ANALYSIS AND COMPARISON OF ECONOMIC LOAD DISPATCH USING GENETIC ALGORITHM AND PARTICLE SWARM OPTIMIZATION

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

Master of Engineering
In
Power Systems & Electric Drives

Thapar University, Patiala

By
ARUNPREET KAUR
Roll No: 800941004

Under the supervision of

Ms. Suman Bhullar
Assistant Professor, EIED

JULY 2011

ELECTRICAL & INSTRUMENTATION ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA-147004

CERTIFICATE

I hereby certify that the work which is being presented in this thesis entitled. **"Analysis and Comparison of Economic Load Dispatch Using Genetic Algorithm and Particle Swarm Optimization"**, in partial fulfillment of the requirements for the award of degree of Master of Engineering in Power Systems and Electric Drives submitted in Electrical and Instrumentation Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of Ms. Suman Bhullar, Assistant Professor (EIED). The matter embodied in this thesis has not been submitted for the award of any other degree to any other university.

Arunpreet Kaur

(Arunpreet Kaur)

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

Bhullar 12/7/11

Ms. Suman Bhullar

Assistant Professor (EIED)

Thapar University,

Patiala-147004

Countersigned by:

S. Ghosh 14/7/11

(Dr. Smarajit Ghosh)

Professor & Head

Electrical & Instrumentation Engg. Department

Thapar University

(Dr. S. K. Mohapatra)

Dean (Academic Affairs)

Thapar University

Patiala

ACKNOWLEDGEMENT

I wish to express my special thanks and deepest regards to my Supervisor, Ms. Suman Bhullar, 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. Without her wise counsel and able guidance, it would have been impossible to complete the present work.

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 for their valuable advice and suggestion during the thesis work.

The paucity of words does not compromise for extending my thanks to my parents, whose uninterrupted love, inspiration and blessings helped me in completing this research report. I wish to thank all my classmates of M.E. (PSED) for their time to time suggestions and cooperation. In the last but not the least, multi million thanks to the Almighty God, who gave me the opportunity and strength to carry out this thesis work.

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

ARUNPREET KAUR

Regn no. 800941004

ABSTRACT

Economic Load Dispatch (ELD) problem is one of the most important ones in power system operation and planning. The main objective of the ELD problems is to determine the optimal combination of power outputs of all generating units so as to meet the required demand at minimum cost while satisfying the constraints. Conventionally, the cost function for each unit in ELD problems has been approximately represented by a quadratic function and is solved using mathematical programming techniques. Generally, these mathematical methods require some marginal cost information to find the global optimal solution. Unfortunately, the real-world input output characteristics of generating units are highly nonlinear and non-smooth because of prohibited operating zones, valve point loadings, and multi-fuel effects, etc. Thus, the practical ELD problem is represented as a non-smooth optimization problem with equality and inequality constraints, which directly cannot be solved by the mathematical methods. Over the past decade, in order to solve these non-smooth ELD problems, many salient methods have been developed such as hierarchical numerical method, genetic algorithm, evolutionary programming, neural network approaches, differential evolution, particle swarm optimization, and the hybrid method.

In this thesis, the two main types evolutionary optimization techniques namely Genetic Algorithm (GA) and Particle Swarm Optimization (PSO), which are generic population based probabilistic search optimization algorithms and can be applied to real world problem are respectively applied to solve an ELD problem. And at the last the comparison between both the methods has been presented. The PSO provides the generation level such that the generation cost is coming out to be lower than the cost resulted with Genetic Algorithm method.

TABLE OF CONTENTS

Page no.

LIST OF FIGURES

Page no.

LIST OF TABLES

Page no.

CHAPTER 1: INTRODUCTION

1.1 OVERVIEW

Electrical power systems are designed and operated to meet the continuous variation of power demand. In power system, minimization of the operation cost is very important. Economic Load Dispatch (ELD) is a method to schedule the power generator outputs with respect to the load demands, and to operate the power system most economically, or in other words, we can say that main objective of economic load dispatch is to allocate the optimal power generation from different units at the lowest cost possible while meeting all system constraints.

Over the years, many efforts have been made to solve the ELD problem, incorporating different kinds of constraints or multiple objectives through various mathematical programming and optimization techniques. The conventional methods include Newton-Raphson method, Lambda Iteration method, Base Point and Participation Factor method, Gradient method, etc **[1]**. However, these classical dispatch algorithms require the incremental cost curves to be monotonically increasing or piece-wise linear **[2]**. The input/output characteristics of modern units are inherently highly nonlinear (with valve-point effect, rate limits etc) and having multiple local minimum points in the cost function. Their characteristics are approximated to meet the requirements of classical dispatch algorithms leading to suboptimal solutions and therefore, resulting in huge revenue loss over the time. Consideration of highly nonlinear characteristics of the units requires highly robust algorithms to avoid getting stuck at local optima **[3]**. The classical calculus based techniques fail in solving these types of problems. In this respect, stochastic search algorithms like genetic algorithm (GA**) [4]-[9]**, evolutionary strategy (ES) **[10]-[12]**, evolutionary programming (EP) **[2], [3], 13]**, particle swarm optimization (PSO) **[27]** and simulated annealing (SA) may prove to be very efficient in solving highly nonlinear ELD problem without any restrictions on the shape of the cost curves. Although these heuristic methods do not always guarantee the global optimal solution, they generally provide a fast and reasonable solution (sub optimal or near global optimal).

This thesis work proposes evolutionary optimization techniques namely Genetic Algorithm (GA) and Particle Swarm optimization (PSO) 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. Both techniques are respectively applied to solve an ELD problem by using the proposed algorithms mentioned in chapter 3 and 4 respectively. And at the last the comparison between the two methods has been presented.

1.2 LITERATURE REVIEW

Fink L. H., *et al.* [14] described the valve-point loading logic which is intended to meet at any time in the most economical fashion a generation commitment. This objective is approached by insuring that as great a portion of the load as practicable will be carried by units loaded to valve points, that the remainder of the load will be carried by units reserved for regulation, and that in both categories the assignments will be made to those units which can provide the requisite capacity at the lowest cost.

Reid and Hasdorff [15] formulated the economic load dispatch problem as a quadratic programming problem and solved using Wolfe's algorithm. The method is capable of handling both equality and inequality constraints on p, q, and v and can solve the load flow as well as the economic load dispatch problem. The quadratic programming algorithm does not require the use of penalty factors or the determination of gradient step size which can cause convergence difficulties.

Megahed *et al.* [16] developed a method for solving the economic load dispatching problem by changing it from constrained nonlinear programming problem to a sequence of constrained linear programming problems. The formulation of the load scheduling is exact in the sense that all the system voltages, active and reactive generation, as well as the phase angles are considered as independent variables. In addition, the effect of bus voltages on the loads is taken into consideration.

Happ [17] reviewed the progress of optimal dispatch, also called economic load dispatch, since its inception to the present in chronological sequence. The classic single area as well as multi area cases is summarized, and the important theoretical work in optimal load flows suggested to date reviewed. Approaches to the optimal load flow taken by industry are also reported, as well as an itemization of problems that still remain to be solved.

Kwatny and Athay [18] presented the coordination of the economic load dispatch and regulation functions of automatic generation control in electric power systems. The point of view taken is that such coordination appropriately takes place at the regulation or load frequency control level. Thus, the coordinating controller is obtained through the formulation of a suitably extended load frequency control problem in the context of linear multivariable control theory.

Ross and Kim [19] developed a set of procedures and algorithms for dynamic economic dispatch of generation units. When coupled with a short-term load predictor, "look-ahead" capability is provided by the dynamic economic dispatch that coordinates predicted load changes with the rate-of-response capability of generation units. Dynamic economic load dispatch also enables valve-point loading of generation units.

Bottero, *et al.* [20] discussed that in general, second derivative or Hessian based optimization methods have much higher convergence efficiency than those techniques based on the gradient of the objective function. Unfortunately, for problems such as economic load dispatch where the generation cost has to be minimized subject to the load flow equality constraints, the reduced Hessian with respect to the controllable variables is, in general, non-sparse and requires extensive computations to evaluate. In the literature this obstacle has been bypassed either by approximating the reduced Hessian or by handling equality constraints through penalty functions added to the cost.

Aoki and Satoh [21] presented an efficient method to solve an economic load dispatch problem with dc load flow type network security constraints. The conventional linear programming and quadratic programming methods cannot deal with transmission losses as a quadratic form of generator outputs. In order to overcome this defect, the extension of the quadratic programming method is proposed, which is designated as the parametric quadratic programming method. The upper bounding technique and the relaxation method are coupled with the proposed method for the purpose of computational efficiency. The test results show that the proposed method is practical for real-time applications.

Lin and Viviani [22] presented a method to solve the economic power dispatch problem with piecewise quadratic cost functions. The solution approach is hierarchical, which allows for decentralized computations. An advantage of this approach is the capability to optimize over a greater variety of operating conditions. Traditionally, one cost function for each generator is assumed. In this formulation multiple intersecting cost functions are assumed. This method has application to fossil generation units capable of burning gas and oil, as well as other problems which result in multiple intersecting cost curves for a particular unit. The results show that the solution method is practical and valid for real-time application.

Ramanathan [23] presented an extremely fast, simple, efficient and reliable economic load dispatch algorithm. The algorithm utilizes a closed form expression for the calculation of the Lambda, as well' as taking care of total transmission loss changes due to generation change, thereby- avoiding ,any iterative processes in the calculations. The closed form expression presented for Lambda can be used with 'any type of incremental transmission loss calculation. For this algorithm, penalty factors are derived based upon the Newton's method.

Chowdhury and Rahman [24] presented a survey of papers and reports which address various aspects of economic load dispatch. The time period considered is 1977-88. This is done to avoid any repetition of previous studies which were published prior to 1977. Four very important and related areas of economic load dispatch are identified and papers published in the general area of economic dispatch are classified into these. These areas are: (i) Optimal power flow, (ii) Economic dispatch in relation to AGC, (iii) Dynamic dispatch and (iv) Economic dispatch with non-conventional generation sources.

