ECONOMIC LOAD DISPATCH USING FUZZY LOGIC CONTROLLED GENETIC ALGORITHM

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

Master of Engineering

in

Power Systems & Electric Drives

Thapar University, Patiala

By:

Satyendra Pratap Singh

(Regn. No. 80741021)

Under the supervision of:

Ms. Suman Bhullar

Lecturer, EIED

JUNE 2009

ELECTRICAL & INSTRUMENTATION ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

DEDICATED

TO MY

PARENTS

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "**Economic Load Dispatch using Fuzzy Logic Controlled Genetic Algorithm",** in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Power Systems & Electric Drives* submitted in Electrical & Instrumentation Engineering Dapartment of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of Ms. Suman Bhullar, Lecturer, EIED.

The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.

SPSingh 14/7/09

(Satyendra Pratap Singh)

Regn.No. 80741021

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

Bhullar

(Ms. Suman Bhullar)

Electrical & Instrumentation Engg. Department

Thapar University

Patiala

Countersigned by:

S-Ghosh 15/7/09

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

RKSharma 21/7/09

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

ii

ACKNOWLEDGEMENT

First of all, I thank the Almighty God, who gave me the opportunity and strength to carry out this work.

I would like to thank **Ms. Suman Bhullar, Lecturer, EIED** for the opportunity to work with her, and also for her encouragement, trust and untiring support. **Ms. Suman Bhullar** has been an advisor in the true sense both academically and morally throughout this project work.

I take this opportunity to express my gratitude and sincere thanks to **Dr. Smarajit Ghosh, Prof. & Head, EIED** for their valuable suggestion and for providing me the opportunity to complete my thesis work simultaneously.

The paucity of words does not compromise for extending my thanks to my parents, faculty of EIED and friends of M.E. (PSED, EIC) whose uninterrupted love, inspiration and blessings helped me in completing this research report.

Satyendra Pratap Singh

Regn.No.80741021

ABSTRACT

The sizes of the electric power systems are increasing rapidly to meet the energy requirements. A number of power plants are connected in parallel to supply the system load by interconnection of power station. With the development of integrated power system, it becomes necessary to operate the plant units economically. Thus evolves Economic Load Dispatch (ELD) problem.

The ELD problem in a power system is to determine the optimal combination of power outputs for all generating units which will minimize the total fuel cost while satisfying all practical constraints. To solve the problem of ELD there are many methods (traditional) like Lambda Iterative Method (LIM), Newton's Linear Programming etc. But all those methods are based on the assumption of continuity and differentiability of cost function. Practically, the real power limits of the generators vary between minimum and maximum limits. Hence the optimization of real time ELD problem becomes more non-linear.

In this thesis, fuzzy logic has been applied in combination with Genetic Algorithm (GA) to solve various power system problems. For smooth and better convergence in GA, the crossover probability and mutation rate are adjusted by fuzzy logic strategy leading to an improved GA technique termed as Fuzzy Logic Controlled Genetic Algorithm (FCGA). The proposed FCGA can be applied to a wide range of optimization problem. The computational results reveal that the proposed algorithm has excellent convergence characteristics and is superior to the GA and LIM.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW

The efficient and optimum economic operation of electric power systems has always occupied an important position in electric power industry. In recent decades, it is becoming very important for utilities to run their power systems with minimum cost while satisfying their customer demand all the time and trying to make profit. With limited availability of generating units and the large increase in power demand, fuel cost and supply limitation, the committed units should serve the expected load demand with the changes in fuel cost and the uncertainties in the load demand forecast in all the different time intervals in an optimal manner.

The basic objective of ELD of electric power generation is to schedule the committed generating unit outputs, so as to meet the load demand at minimum operating cost while satisfying all unit and system equality and inequality constraints **[1]**. The ELD problem has been tackled by many researchers in the past **[2]**. ELD problem involves different problems. The first is Unit Commitment or pre-dispatch problem where in it is required to select optimally out of the available generating sources to operate to meet the expected load and provide a specified margin of operating reserve over a specified period of time.

The second aspect of ELD is on-line economic dispatch where in it is required to distribute the load among the generating units actually parallel with the system in such a manner as to minimize the total cost of supplying power. In case of ELD, The generations are not fixed but they are allowed to take values again within certain limits so as to meet a particular load demand with minimum fuel consumption.

1.2 LITERATURE SURVEY

Dhillon et. al [3] formulated the problem as multiobjective one. They considered objectives such as operating cost, minimal emission and minimum transmission losses in thermal power dispatch systems, considering uncertainties and inaccuracies in system data. The validity and effectiveness of the method was demonstrated by analysing a 6-generator case.

