.C. Chang, "*Large scale economic dispatch by genetic algorithm*", IEEE Trans, Power Syst. 10 (4) (1995) 1919-1926.

56. IEEE Committee Report, "Present Practices in the Economic Operation of Power Systems," *IEEE Transactions on Power Apparatus and Systems,* Vol. PAS-90, July/ August 1971, pp. 1768-1775.

57. H. Saadat, "*Power System Analysis, Second Edition*," Mcgraw Hill, United States of America, 2002.