Walters and Sheble [25] used genetics-based algorithm to solve an economic dispatch problem for valve point discontinuities. The algorithm utilizes payoff information of candidate solutions to evaluate their optimality. Thus, the constraints of classical Lagrange techniques on unit curves are circumvented. The formulations of an economic dispatch computer program using genetic algorithms are presented and the program's performance using two different encoding techniques is compared. The results are verified for a sample problem using a dynamic programming technique.

Chen and Chang [26] presented a new genetic approach for solving the economic dispatch problem in large-scale systems. A new encoding technique is developed. The chromosome contains only an encoding of the normalized system incremental cost in this encoding technique. Therefore, the total number of bits of chromosome is entirely independent of the number of units. The salient feature makes the proposed genetic approach attractive in large and complex systems which other methodologies may fail to achieve. Moreover, the approach can take network losses, ramp rate limits, and prohibited zone avoidance into account.

Eberhart and Kennedy [27] described the optimization of nonlinear functions using particle swarm methodology. Implementations of two paradigms are discussed and compared, including a recently developed locally oriented paradigm. Benchmark testing of both paradigms is described, and applications, including neural network training and robot task learning, are proposed. Relationships between particle swarm optimization and both artificial life and evolutionary computation are reviewed.

Orero and Irving [28] explored the use of a genetic algorithm for the solution of an economic dispatch problem in power systems where some of the units have prohibited operating zones. Genetic algorithms have a capability to provide global optimal solutions in problem domains where a complete traversing of the whole search space is computationally infeasible. Two different implementations of the genetic algorithm for the solution of this dispatch problem are presented: a standard genetic algorithm, and a deterministic crowding genetic algorithm model.

Wang, *et al.* [29] developed a new effective artificial neural network method for the solution of economic emission load dispatch (EELD) problems with thermal generations. The proposed-method can overcome numerical difficulty caused by conventional neural networks with network parameters, and the states of the dynamic system described by the new neural network converge globally to the optimal solution of the EELD problem whenever its initial points are located inside or outside the feasible region of the problem. The application and validity of the proposed algorithm are demonstrated with a sample system with three generators.

Venkatesh, ***et al.* [30]** applied Economic load dispatch (ELD) and economic emission dispatch (EED) to obtain optimal fuel cost and optimal emission of generating units, respectively. Combined economic emission dispatch (CEED) problem is obtained by considering both the economy and emission objectives. This bio objective CEED problem is converted into a single objective function using a price penalty factor approach. A novel modified price penalty factor is proposed to solve the CEED problem.

Madouh and El- Hawary [31] presented a new and simple technique to solve the optimal solution of a short-term economic dispatch problem of all thermal power system, when the load on the system is fuzzy. The hard constrained, using this technique are transferred to soft constraints. A triangular membership for the load is assumed.

Sinha, *et al.* [2] Evolutionary programming has emerged as a useful optimization tool for handling nonlinear programming problems. Various modifications to the basic method have been proposed with a view to enhance speed and robustness and these have been applied successfully on some benchmark mathematical problems. The performance of evolutionary programs on ELD problems is examined and presented in this paper in two parts. In Part I, modifications to the basic technique are proposed, where adaptation is based on scaled cost. In Part II, evolutionary programs are developed with adaptation based on an empirical learning rate.

Chiang [5] developed an improved genetic algorithm with multiplier updating (IGAMU) to solve practical power economic load dispatch (PELD) problems of different sizes and complexities with non-convex cost curves, where conventional mathematical methods are inapplicable. The improved genetic algorithm (IGA) provides an improved evolutionary direction operator and a migrating operator, enabling it to efficiently search and actively explore solutions. Multiplier updating (MU) is introduced to avoid deforming the augmented Lagrange function, which is adopted to manage the system constraints of PELD problems. The proposed IGAMU integrates the IGA with the MU.

Zhang, *et al.* [32] proposed a new economic load dispatch model that considers cost coefficients with uncertainties and the constraints of ramp rate. The uncertainties are represented by fuzzy numbers, and the model is known as fuzzy dynamic economic load dispatch model (FDELD). A novel hybrid genetic algorithm with quasi-simplex techniques is proposed to handle the FDELD problem. The algorithm creates offspring by using generic operation and quasi-simplex techniques in parallel. The quasi-simplex techniques consider two potential optimal search directions in generating prospective offspring.

Kumari and Sydulu [33] presented a Fast Genetic Algorithm (FGA) approach for solving Economic Load Dispatch (ELD) problem. GA's perform powerful global searches, but their long computation times limit them when solving large scale optimization problems. This method was described to overcome the limitation by starting with random solutions within the search space and narrowing down the search space by considering the minimum and maximum errors of the population members. Since the search space is restricted to a small region within the available search space, the algorithm works very fast. This feature of the algorithm is attractive when applied to ELD of large systems.

Fang and Hua [34]. introduced an improved PSO with the constraints partially solved combined with penalty function, the improved PSO refined in constraints management, swarm initialization method of PSO with respect to the features of ELD in power plant. Based on above improvements, the improved PSO solved the problem of premature convergence in PSO. This method has the advantage of few parameters to adjust, easy to implement, with high computational efficiency and high accuracy.

1.3 OBJECTIVE OF THE WORK

The objectives of the thesis work are summarised as follows –

- To find solution of economic load dispatch problem so that the total fuel cost is minimized while satisfying the power generation limits.
- Use global search techniques like GA/PSO to find the optimal settings.
- Investigate the effectiveness of these methods for ELD problem while neglecting the transmission losses.
- Compare the results obtained from the two methods i.e. Genetic Algorithm (GA) and Particle Swarm Optimization (PSO).

1.4 ORGANIZATION OF THE THESIS

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

Chapter 1: This chapter summarizes the brief introduction of economic load dispatch, literature review, scope of the work and organization of the thesis.

Chapter 2: This chapter describes the method of solving economic load dispatch.

Chapter 3: This chapter includes theory of Genetic Algorithm and algorithm for economic load dispatch using genetic algorithm.

Chapter 4: This chapter explains the theory of Particle Swarm Optimization and algorithm for economic load dispatch using PSO.

Chapter 5: This chapter presents the conclusions and also the comparison of both methods used for solving ELD, i.e. GA and PSO.

CHAPTER 2: ECONOMIC LOAD DISPATCH

2.1 ECONOMIC LOAD DISPATCH

The Economic Load Dispatch (ELD) can be defined as the process of allocating generation levels to the generating units, so that the system load is supplied entirely and most economically. For an interconnected system, it is necessary to minimize the expenses. The economic load dispatch 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. The objective of economic load dispatch is to minimise the overall cost of generation. The method of economic load dispatch for generating units at different loads must have total fuel cost at the minimum point.

In a typical power system, multiple generators are implemented to provide enough total output to satisfy a given total consumer demand **[35]**. 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.

There are many conventional methods that are used to solve economic load dispatch problem such as Lagrange multiplier method, Lambda iteration method and Newton- Raphson method. In the conventional methods, it is difficult to solve the optimal economic problem if the load is changed. It needs to compute the economic load dispatch each time which uses a long time in each of computation loops. It is a computational process where the total required generation is distributed among the generation units in operation, by minimizing the selected cost criterion, and subjects it to load and operational constraints as well.

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 Necessity of generation scheduling:

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 load 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 **[35].**

Therefore, the main aim in the economic load 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.

2.2 GENERATOR OPERATING COST

The total cost of operation includes the fuel cost, cost of labour, supplies and maintenance. Generally, costs of labour, supplies and maintenance are fixed percentages of incoming fuel costs. The power output of fossil plants is increased sequentially by opening a set of valves to its steam turbine at the inlet. The throttling losses are large when a valve is just opened and small when it is fully opened.

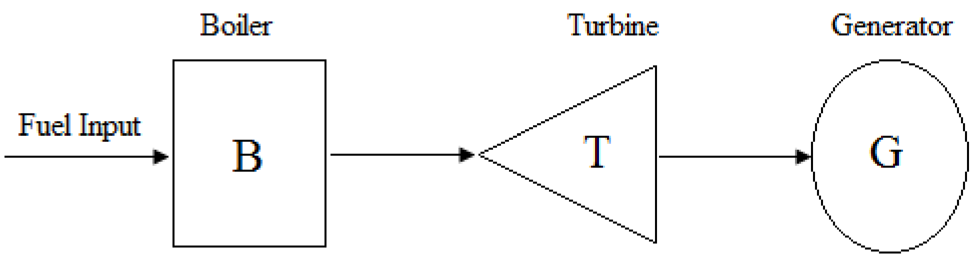

Figure 2.1 Simple model of a fossil plant

Figure 2.1 shows the simple model of a fossil plant dispatching purposes. The cost is usually approximated by one or more quadratic segments. The operating cost of the plant has the form shown in Figure 2.2. For dispatching purposes, this cost is usually approximated by one or more quadratic segments. So, the fuel cost curve in the active power generation, takes up a quadratic form, given as:

$$F(P_{gi}) = a_i P_{gi}^2 + b_i P_{gi} + c_i Rs/hr \tag{2.1}$$

where

a_i, b_i, c_i are cost coefficients for i^{th} unit

$F(P_{gi})$ is the total cost of generation

P_{gi} is the generation of i^{th} plant

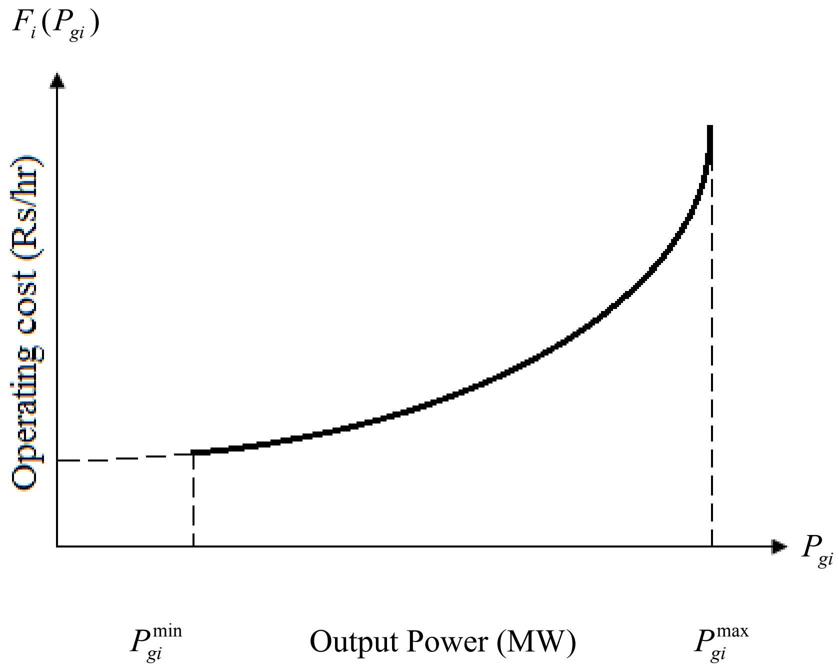

Figure.2.2 Operating costs of a fossil fired generator

The fuel cost curve may have a number of discontinuities. The discontinuities occur when the output power is extended by using additional boilers, steam condensers, or other equipment. They may also appear if the cost represents the operation of an entire power station, and hence cost has discontinuities on paralleling of generators. Within the continuity range the incremental fuel cost may be expressed by a number of short line segments or piece-wise linearization.