Coonick A.H., Knight U.G. [4] proposed a new improvement in the decision trees (DT) technique by adding fuzzy logic to the unit limits and load. Furthermore, the generating cost obtained via Fuzzy Logic Decision Trees (FLDT) is lower than that obtained with the classical formulation due to the fuzzyfying of the generating unit limits. This shortens the time needed to reduce the system generation cost in a controlled and formal manner. By using a fuzzy demand, it was also possible to find an expected generating cost and its corresponding standard deviation.

Palanichamy and Shrikrishna K., [5] discussed Simple algorithm for economic power dispatch for optimizing the problem while satisfying a set of system operating on straints, including constraints dictated by the electric network. ELD has been widely used in power system operation and planning discussed by Wood and Woolenberg in **[6].** Many approaches **[42]** have been listed to formulate and solve this problem. These approaches include combining emission dispatch with the economic load dispatch **[7-11]**, includes use of Hopfield Neural Network **[13-23]**, Fuzzy approach **[32,34,37,43]**, Evolutionary Algorithms**[12,25-26,44-49]** and hybrid methodologies**[26,36,43,53-54].** Evolutionary Algorithms constitutes Genetic algorithms **[43-47]**, Particle Swarm Optimisation **[50-52]** and Simulated Annealing **[60].**

Chen Po- Hung & chan hong [11] this paper proposed a gentic approach for solving the economic dispatch problem in large scale system. A new coding technique for solving the ELD solution is developed in this paper. It is faster than lambda iteration method.

Bouzeboudja Hamid, Chaker Abdelkader and Alali Ahmed [12] proposed about the economic load dispatch using the real coded genetic algorithm (RCGA).The use of real valued representation in the GAs gives number of advantages in numerical function optimization over binary encoding, the efficiency of GAs is increased as there is no need to convert chromosomes to the binary type,less memory is required.

Ongsakul W. [16] proposed a genetic algorithm based on merit order loading solution to solve dynamic economic dispatch problem for combined cycle units with linear decreasing & decresing incrementel cost function.with different migration strategies, the proposed method compromises the solution quality and speed up upper bounds for the best performance.

Subburaj P., Ganesan L., Ramar K. and Rajkumar [33] presented the EP based ELD problem with stochastic method for competition and selection. The conventional optimization methods require the objective functions in continuous differentible form, therefore they fail to provide global minima. The Evolutionary Computation (EC) methods can handle non-differentiable and non convex objective functions and give global or near global optimum solutions. Evolutionary Computation methods such as GA, EP are applied to ELD problem. By avoiding the coding and decoding process of transformations in GA,use of EP among other EC methods resulted in less population size and number of iterations. This study project a new approch and it was developed in such a way that a stochastic optmization technique evolutionary programing is used to solve ELD problem with non smooth fuel cost functions.

Lee F. & Aggarwal R.K. [34] proposed hybrid scheme is constracted in such a way that a genetic algorithm performs a base level search, makes rapid decisions to make the local gradient technique to find the potential hill. The proposed method also ensure the dispatch quality as well as speed by allowing a loose match between a power generation and load demand at the base search , and compensate for any mismatch at the begning of local search.

Yalcinoz T., Altun H. and Uzam M [36] presented the new genetic approach based on arithmetic crossover which improves the quality of the ELD solution. The new genetic approach is compared with an improved Hopfield NN, a fuzzy logic controlled genetic algorithm, an advance engineered-conditioning genetic approach and an advance Hopfield NN approach. In this paper , elitism, arithmetic crossover which defines a linear combination of two cromosome and mutation are used in the genetic algorithm to generate succesive sets of possible operating policies.

Sepulveda C.A.Roa, Herrera M., Coonick A.H. [37] presented the economic load dispatch using the fuzzy decision trees, this paper presents a new improvement in the DT technique by adding fuzzy logic (FL) to the unit limits and load (FLDT). By doing so, the numerical convergence of the overall technique improves. Furthermor, the generating cost obtained via FLDT is lover than that obtained with the classical formulation due to the fuzzyfyingof the generating unit limits. By using a Fuzzy demand, it was also possible to find an expected generating cost and its corresponding standard deviation.

Dang Chuangyin & Li Minqiang [39] This paper proposed floating point genetic algorithm to solve the unit commitment problem. Based on the characteristic of typical load demand, a floating point chromosome representation and encoding-decoding scheme are designed to reduce the complexcity in handling minimum up/ down time limit. Penalty factors are defined which blend the emission costs with the fuel costs. The familiar quadratic form of objective functions used which gives the optimal dispatch directly. The capacity limits (lower and upper) of plants treated as the operating constraints and the total generation which is a function of load plus transmission losses is considered as the demand constraint.

A.Bakirtzis, V. Petridis & kazarlis S. [40] presented the two genetic algorithm solutions to ELD problem. Both of them outerform the dynamic programing (DP) solution to the problem. The DP gives the general solution to the economic dispatch problem and does not require any convexity assumptions. The basic disadvantage of the GAs is the fact that they may miss the optimum and provide a near-optimum solution in a limited runtime period.

Hazra Jagabondhu and Sinha Avinash [41] presented a comparative study of four different evolutionary algorithms i.e. genetic algorithm, bacteria foraging optimization, ant colony optimization and particle swarm optimization for solving the economic dispatch problem. All the methods are tested on IEEE 30 bus test system.

Song Y.H., Wang G.S. [**43**] proposed an application of a fuzzy logic controlled genetic algorithm to economic dispatch. The improved genetic algorithm with two fuzzy controllers based on some heuristics to adaptively adjust the crossover probability and mutation rate during the optimisation process.The proposed method can be applied to wide range of optimization problems.

Benasla L., Belmadant Abderrahim and Rahli Mostefa [55] proposed a New Economic Dispatch Problem Formulation (NEDPF) to solve Economic Dispatch Problem (EDP). This new formulation was based on the reduction of the number of variables (number of generators) and elimination of the equality and inequality constraints, thus the transformation of the constrained non linear programming problem to an unconstrained one. The new unconstrained objective function, is minimized by Hooke-Jeeves' method. The advantage of using Hooke-Jeeves' method to minimize the new unconstrained objective function is its simplicity and no necessity of gradients.

Kumar Sushil and Naresh R. [56] proposed an efficient optimization technique based on genetic algorithm for solution of economic load dispatch problem with continuous and non smooth cost function and with various constraints being considered. The effectiveness proposed algorithm has been demonstrated on different system considering transmission loss in thermal power plant.

Mahdad Belkacem, Bouktir Tarek [57] proposed a hybrid Genetic Algorithm and Fuzzy Logic rules for solving the economic dispatch problem under constrained emission with multi shunt Flexible AC Transmission Systems (FACTS). This approach proposed a flexible Genetic Algorithm which based on fuzzy logic rules with the ability to adjust continuously the crossover and mutation parameters.

Nanda J. & Badri R. [59] this paper discussed the solution of ELD with line flow constraints through the application of genetic algorithm. The most important advantages of GA is that they use only the pay of information hence independent of nature of search space such as smoothness , convexity.

Holland [63] invented GA in 1970's, is a stochastic global search method that mimics the metaphor of natural biological evolution. GA's operate on a population of candidate's solution encoded to string called chromosomes in order to obtain optimality. Each chromosome exchanges information by using string operators borrowed from natural genetic to produce the better solution. Although GA seems to be a good method to solve optimization problem, sometimes the solution obtained from GA is only a near global optimum solution. GA and EP comes under the category of EA. In EP, invented by Fogel **[61-62]**, the state of the machine is mutated**.** Therefore, EP has been become an optimal tool and was used in many practical problems.

1.3 SCOPE OF WORK

The scope of the thesis work are summarised as follows

1. Main objective of project is to reduce the total cost of generation while satisfying the power generation limits.
2. Use global search techniques like GA, FCGA to find the optimal solution.
3. Investigate the effectiveness of these methods for ELD problem while considering the transmission losses.
4. Compare the result obtained from the FCGA method with the results obtained from GA and LIM.

1.4 ORGANIZATION OF THE THESIS

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

Chapter 1.

This chapter presents the brief introduction, literature review by various researchers.It also includes the scope of the work and organization of the thesis.

Chapter 2.

This chapter covers the Economic Load Dispatch formulation and Economic Load Dispatch solution using Lambda Iteration method

Chapter 3.

This chapter entitled as "Theory of Genetic Algorithm and Fuzz Logic" which includes the basic theory of both methods. Also presents the mapping of Economic Load Dispatch problem in GA also in Fuzzy Logic.

Chapter 4.

This chapter presents the application of a fuzzy logic controlled genetic algorithm to ELD, also present the flow chart of ELD using FCGA.

Chapter 5.

This chapter presents the detail of results pertaining to various cases and the comparison of results obtained for various methods.

Chapter 6.