The P_{gi}^{min} is the minimum loading limit below which, operating the unit proves to be uneconomical (or may be technically infeasible) and P_{gi}^{max} is the maximum output limit **[36]**.

2.3 THE ECONOMIC LOAD DISPATCH PROBLEM

2.3.1 Economic Load Dispatch without Losses

The simplest economic load dispatch problem is the case when transmission line losses are neglected. Due to this the total demand P_D is the sum of all generations. A cost function $F_i(P_{gi})$ is assumed to be known for each plant. The problem is to find the real power generation, P_{gi} for each plant such that the total operating cost $F(P_{gi})$ is minimum and the generation remains within the lower generation $P_{gi}^{\min}$ and upper generation $P_{gi}^{\max}$. Suppose there is a station with NG generators committed and the active power load demand P_D is given, the real power generation P_{gi} for each generator has to be allocated so as to minimize the total cost. The optimization problem can be therefore be stated as

Minimize: $$F(P_{gi}) = \sum_{i=1}^{NG} F_i(P_{gi}) \qquad (2.2a)$$

Subject to:

(i) The energy balance equation

$$\sum_{i=1}^{NG} P_{gi} = P_D \qquad (2.2b)$$

(ii) And the inequality constraints

$$P_{gi}^{\min} \le P_{gi} \le P_{gi}^{\max} \qquad (i = 1,2,.........., NG) \qquad (2.2c)$$

where

P_{gi} is the decision variable, i.e. real power generation

P_D is the real power demand

NG is the number of generation plants

$P_{gi}^{\min}$ is the lower permissible limit of real power generation

$P_{gi}^{\max}$ is the upper permissible limit of real power generation

$F_i(P_{gi})$ is the operating fuel cost of the i^{th} plant and is given by the quadratic equation

$$F_i(P_{gi}) = a_i P_{gi}^2 + b_i P_{gi} + c_i Rs/h \tag{2.2d}$$

The above constrained optimization problem is converted into an unconstrained optimization problem. Lagrange multiplier is used in which a function is minimized (or maximized) with side conditions in the form of equality constraints. Using the method an augmented function is defined as

$$L(P_{gi},\lambda) = F(P_{gi}) + \lambda(P_D - \sum_{i=1}^{NG} P_{gi}) \tag{2.3}$$

Where λ is the Lagrange multiplier.

A necessary condition for a function $F(P_{gi})$, subject to energy balance constraint to have a relative minimum at point P_{gi}^* is that the partial derivative of the Lagrange function defined by $L = L(P_{gi},\lambda)$ with respect to each of its arguments must be zero. So, the necessary conditions for the optimization problem are

$$\frac{\partial L(P_{gi},\lambda)}{\partial P_{gi}} = \frac{\partial F(P_{gi})}{\partial P_{gi}} - \lambda = 0 \qquad (i = 1,2,........,NG) \tag{2.4}$$

And

$$\frac{\partial L(P_{gi},\lambda)}{\partial \lambda} = P_D - \sum_{i=1}^{NG} P_{gi} = 0 \tag{2.5}$$

From equation (2.4),

$$\frac{\partial F(P_{gi})}{\partial P_{gi}} = \lambda \qquad (i = 1,2,........, NG) \tag{2.6}$$

Where $\partial F(P_{gi}) / \partial P_{gi}$ is the incremental fuel cost of the i^{th} generator.

Optimal loading of generators corresponds to the equal incremental cost point of all the generators. Eq. (2.6), called the coordination equations numbering NG are solved simultaneously with the load demand to yield a solution for Lagrange multiplier λ and the optimal generation of NG generators. Considering the cost function given by Eq. (2.2d), the incremental cost can be defined as

$$\frac{\partial F(P_{gi})}{\partial P_{gi}} = 2a_i P_{gi} + b_i \tag{2.7}$$

Substituting the incremental cost into equation (2.6), this equation becomes

$$2a_i P_{gi} + b_i = \lambda \qquad (i = 1,2,......, NG) \tag{2.8}$$

Rearranging Eq. (2.8) to get P_{gi}

$$P_{gi} = \frac{\lambda - b_i}{2a_i} \qquad (i = 1,2,......, NG) \tag{2.9}$$

Substituting the value of P_{gi} in Eq. (2.5), we get

$$\sum_{i=1}^{NG} \frac{\lambda - b_i}{2a_i} = P_D \tag{2.10a}$$

Or

$$\lambda = \frac{P_D + \sum_{i=1}^{NG} \frac{b_i}{2a_i}}{\sum_{i=1}^{NG} \frac{1}{2a_i}} \tag{2.10b}$$

2.3.2 Economic Load Dispatch with Losses

Transmission losses may be neglected when transmission losses are very small but in a large interconnected network where power is transmitted over long distances, transmission losses are a major factor and affect the optimum dispatch of generation. The economic load dispatch problem considering the transmission power loss P_L for the objective function is thus formulated as:

Minimize:

$$F(P_{gi}) = \sum_{i=1}^{NG} F_i(P_{gi}) \tag{2.11a}$$

$$F_i(P_{gi}) = a_i P_{gi}^2 + b_i P_{gi} + c_i Rs/h \tag{2.11b}$$

Subject to:

(i) The energy balance equation

$$\sum_{i=1}^{NG} P_{gi} = P_D + P_L \tag{2.11c}$$

(ii) And the inequality constraints

$$P_{gi}^{\min} \le P_{gi} \le P_{gi}^{\max} \qquad (i = 1,2,.........., NG) \tag{2.11d}$$

The general form of the loss formula using B- coefficients is

$$P_L = \sum_{i=1}^{NG} \sum_{j=1}^{NG} P_{gi} B_{ij} P_{gj} \ \text{MW} \tag{2.12}$$

where

P_{gi} and P_{gj} are the real power generations at i^{th} and j^{th} buses respectively

B_{ij} are the loss coefficients or B-coefficients

The transmission loss formula of Eq. (2.12) is known as George's formula. Using the Lagrange multiplier λ, the augmented function is,

$$L(P_{gi},\lambda) = F(P_{gi}) + \lambda\left(P_D + P_L - \sum_{i=1}^{NG} P_{gi}\right) \tag{2.13}$$

For minimisation of augmented function,

$$\frac{\partial L(P_{gi},\lambda)}{\partial P_{gi}} = 0 \tag{2.14a}$$

$$\frac{\partial L(P_{gi},\lambda)}{\partial \lambda} = 0 \tag{2.14b}$$

$$\frac{\partial F_i(P_{gi})}{\partial P_{gi}} = \frac{\partial F(P_{gi})}{\partial P_{gi}} \tag{2.15}$$

The condition given by (2.14) results as,

$$\frac{\partial L(P_{gi,}\lambda)}{\partial P_{gi}} = \frac{\partial F(P_{gi})}{\partial P_{gi}} + \lambda\left(\frac{\partial P_L}{\partial P_{gi}} - 1\right) = 0 \qquad (i = 1,2,......, NG) \tag{2.16}$$

$$\frac{\partial F(P_{gi,})}{\partial P_{gi}} = \lambda\left(1 - \frac{\partial P_L}{\partial P_{gi}}\right) \qquad (i = 1,2,......, NG) \tag{2.17}$$

Or

$$\frac{\partial F(P_{gi,})}{\partial P_{gi}} + \lambda\left(\frac{\partial P_L}{\partial P_{gi}}\right) = \lambda \qquad (i = 1,2,......, NG) \tag{2.18}$$

Where $\frac{\partial F(P_{gi,})}{\partial P_{gi}}$ is called incremental fuel cost $(IC)_i$ and $\frac{\partial P_L}{\partial P_{gi}}$ is known as incremental transmission loss $(ITL)_i$, associated with i^{th} generating unit. Rearranging (2.18) results as,

$$\frac{\frac{\partial F(P_{gi})}{\partial P_{gi}}}{1-\frac{\partial P_L}{\partial P_{gi}}}=\lambda \tag{2.19}$$

$$\left(\frac{1}{1-\frac{\partial P_L}{\partial P_{gi}}}\right)\frac{\partial F(P_{gi})}{\partial P_{gi}}=\lambda \qquad (i=1,2,.......,NG) \tag{2.20}$$

Or

$$L_i\left(\frac{\partial F(P_{gi})}{\partial P_{gi}}\right)=\lambda \qquad (i=1,2,.......,NG) \tag{2.21}$$

Where L_i is called the penalty factor of the i^{th} plant given by

$$L_i=\frac{1}{1-\frac{\partial P_L}{\partial P_{gi}}} \tag{2.22}$$

Equation (2.13) shows that the minimum cost is obtained when the incremental cost of each plant multiplied by its penalty factor is same for all plants. Equation (2.20) is also written in alternative form as **[37]**:

$$(IC)_i=\lambda\left[1-(ITL)_i\right] \qquad (i=1,2,.......,NG) \tag{2.23}$$

This equation is referred to as the exact coordination equation. Thus it is clear that to solve the economic load dispatch problem, it is necessary to compute ITL for each plant and therefore functional dependence of transmission loss on real powers of generating plants must be determined. There are several methods, approximate and exact, for developing a transmission loss model. One of the most important, simple but approximate methods of expressing transmission loss as a function of generator powers is through B-coefficients. This method is reasonably adequate for treatment of loss coordination in economic scheduling of load between plants. The general form of loss formula using B-coefficients is given in (2.12) Simplifying the equation (2.12) and recognizing that $B_{ij} = B_{ji}$,

$$\frac{\partial P_L}{\partial P_{gi}} = \sum_{i=1}^{NG} 2B_{ij} P_{gj} \tag{2.24}$$