This chapter entitled as "Conclusion and Future Scope" mainly includes the conclusion and future scope of this thesis work.

CHAPTER 2
ECONOMIC LOAD DISPATCH

In this chapter, the Basic concepts of Economic Load Dispatch (ELD), problem formulation and solution using Lambda Iteration Method (LIM) are summarized in brief.

2.1 BASIC THEORY OF ELD

The efficient and optimum economic operation and planning of electric power generation systems have always occupied an important position in the electric power industry. In the power system, transmission networks are interconnected through tie lines. Hence the utilities may interchange power, share reserve and render assistance to one another at the time of need. Since the sources of energy are so diverse, so the choice of the required sources is made on economic, technical and geographical basis. As there are few facilities to store electrical energy, the net production of a utility must clearly track its total load.

The Economic Dispatch 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. The objective of ELD is to minimise the overall cost of generation.

2.1.1 FUEL COST FUNCTION

The components of the cost that fall under the category of dispatching procedures are the costs of the fuel burnt in the fossile plant because nuclear plants tend to be operated at constant output levels and hydro plants have essentially no variable operating costs. The total cost of operation includes the fuel cost, costs of labour, supplies and maintenance. Generally, costs of labour, supplies and maintenance are fixed percentages of incoming fuel costs.

We assume that the variation of fuel cost of each generator (Fi) with the active power output (Pi) is given by a quadratic polynomial

$$F_i(P_i) = \sum_{i=1}^{NG} (a_i P_i^2 + b_i P_i + c_i) \qquad \frac{Rs}{hr} \qquad (2.1)$$

Where,

F_i = fuel cost of generator i.

P_i = power output of generator i

a_i = measure of losses in the system.

b_i = represents the fuel cost.

c_i = includes salary and wages, interests and depreciation and is independent of generation

NG = number of generation buses

Input of thermal plant is generally measured in Btu/hr and the output is measured in MW. A simplified input output curve of the thermal unit known as heat rate curve is given in following fig. 2.1(a). Converting the ordinate of heat rate curve from Btu/hr to Rs/hr, Results in the operating cost curve shown in fig. 2.1(b).