Assuming quadratic plant cost curves as given in equation (2.15), incremental cost is obtained as,

$$\frac{dF_i(P_{gi})}{dP_{gi}} = 2a_i P_{gi} + b_i \tag{2.25}$$

Substituting $\frac{\partial P_L}{\partial P_{gi}}$ and $\frac{dF_i(P_{gi})}{dP_{gi}}$ from above in the coordination equation (2.18),

$$2a_i P_{gi} + b_i + \lambda \sum_{i=1}^{NG} 2B_{ij} P_{gj} = \lambda \qquad (i = 1,2,......., NG) \tag{2.26}$$

Collecting all terms of P_{gi} and solving for P_{gi},

$$(2a_i + 2\lambda B_{ii}) P_{gi} = -\lambda \sum_{\substack{j=1 \\ j \neq i}}^{NG} 2B_{ij} P_{gj} - b_i + \lambda \qquad (i = 1,2,......, NG) \tag{2.27}$$

Or

$$P_{gi} = \frac{1 - \frac{b_i}{\lambda} - \sum_{\substack{j=1 \\ j \neq 1}}^{NG} 2B_{ij} P_{gj}}{\frac{2a_i}{\lambda} + 2B_{ii}} \qquad (i = 1,2,......, NG) \tag{2.28}$$

For any particular value of λ, above equation can be solved iteratively by assuming initial values of P_{gi}'s. Iterations are stopped when P_{gi}'s converge within specified accuracy.

2.4 ECONOMIC LOAD DISPATCH WITH VALVE POINT LOADING

Economic load dispatch (ELD) is considered one of the key functions in electric power system operation. The economic load dispatch problem is commonly formulated as an optimization problem, with the aim of minimizing the total generation cost of power system but still satisfying specified constrains. The input-output characteristics (or cost functions) of a generator are approximated using quadratic or piecewise quadratic function, under the assumption that the incremental cost curves of the units are monotonically increasing piecewise-linear functions. However, real input-output characteristics display higher-order nonlinearities and discontinuities due to valve-point loading in fossil fuel burning plant. The valve-point loading effect has been modelled in as a recurring rectified sinusoidal function, such as the one show in figure 2.3 **[36]**.

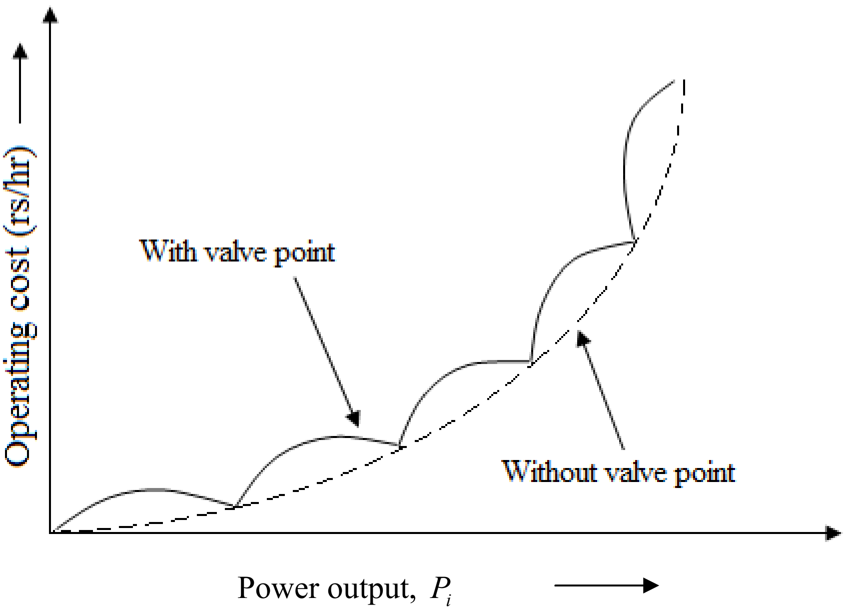

Figure.2.3 Operating cost characteristics with valve point loading

The generating units with multi-valve steam turbines exhibit a greater variation in the fuel cost functions. The valve-point effects introduce ripples in the heat-rate curves.

Mathematically, economic load dispatch problem considering valve point loading is defined as:

Minimize operating cost

$$F(P_i) = \sum_{i=1}^{NG} \left(a_i P_i^2 + b_i P_i + c_i + \left| d_i * \sin\{e_i * (P_i^{\min} - P_i)\} \right|\right) \qquad (2.29)$$

where

a_i, b_i, c_i, d_i, e are cost coefficients of the i^{th} unit.

Subject to:

(i) The energy balance equation is given by Eq. (2.11c) and

(ii) The inequality constraints are given by Eq. (2.11d)

CHAPTER 3: GENETIC ALGORITHM

3.1 GENETIC ALGORITHM STRUCTURE

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). GA is inspired by the evolutionary theory explaining the origin of species. In nature, weak and unfit species within their environment are faced with extinction by natural selection. The strong ones have greater opportunity to pass their genes to future generations via reproduction. In the long run, species carrying the correct combination in their genes become dominant in their population. Sometimes, during the slow process of evolution, random changes may occur in genes. If these changes provide additional advantages in the challenge for survival, new species evolve from the old ones. Unsuccessful changes are eliminated by natural selection.

The Genetic Algorithm (GA) is a search heuristic that mimics the process of natural evolution. This heuristic is routinely used to generate useful solutions to optimization and search problems. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover **[38]**.

Genetic Algorithms are search mechanisms based on the Darwinian principle of natural evolution. They operate on the law of coincidence, which takes advantage of pre-information in order to derive improvement from it. Genetic Algorithms used for optimization are based on the principle of biological evolution. They are very different to many conventional methods in the sense that they simultaneously consider many possible solutions to the problem. By considering many points in the search space, the algorithm simultaneously reduces the chance of getting trapped at a local minimum. They are the result of research done to incorporate the adaptive process of natural systems into design of artificial systems. GAs are computationally simple and provide robust search in complex problem spaces" **[39].** They work not with the parameters themselves but with a string of numbers representing the parameter set. Genetic Algorithms use a set of probabilistic rules in order to guide their search.

3.2 GENETIC ALGORITHM VERSUS TRADITIONAL METHODS OF OPTIMIZATION

Genetic algorithms are based on the principles of natural genetics and natural selection. The basic elements of natural genetics: reproduction, crossover, mutation are used in the genetic search procedure. Genetic Algorithms differ from the traditional methods of optimization in the following respect:

1.) A population of points (trial design vectors) is used for starting the procedure instead of a single design point. If the number of design variables is n, usually the size of the population is taken as 2n to 4n. Since several points are used as candidate solutions, Genetic Algorithms are less likely to get trapped at a local optimum.
2.) Genetic Algorithms use only the values of objective function. The derivatives are not used in search procedures.
3.) In GAs the design variables are represented as strings of binary variables that correspond to the chromosomes in natural genetics. Thus the search method is naturally applicable for solving discrete and integer programming problems. For continuous design variables, the string length can be varied to achieve any desired resolution.
4.) The objective function value corresponding to design vector plays the role of fitness in natural genetics.
5.) In every new generation, a new set of strings is produced by using randomized parents selection and crossover from the old generation (old set of strings). Although randomized, GAs are not simple random search techniques. They efficiently explore the new combination with the available knowledge to find the new generation with better fitness or objective function value.

The process of GA follows this pattern [40]:

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).

3.3 FLOW CHART OF GENETIC ALGORITHM

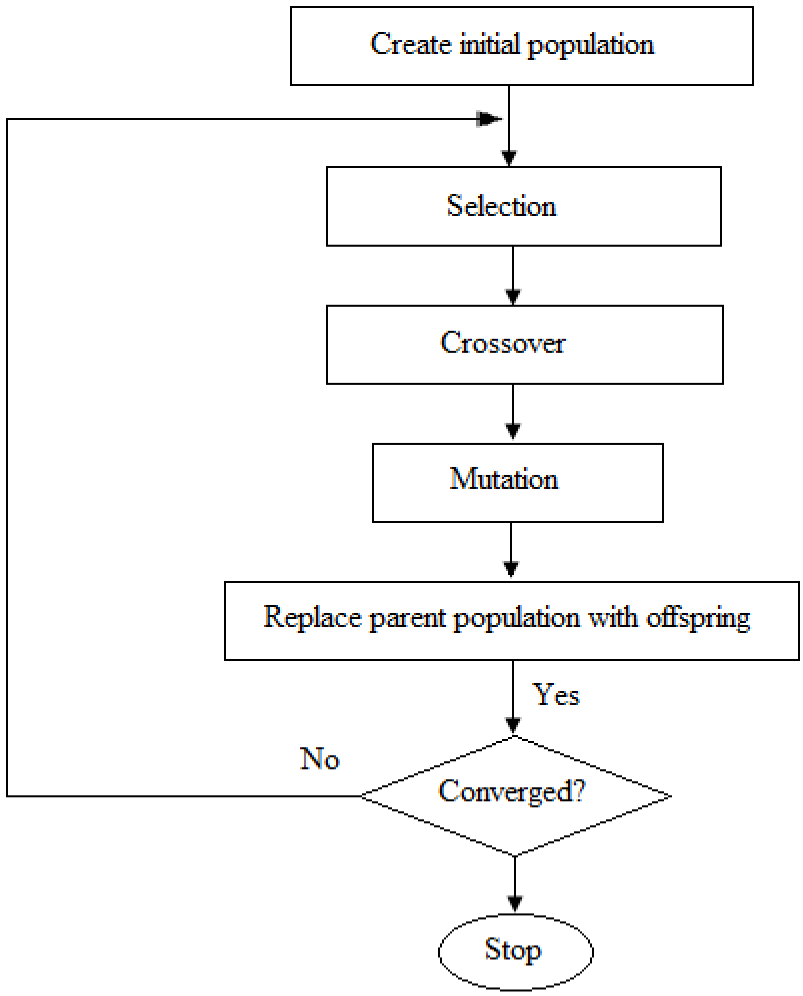

Figure.3.1 Flow chart of Genetic Algorithm

3.4 ADVANTAGES OF GENETIC ALGORITHM

The advantages associated with a Genetic Algorithm are **[41]-[43]**:

- Ease of implementation.
- Differentiability of the objective function is not required.

- Can handle complex, multi-nodal optimization problems.
- Computational simplicity.
- Power-full search ability to attain the global optimum.
- Extremely robust with respect to the complexity of the problem.
- Diversity of solutions is maintained with mutation.
- Takes into account the overall effect on the system.
- Simultaneously searches from a wide sampling of the cost surface.
- Deals with a large number of variables.
- Is well suited for parallel computers.
- Optimizes with continuous or discrete variables.
- 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.

3.5 DISADVANTAGES OF GENETIC ALGORITHM

The disadvantages associated with the use of a Genetic Algorithm are **[42, 44]**:

- Relatively complex when it comes to incorporating the algorithm into a software program.
- Relatively large computational time and effort.
- Premature convergence problems.

3.6 COMPONENTS NEEDED TO IMPLEMENT A GENETIC ALGORITHM

The components that are needed to implement a genetic algorithm are:

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

3.6.1 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. Chromosomes are made of discrete units called genes.

Encoding

Normally, a chromosome corresponds to a unique solution **x** in the solution space. This requires a mapping mechanism between the solution space and the chromosomes. This mapping is called an encoding. In fact, GA works on the *encoding* of a problem, not on the problem itself.

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. Traditionally, genetic algorithms have used mostly string structures containing binary decision variables. The binary coding is used in solving all the problems. The terminology used in GAs is borrowed from real genetics. The structure that encodes a solution is called a chromosome or individual. A decision variable is called a gene and its value is called allele.

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.

The equivalent decimal integer of binary string λ is obtained as

$$y^j = \sum_{i=1}^{l} 2^{i-1} b_i^j \qquad (j = 1,2,.....,L) \tag{3.1}$$

Where

b_i^j is the i^{th} binary digit of the j^{th} string

l is the length of the string

L is the number of strings or population size.

The continuous variable λ can be obtained to represent a point in the search space according to a fixed mapping rule, i.e.

$$\lambda^j = \lambda^{\min} + \frac{\lambda^{\max} - \lambda^{\min}}{2^l - 1} y^j \qquad (j = 1,2,.....,L) \tag{3.2}$$

where

$\lambda^{\min}$ is the minimum number of variable,

$\lambda^{\max}$ is the maximum value of variable,

y^j is the binary coded value of the string

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.

3.6.2 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 **[45]-[47].** Genetic Algorithms operate with a set of strings instead of a single string. This set of strings is known as a population and is put through the process of evolution to produce new individual strings. To start with, the initial population could be made up of chromosomes chosen at random or based on heuristically selected strings. The initial population should contain a wide variety of structures **[39]**. The number of chromosomes in a population is usually selected to be between 30 and 100 **[48]**. We need two parents population size and string length. Population size indicates the effective representation of whole search space in one population. It affects the efficiency and performance of GA. The selection of string length depends on the accuracy requirements of the optimization problem.

3.6.3 Evaluation

Suitability of the solutions is determined from the initial set of solution of the problem. For this suitability determination, we use a function called fitness function. This function is derived from the objective function and used in successive genetic operation. 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 **[39].**

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 $F(X)$

The fitness function for the maximization problem is:

$$f(x) = F(X) \tag{3.3}$$

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:

$$F(X) = 1/(1 + f(x)) \tag{3.4}$$

Here $f(x)$ is fitness function and $F(X)$ is objective function.

3.6.4 Genetic Operators

Genetic operators are a set of random transition rules employed by a Genetic Algorithm. In Genetic operation, a new and improved population is generated from the previous population using genetic operators. Genetic operators which are used in a Genetic Algorithm are **[39]**:

- Reproduction
- Crossover

- Mutation

3.6.4.1 Reproduction

Reproduction is a random selection process based on the rules of probability, in which chromosomes are selected to produce offspring based on their fitness values. This will ensure that the expected number of times a chromosome is chosen is proportional to its fitness, relative to the rest of the population. Strings with higher fitness values are more likely to contributing offspring, and are simply copied on into the next generation **[39]**. This operator is used to copy the old chromosome into matting pool according to its fittest valve. According to Darwin's fittest principle the best one should survive and create new offspring. 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. 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 i^{th} 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 i^{th} string is:

$$P_i = \frac{F_i}{\sum_{i=1}^{n} F_i} \qquad (3.5)$$

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, the roulette-wheel mechanism is expected to make / copies of the i^{th} string in the mating pool. The average fitness of the population is calculated as:

$$f_{av} = \left(\sum_{i=1}^{n} f_i \right) * \frac{1}{n} \tag{3.6}$$

The various methods of selecting chromosomes for parents to crossover are:

- Roulette-wheel selection
- Boltzmann selection
- Tournament selection
- Rank selection
- Steady state selection

The commonly used reproduction operator is the roulette-wheel selection method where a string is selected from the mating pool with a probability proportional to the fitness. Figure.3.2. shows a roulette-wheel for five individuals having different fitness values.

Since the third individual has a higher fitness value than any other, it is expected that the Roulette wheel selection will choose the third individual more than any other individual. This roulette-wheel selection scheme can be simulated easily.

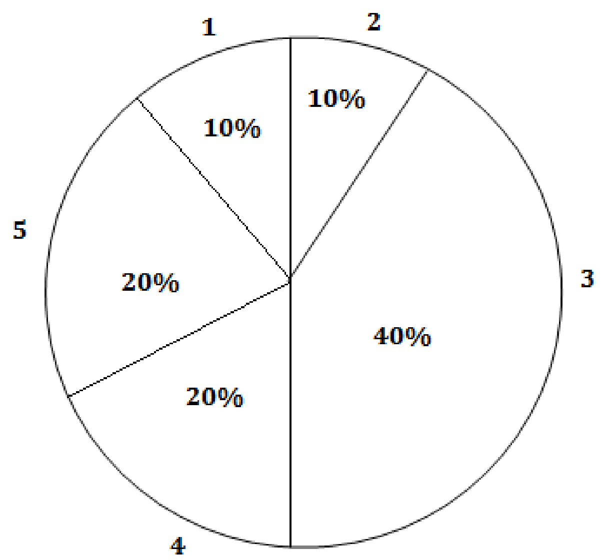

Figure.3.2 Roulette Wheel Selection

The tournament selection strategy provides selective fitness by holding a tournament competition among individuals. The best individual from the tournament is the one with the highest fitness which is the winner of individuals. Tournament competitor and the winner are then inserted into the mating pool. The tournament competition is repeated until the mating pool for generating new offspring is filled.

3.6.4.2 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 is expected from the crossover operator that good substrings from the parent strings will be combined to form a better child offspring. The aim of the crossover operator is to search the parameter space. Crossover is a recombination operator, which proceeds in three steps. First, the reproduction operator selects at random a pair of two individual string for mating, then a crossover site is selected at random along the string length and the position values are swapped between two string following the cross site.

Different forms of crossover are:

1.) Single point crossover
2.) Two point crossover
3.) Multi point crossover
4.) Uniform crossover

1.) Single point crossover: In the single point crossover, two individual strings are selected at random from the matting pool. Next, a crossover site is selected randomly along the string length and binary digits (alleles) are swapped between the two strings at crossover site. Suppose site 3 is selected at random. It means starting from the 4th bit and onwards, bits of strings will be swapped to produce offspring which is given in figure 3.3.

Parent 1: x1= { 0 1 0 **1 1 0 1 0 1 1** }
Parent 2: x2= { 1 0 0 **0 0 1 1 1 0 0** }

Offspring 1: x1= { 0 1 0 **0 0 1 1 1 0 0** }

Offspring 2: x2= { 1 0 0 **1 1 0 1 0 1 1** }

Figure.3.3. Single point crossover operation

2.) Two point crossover: In a two point crossover operator, two random sites are chosen and the contents bracketed by these sites are exchanged between two mated parents. If the cross site 1 is three and cross site 2 is six, the strings between three and six are exchanged which is shown in figure.3.4.

Parent 1: x1= {0 1 0 **1 1 0** 1 0 1 1}
Parent 2: x2= {1 0 0 **0 0 1** 1 1 0 0}

Offspring 1: x1= {0 1 0 **0 0 1** 1 0 1 1}
Offspring 2: x2= {1 0 0 **1 1 0** 1 1 0 0}

Figure.3.4 Two point crossover operation

3.) Multipoint crossover: In a multipoint crossover, again there are two cases. One is even number of cross sites and other is odd number of sites. For even number of sites the string is treated as a ring and cross sites are selected around the circle uniformly at random if the number of cross sites is odd, and then a different cross point is always assumed at the string beginning. For multipoint crossover, from m crossover positions along the string length, l are chosen at random with no duplicates and sorted into ascending order.

$$k_i \in \{1,2,.....,l-1\} \tag{3.7}$$

where

k_i is the i^{th} crossover point,

l is the length of the chromosome.

The bits between successive crossover points are exchanged alternatively between two parents to give two new offspring.

Suppose k € {3, 6, 9} is selected at random. It means the bits 4th, 5th, 6th are exchanged, bits 7th, 8th, and 9th of parent string are not exchanged and bits 10th, 11th, and12th of parent string are exchanged to produce offspring.

Parent 1: x1 = {000 **000** 000 **000**}
Parent 2: x2 = {111 **111** 111 **111**}

Offspring 1: x1 = {000 **111** 000 **111**}
Offspring 2: x2 = {111 **000** 111 **000**}

Figure.3.5 Multipoint crossover

4.) **Uniform crossover:** Single and multipoint crossovers define cross points as places within length of the string where a chromosome can be split. Uniform crossover generalizes this scheme to make every locus a potential crossover point. A crossover mask having same length as the chromosome structure is created at random and the parity of the bits in the mask indicates which parent will supply the offspring with which bits. The '1' in the random mask means bits swapping and the '0' means bit replicating.

3.6.4.3 Mutation

The final genetic operator in the algorithm is mutation. 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. In general evolution, mutation is a random process where one allele of a gene is replaced by another to produce a new genetic structure. Mutation is an important operation, because newly created individuals have no new inheritance information and the number of alleles is constantly decreasing. This process results in the contraction of the population to one point, which is wished at the end of convergence process. Diversity is one goal of the learning algorithm to search always in regions not viewed before. Therefore, it is necessary to enlarge the information contained in the population. One way to achieve this goal is mutation.

Mutation operator changes 1 to 0 at only one place in the whole string with a small probability and vice versa. Let mutation is done at location 3 the new offspring will be:

Offspring 1: x 1 =1 1 **1** 1 0 1 0

New offspring 1: x 2 = 1 1 **0** 1 0 1 0

Figure.3.6 Mutation operation

In general, the mutation probability is fixed throughout 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. It is not a primary operator but it ensures that the probability of searching any region in the problem space is never zero. This prevents complete loss of genetic material through reproduction and crossover **[39].**

3.6.5 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 used by researchers to tune the performance of Genetic Algorithms **[39].**

1.) **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.

2.) **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.

3.) **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.

3.6.6 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.

3.7 ALGORITHM FOR ELD USING GA

The step-wise procedure is outlined below:

1. Read data, namely cost coefficients, a_i, b_i, c_i, no. of iterations, length of string, population size, probability of crossover and mutations, power demand and $P^{\min}$ and $P^{\max}$.
2. Create the initial population randomly in the binary form.
3. Decode the string, or obtain the decimal integer from the binary string using Eq. (3.1)

4. Calculate the power generated from the decoded population by using Eq. (3.8)

$$P_i^j = P_i^{\min} + \frac{P_i^{\max} - P_i^{\min}}{2^l - 1} y_i^j \qquad (i = 1,2,....,NG; j = 1,2,...,L) \tag{3.8}$$

   where

   L is the number of strings or population size.

   y_i^j is the binary coded value of the i^{th} substring

5. Check P_i^j,

   If $P_i^j > P_i^{\max}$, then set $P_i^j = P_i^{\max}$

   If $P_i^j < P_i^{\min}$, then set $P_i^j = P_i^{\min}$

6. Find the fitness or cost function from Eq. (2.2a).
7. Find population with maximum fitness and average fitness of the population.
8. Perform the Reproduction Process, which includes the following steps:

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

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

   8(c) Select percentage of 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 / Σ Fitness's.

   8(d) Calculate cumulative sum (CS) to normalize the values between 0.0 and 1.0.

9. Perform Crossover Process:

   9(a) Choose a pair of random numbers between 0 and 1 to select one mother and one father chromosome, so as to produce new offspring.

   9(b) Pairing the chromosomes from different location, for different locations, crossover point has to be selected which can be selected randomly. Generate offspring by applying crossover.

10. Perform mutation by randomly selecting the mutation points from the total number of bits in the population matrix.
11. Update the population.
12. If the number of iterations reaches the maximum, then go to step 13. Otherwise, go to step 6.
13. The fitness that generates the minimum total generation cost is the solution of the problem.

CHAPTER 4: PARTICLE SWARM OPTIMIZATION

4.1 INTRODUCTION

Particle Swarm Optimization (PSO) is one of the modern heuristic algorithms, which can be effectively used to solve nonlinear and non-continuous optimization problems. It is a population-based search algorithm and searches in parallel using a group of particles similar to other AI-based optimization techniques.

Eberhart and Kennedy suggested a Particle Swarm Optimization (PSO) based on the analogy of swarm of bird and school of fish **[27]**. In PSO, each individual makes its decision based on its own experience together with other individual's experiences. Particle swarm optimization (PSO) is a population-based stochastic optimization technique, inspired by simulation of a social psychological metaphor instead of the survival of the fittest individual. In PSO, the system (swarm) is initialized with a population of random solutions (particles) and searches for optima using cognitive and social factors by updating generations. The particles are drawn stochastically toward the position of present velocity of each particle, their own previous best performance, and the best previous performance of their neighbours **[49], [50]**. PSO has been successfully applied to a wide range of applications, mainly in solving continuous nonlinear optimization problems.

PSO is a kind of evolutionary algorithm based on a population of individuals and motivated by the simulation of social behaviour instead of the survival of the fittest individual. It is a population-based evolutionary algorithm. Similar to the other population-based evolutionary algorithms, PSO is initialized with a population of random solutions. Unlike the most of the evolutionary algorithms, each potential solution (individual) in PSO is also associated with a randomized velocity, and the potential solutions, called particles, are then "flown" through the problem space.

The main advantages of the PSO algorithm are summarized as: simple concept, easy implementation, and computational efficiency when compared with mathematical algorithm and other heuristic optimization techniques **[24]**. The practical ELD problems with valve-point loading effects are represented as a non smooth optimization problem with equality and inequality constraints.

4.2 BASIC PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization has roots in two main component methodologies. Perhaps more obvious are its ties to artificial life (A-life) in general, and to bird flocking, fish schooling, and swarming theory in particular. It is also related, however, to evolutionary computation, and has ties to both genetic algorithms and evolution strategies. Particle swarm optimization comprises a very simple concept, and paradigms are implemented in a few lines of computer code. It requires only primitive mathematical operators, and is computationally inexpensive in terms of both memory requirements and speed. Early testing has found the implementation to be effective with several kinds of problems. Particle swarm optimization has also been demonstrated to perform well on genetic algorithm test functions, and it appears to be a promising approach for robot task learning.

The position of each particle is represented by P and also its velocity is expressed by V. Each particle knows its best value so far ($pbest$) and its position. Moreover, each particle also knows the best value so far in the group ($gbest$) among $pbests$. The modified velocity and position of each particle can be calculated using the current velocity and the distance from best previous position of each particle (Pb_{ij}) to best particle among all the particles in the group (G_j). Velocity and position of each particle can be modified by the following equation:

$$V_{ij}^{r+1} = wV_{ij}^{r} + C_1R_1(Pb_{ij}^{r} - P_{ij}^{r}) + C_2R_2(G_j^{r} - P_{ij}^{r})(i = 1,2,...,NP; j = 1,2,...,NG) \quad (4.1)$$

$$P_{ij}^{r+1} = P_{ij}^{r} + V_{ij}^{r+1} \qquad (i = 1,2,...,NP; j = 1,2,...,NG) \quad (4.2)$$

where

V_{ij}^{r} is the velocity of j^{th} member of i^{th} particle at r^{th} iteration,

$V_j^{min} \le V_{ij}^r \le V_j^{max}$

P_{ij}^r is the current position of j^{th} member of i^{th} particle at r^{th} iteration

w is the weighing function or inertia weight factor

C_1, C_2 are the acceleration constants

R_1, R_2 is random number between 0 and 1

NP is the number of particles in a group

NG is the number of members in a particle

The velocity is usually limited to a certain maximum value. PSO using Eq. (4.1) is called the *gbest* model. The particles in the swarm are accelerated to new positions by adding new velocities to their present positions. The new velocities are calculated using Eq. (4.1) and positions of the particles are updated using Eq. (4.2).

$$V_{ij}^{new} = wV_{ij} + C_1R_1(P_{ij}^{best} - P_{ij}) + C_2R_2(G_j^{best} - P_{ij})(i = 1,2,...,NP; j = 1,2,...,NG) \quad (4.3)$$

$$P_{ij}^{new} = P_{ij} + V_{ij}^{new} \qquad (i = 1,2,...,NP; j = 1,2,...,NG) \quad (4.4)$$

Suitable selection of inertia weight w provides balance between global and local explorations, thus requiring less iteration on average to find a sufficiently optimal solution. In general inertia weight is set according to the following equation:

$$w = w^{max} - \frac{w^{max} - w^{min}}{IT^{max}} * IT \quad (4.5)$$

where

IT^{max} is the maximum number of iterations(generation) and

IT is the current number of iterations.

The maximum and minimum velocity limit in the j^{th} dimension is computed as:

$$V_j^{\max} = \frac{P_j^{\max} - P_j^{\min}}{\alpha} \quad \text{and} \quad V_j^{\min} = -\frac{P_j^{\max} - P_j^{\min}}{\alpha} \tag{4.6}$$

Where α is the chosen number of intervals in the j^{th} dimension **[36]**.

4.3. FLOW CHART OF BASIC PSO

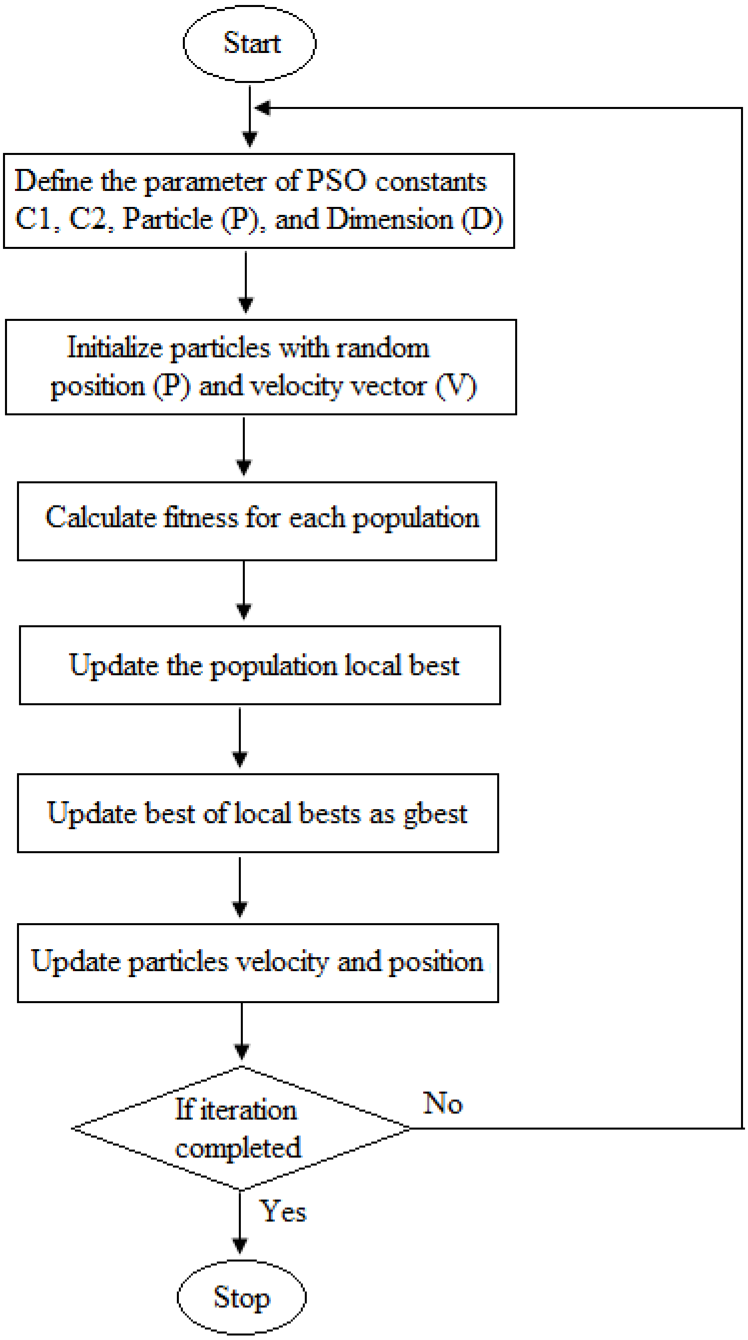

Figure.4.1 Flow chart of basic PSO

4.4 IMPLEMENTATION OF PSO FOR ELD SOLUTION

The main objective of ELD is to obtain the amount of real power to be generated by each committed generator, while achieving a minimum generation cost within the constraints. The details of the implementation of PSO components are summarized in the following subsection.

4.4.1 Representation of an Individual Population

For an efficient evolutionary method, the representation the population is important. Since the decision variables of the ELD problems are real power generations, the generation power output of each unit is represented as a gene, and many genes comprise an particle in the swarm. Each particle within the population represents a candidate solution for an ELD problem. For example, if there are d units that must be operated to provide power to loads, then the i^{th} particle can be defined as follows:

$$P_{gi} = [P_{i1}, P_{i2}, ...P_{id}] \quad (i = 1,2,....n) \tag{4.7}$$

Where, n means population size, d is the number of generator, P_{id} is the generation power output of d^{th} unit at i^{th} particle. The dimension of a population is $(n * d)$. These genes in each individual are represented as real values.

4.5 ALGORITHM FOR ECONOMIC LOAD DISPATCH USING PSO

The search procedure for calculating the optimal generation quantity of each unit is summarized as follows:

1. In the ELD problems the number of online generating units is the 'dimension' of this problem. The particles are randomly generated between the maximum and the minimum operating limits of the generators and represented using Eq. (4.7).
2. To each individual of the population calculate the dependent unit output from the power balance.

3. Calculate the evaluation value of each particle P_{gi} in the population using the evaluation function given by equation (2.2a).
4. Compare each particle's evaluation value with its $pbest$. The best evaluation value among them $pbest$ is identified as $gbest$.
5. Modify the Velocity of each particle by using the Eq. (4.1)
6. Check the velocity constraints of the members of each particle from the following conditions :

   If $V_{ij}^{r+1} > V_j^{\max}$, then $V_{ij}^{r+1} = V_j^{\max}$

   If $V_{ij}^{r+1} < V_j^{\min}$, then $V_{ij}^{r+1} = V_j^{\min}$

   Where, $V_j^{\min} = -0.5 P_j^{\min}$

   Where $V_J^{\max} = +0.5 P_j^{\max}$
7. Modify the position of each particle using the Eq. (4.2). P_{ij}^{r+1} must satisfy the constraints, namely the generating limits, described by equation (2.2c). If P_{ij}^{r+1} violates the constraints, then P_{ij}^{r+1} must be modified towards the nearest margin of the feasible solution.
8. If the evaluation value of each particle is better than previous $pbest$, the current value is set to be $pbest$. If the best $pbest$ is better than $gbest$, the best $pbest$ is set to be $gbest$.
9. If the number of iterations reaches the maximum, then go to step 10. Otherwise, go to step 2.
10. The individual that generates the latest $gbest$ is the optimal generation power of each unit with the minimum total generation cost.

CHAPTER 5: RESULTS AND DISCUSSION

In this section, the results of ELD after the implementation of proposed GA and PSO methods are discussed. The programs are implemented in MATLAB 7.6.0. The developed algorithms for ELD problem based on GA and PSO, have been discussed in chapter 3 and 4. The main objective is to minimize the cost of generation of plants using GA and PSO methods. The performance is evaluated without considering losses using two generator test systems, i.e. three generator test system and six generator test system, whose input data is given in Table 5.1 **[1], [51]** and table 5.5 **[52]** respectively.

5.1 CASE STUDY 1: 3 GENERATOR TEST SYSTEM

The coefficients of fuel cost and maximum and minimum power limits are given in Table 5.1. The power demand is considered to be 850 (MW). The results corresponding to GA and PSO are detailed in section 5.2 and 5.3 respectively, and the comparison of results of both methods is shown in table 5.4.

Table 5.1 Specifications for three generator test system

Unit no.	a_i	b_i	c_i	$P_i^{\min}$	$P_i^{\max}$
1	0.001562	7.92	561	150	600
2	0.00194	7.85	310	100	400
3	0.00482	7.97	78	50	200

5.1.1 Optimum Solution Using Genetic Algorithm for Case Study 1

The description of the results obtained by utilizing GA without considering losses is detailed herewith. The length of string in the population has been taken as 48 and the population size has been assumed as 10. The single string represents three substrings, each of 16 bits. Therefore, population we have taken is 10 x 48 in binary. The mutation probability is taken as 0.4. The crossover and mutation points are chosen randomly. In order to simulate the proposed GA method to solve ELD, a set of steps were produced which have been discussed in Chapter 3. The optimal result of GA is shown below in Table 5.2.

Table 5.2 Optimal Result of GA for Case Study 1

P_1 (MW)	393.0103
P_2 (MW)	319.2256
P_3 (MW)	137.7642
Total Power	850.00
Total Cost	8195.9790

5.1.2 Optimum Solution Using Particle Swarm Optimization for Case Study 1:

In order to simulate the proposed PSO method to solve ELD, a set of steps were produced which have been discussed in Chapter 4. The optimal result of PSO is shown below in Table 5.3.

Table 5.3 Optimal Result of PSO for Case Study 1

P_1 (MW)	387.9446
P_2 (MW)	340.52180
P_3 (MW)	121.5336
Total Power	850.00
Total Cost	8194.45

Table 5.4 Comparison of GA and PSO for Case Study 1

Technique	Total Cost
GA	8195.9790
PSO	8194.45

5.2 CASE STUDY 2: 6 GENERATOR TEST SYSTEM

The coefficients of fuel cost and maximum and minimum power limits are given in Table 5.5. The power demand is considered to be 450 (MW).). The results corresponding to GA and PSO are detailed in section 5.6 and 5.7 respectively, and the comparison of results of both methods is shown in Table 5.8.

Table 5.5 Specifications for Six Generator Test System

Unit No.	a_i	b_i	c_i	P_i^{min}	P_i^{max}
1	0.005	2.0	100	10	85
2	0.010	2.0	200	10	80
3	0.020	2.0	300	10	70
4	0.003	1.95	80	50	250
4	0.015	1.45	100	5	150
5	0.010	0.95	120	15	100

5.2.1 Optimum Solution Using Genetic Algorithm for Case Study 2:

The description of the results obtained by utilizing GA without considering losses is detailed herewith. The length of string in the population has been taken as 96 and the population size has been assumed as 20. The single string represents three substrings, each of 16 bits. Therefore, population we have taken is 20 x 96 in binary. The mutation probability is taken as 0.4. The crossover and mutation points are chosen randomly. The optimal result of GA is shown below in Table 5.6.

Table 5.6 Optimal Result of GA for Case Study 2

P_1 (MW)	65.197850
P_2 (MW)	29.951560

P_3 (MW)	11.010200
P_4 (MW)	161.028600
P_5 (MW)	95.403810
P_6 (MW)	87.40800
Total Power	450.0000
Total Cost	1971.0680

5.2.2 Optimum Solution Using Particle Swarm Optimization for Case Study 2

The optimal result of PSO is shown below in Table 5.7.

Table 5.7 Optimal Result of PSO for Case Study 2

P_1 (MW)	73.534480
P_2 (MW)	57.422580
P_3 (MW)	52.089710
P_4 (MW)	114.34110
P_5 (MW)	73.255040
P_6 (MW)	79.357100
Total Power	450.00000
Total Cost	1967.63600

Table 5.8 Comparison of GA and PSO for Case Study 2

Technique	Total Cost
GA	1971.0680
PSO	1967.6360

CHAPTER 6: CONCLUSIONS AND FUTURE SCOPE

6.1 CONCLUSIONS

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

Particle Swarm Optimization can be used to solve many of the same kinds of problems as genetic algorithms. This optimization technique does not suffer, however, from some of GA's difficulties: interaction in the group enhances rather than detracts from progress toward the solution. Further, a particle swarm system has memory, which the genetic algorithm does not have. Change in genetic populations results in destruction of previous knowledge of the problem, except when elitism is employed, in which case usually one or a small number of individuals retain their "identities." In particle swarm optimization, individuals who fly past optima are tugged to return toward them; knowledge of good solutions is retained by all particles. Particle swarm optimization has also been demonstrated to perform well on genetic algorithm test functions, and it appears to be a promising approach for robot task learning.

The effectiveness of the developed program is tested for three generator and six generator test system. The results obtained from these methods are also compared with each other. It is found that PSO is giving better results than GA.

6.2 SCOPE FOR FUTURE WORK

- Extend the problem for large number of units i.e., 30 or 90 or even higher units.
- Extend the problem by incorporating more than two objectives.
- Extend the GA and PSO based ELD solution by including the various Facts devices.

REFERENCES

1. Wood A. J. And Wollenberg B. F., “Power generation, operation and control”, John Wiley & Sons, New York, 1984.
2. Sinha Nidul, Chakrabarthi R. and Chattopadhyay P. K., “Evolutionary programming techniques for economic load dispatch”, IEEE Transactions on evolutionary computation, Vol. 7, pp.83- 94, 2003.
3. Nidul Sinha, Chakrabarti R. and P. K. Chatopadhyay, “Improved fast evolutionary program for economic load dispatch with non-smooth cost curves”, IE (I) Journal. EL Vol. 85, 2004.
4. Mori Hiroyuki and Horiguchi Takuya, “Genetic algorithm based approach to economic load dispatching”, IEEE Transactions on power systems, Vol. 1, pp. 145-150, 1993.
5. Chiang C. L., “Genetic based algorithm for power economic load dispatch”, IET Gener. Transm. Distrib, 1, (2), pp. 261-269, 2007.
6. Ling S. H., Lam H. K., Leung H. F. and Lee Y. S., “Improved genetic algorithm for economic load dispatch with valve- point loadings”, IEEE Transactions on power systems, Vol. 1, pp. 442 – 447, 2003.
7. Ouiddir R., Rahli M. and Korid L. Abdelhakem, “Economic dispatch using genetic algorithm: application to western Algeria’s electrical power network”, Journal of Information Science and Engineering 21, pp. 659 – 668, 2005.
8. Younes Mimoun and Rahli Mostafa, “Economic power dispatch using the combination of two genetic algorithms”, Journal of Electrical and Electronics Engineering, Vol. 6, pp. 175-181, 2006.
9. Achayuthakan Chira and Srivastava S. C., “A genetic algorithm based economic load dispatch solution for eastern region of egat system having combined cycle and cogeneration plants”, IEEE Transactions on power systems, Vol. 1, pp. 165-170, 1998.
10. Younes Mimoun, Mostefa Rahli and Koridak Lahouri Abdelhakem, “Economic power dispatch using Evolutionary Algorithm”, Journal of Electrical Engineering Vol. 57, No. 4, pp. 211-217, 2006.

11. Park June Ho, Yang Sung Oh, Lee Hwa Seok and Young Moon Park, "Economic load dispatch using evolutionary algorithms", IEEE Transactions on power systems, Vol. 2, pp. 441-445, 1996.
12. Kumaran Giridhar and Mouly V. S. R. K., "Using evolutionary computation to solve the economic load dispatch problem", IEEE Transactions on power systems, Vol. 3, pp. 296-301, 2001.
13. Park J. H., Yang S. O., Mun K. J., Lee H. S. and Jung J. W., "An application of evolutionary computations to economic load dispatch with piecewise quadratic cost functions", IEEE Transactions on power systems, Vol. 69, pp. 289-294, 1998.
14. Fink, L. H., Kwatny H. G. And Mcdonald J. P., "Economic dispatch of generation via valve-point loading", IEEE Transactions on Power Apparatus And Systems, Vol. Pas-88, No. 6, June 1969.
15. Reid Gerald F. and Hasdorff Lawrence, "Economic dispatch using quadratic programming", IEEE Transaction on Power Apparatus and Systems, Vol. PAS-92, pp. 2015-2023, 1973.
16. Megahed I., Abou-Taleb N. and Iskandrani M. , "A modified method for solving the economic dispatching problem", IEEEE Transactions on Power Apparatus and Systems, Vol. PAS-96 (I), pp. 124-133, January/February, 1977.
17. Happ H. H., "Optimal power dispatch - a comprehensive survey", IEEE Transactions on Power Apparatus an8 Systems, Vol. PAS-96, no. 3, May / june1977.
18. Kwatny H. G. and Athay T. A., "Coordination of economic dispatch and load frequency control in electric power systems," Proceedings, 18th IEEE Conference on Decision and Control, 1979.
19. Ross D. W. and Kim S., "Dynamic economic dispatch of generation," IEEE Transactions on Power Apparatus and Systems, Vol. PAS-99 (6), pp. 2060-2068, November/December, 1980.
20. Bottero M. H., Galiana F. D., and Fahmideh-Vojdani A. R., "Economic dispatch using the reduced hessian", IEEE Transactions on Power Apparatus and Systems, Vol. Pas-I01 pp. 3679-3688, October, 1982.
21. Aoki K. and Satoh T., " Economic dispatch with network security constraints using parametric quadratic programming", IEEE Transactions on Power Apparatus and Systems, Vol. PAS-101, No. 12 December 1982.

22. Lin C. E. and Viviani G. L., “Hierarchical Economic Dispatch for Piecewise Quadratic Cost Functions”, IEEE Transactions on Power Apparatus and Systems, Vol. PAS-103, No. 6, June 1984.
23. Ramanathan R., “Fast economic dispatch based on the penalty factors from Newton' s method”, IEEE Transactions on Power Apparatus and Systems, Vol. PAS-104, No. 7, July 1985.
24. Chowdhury B. H. and Rahman S., “A review of recent advances in economic dispatch”, IEEE Transactions on Power Systems, Vol. 5, No. 4, November 1990.
25. Walters David C. and Sheble Gerald B., “Genetic algorithm solution of economic dispatch with valve point loading”, IEEE Transactions on Power Systems, Vol. 8, No. 3, August 1993.
26. Chen Po-Hung and Chang Hong-Chan, “Large-scale economic dispatch by genetic algorithm”, IEEE Transactions on Power System. Vol. 10. No. 4. November 1995.
27. Eberhart R. C. and Kennedy J., “A new optimizer using particle swarm theory”, Proceedings of the 6th. International Symposium on Micro Machine and Human Science, Nagoya, Japan, pp. 39-43, 1995.
28. Orero S. O. and Irving M. R., “Economic dispatch of generators with prohibited operating zones: a genetic algorithm approach”, IEE Proc.- Gener. Transm. Distrib., Vol. 143, No. 6, November 1996.
29. Wang Xian, Li Yu- Zeng and Zhang Shao- Wa, “A new neural network approach to economic emission load dispatch”, Proceedings of the First International Conference on Machine Leaning and Cybernetics, Beijing, 4-5 November 2002.
30. Venkatesh P., Gnanadass R., and Padhy Narayana Prasad, “Comparison and application of evolutionary programming techniques to combined economic emission dispatch with line flow constraints”, IEEE Trans. on Power Systems, Vol. 18, no. 2, may 2003.
31. Madouh J. Y. and El- Hawary M. E., “Economic dispatch of all thermal power systems with fuzzy load”, Large Engineering Systems Conference on Power Engineering (LESCPE), pp. 7-18, 2004.
32. Zhang Guo- Li, Lu Hai- Yan, Li Geng- Yin and Zhang Guang- Quan, “Dynamic economic load dispatch using hybrid genetic algorithm and the method of fuzzy number ranking”, Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, 18-21 August 2005.

33. Kumari M. Sailaja and Sydulu M., “A fast computational genetic algorithm for economic load dispatch”, International Journal of Recent Trends in Engineering Vol. 1, No. 1, May 2009.
34. Ting- Fang Y. U and Chun- Hua Peng, “Application of an improved particle swarm optimization to economic load dispatch in power plant”, 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), Vol. 2, pp. 619-624, 2010.
35. Abido M. A, “Particle swarm optimization for multi machine power system stabilizer design”, Proceedings of IEEE Power Engineering Society Summer Meeting IEEE Press, 2001.
36. Kothari D. P. and Dhillon J. S., “Power System Optimization”, Second Edition, PHI learning Private limited, 2011.
37. Kothari D. P. and Nagrath I. J., “Modern Power System Analysis”, Tata McGraw-Hill, Third Edition, 2003.
38. Wong K. P. and Wong Y. W., “Genetic and genetic simulated annealing approaches to economic dispatch”, IEE Proceedings C., 685–692, 1994.
39. Chang C. S. and Lern L. P., “Application of Tabu Search Strategy in Solving Non-Differentiable Savings Function for the Calculation of optimum Savings due to Shunt Capacitor Installation in a Radial Distribution System”, IEEE, pp. 2323-2328, 2000.
40. Central Station Engineers of the Westinghouse Electric Corporation, “Electrical Transmission and Distribution Reference Book”, Fourth Edition, Westinghouse Electric Corporation, Pennsylvania, United States of America, 1964.
41. Salami M., and Cain G., "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.
42. Abdullah W. N. W., Saibon H., Zain A. A. M. and Lo K. L., “Genetic Algorithm For Optimal Reactive Power Dispatch”, IEEE Catalogue No. 98EX137, 1998.
43. Levitin G., Kalyuzhny A., Shenkman A. and M. Chertkov, “Optimal Capacitor Allocation in Distribution Systems Using a Genetic Algorithm and a Fast Energy loss Computation Technique”, IEEE Trans. Power Delivery, Vol. 15, No. 2, pp. 623-628, April 2000.
44. Mantawy A. H. and Al-Ghamdi M. S., “A New Reactive Power Optimization Algorithm”, IEEE Conference paper, Bologna, Italy, June 2003.

45. Damousis J. G., Unkirlzis A. G. and Dokopoulos P. S., "Network- constrained economic dispatch using real-coded genetic algorithm", IEEE Trans. Power Syst. 8 (November (4)) (1995) 198-205.
46. Deb K., "Optimization for engineering design", prentice hall of India, 1995.
47. Goldberg D.B., "Genetic algorithm in search optimization and machine learning", D. E. 1989.
48. Mohan N., Undeland T. M. and Robbins W. P., "Power Electronics, Converters, Applications and Design," John Wiley & Sons, United States of America, 2003.
49. Kennedy J. and Eberhart R. C., "Particle swarm optimization", Proceedings of IEEE International Conference on Neural Networks (ICNN'95), Vol. IV, pp. 1942-1948, Perth, Australia, 1995.
50. Kennedy J. and Eberhart R. C., "Swarm Intelligence", San Francisco, CA: Morgan Kaufmann, 2001.
51. Ah King R. T. F. and Rughooputh H. C. S., "Elitist Multiobjective Evolutionary Algorithm for Environmental/Economic Dispatch", IEEE Congress on Evolutionary Computation, Canberra, Australia, vol. 2, pp. 1108-1114, 2003.
52. Dhillon J. S., Parti S. C. and Kothari D. P., "Multi objective stochastic optimal thermal power dispatch", ibid, pp. 136-140, 1987.