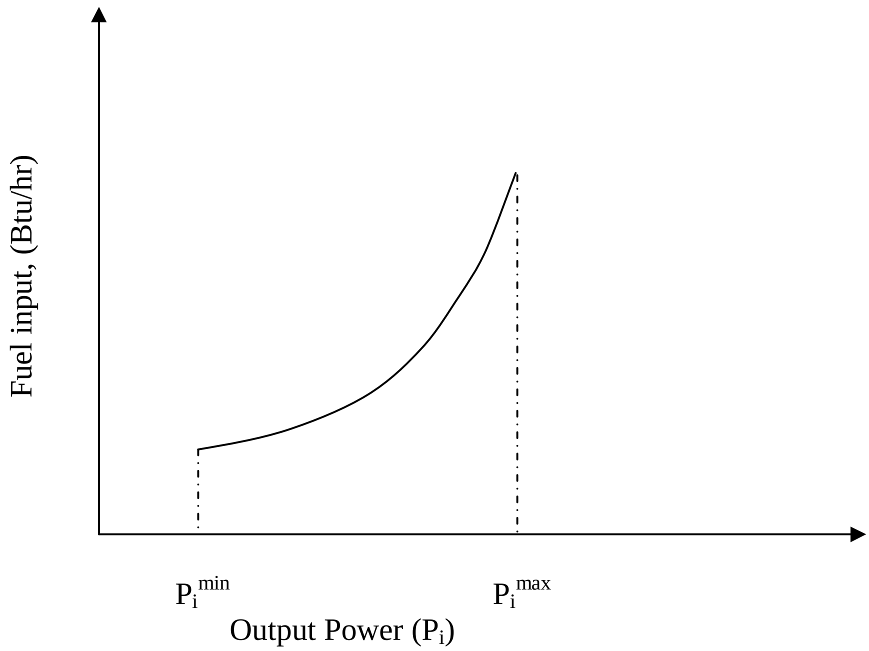

Fig. 2.1(a). Heat- rate curve of a fossile fired generator

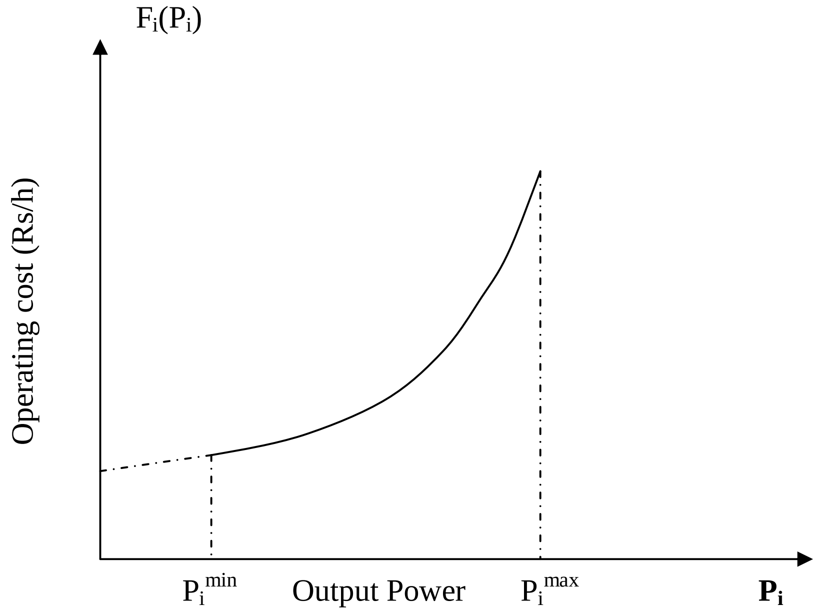

Fig 2.1(b). Operating costs curve of a fossile fired generator

Where P_i^{min} is the minimum loading limit below which it is uneconomical to operate the unit and P_i^{max} is the maximum output limit.

2.1.2 INCREMENTAL FUEL COST

In economic evaluation, there is a great tendency to resort to average rates to arrive at a total cost. Extreme caution should be exercised in doing this, especially in dealing with unit fuel cost of energy produced by a generating station. The additional fuel cost of energy produced depends entirely upon the manner in which the generation is added. The input and output curve of generating units of thermal plants is shown in fig 2.2, The abscissa is output power P_i in MW, x-ordinate as fuel (heat) input in joules per hours of the ith unit. The ordinate of curve may be converted to fuel cost in F_i Rs/hr by multiplying the fuel input by the fuel is Rs/joule.

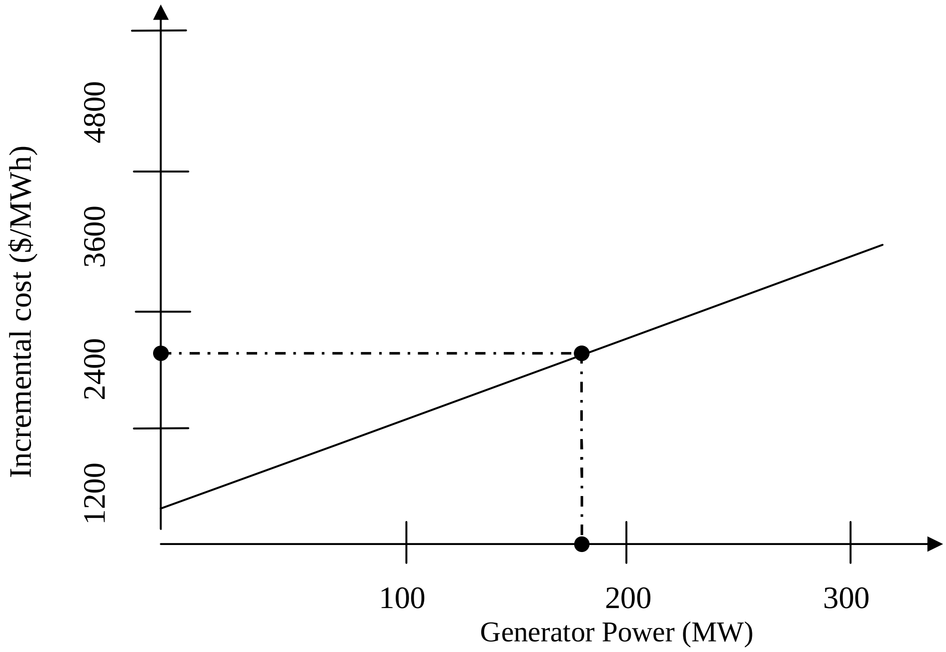

Fig 2.2 Incremental fuel-cost curve (thermal plants)

The slope of cost curve at a point M is given by

$$\tan\theta = \frac{\Delta F_i}{\Delta P_i} \qquad (2.2)$$

Where, ΔF_i = increase in fuel cost corresponding to an increase of power output ΔP_i. The increment fuel cost for a generator for any given electrical power output is defined as the limiting value of the ratio of the increase in cost of fuel in Rs/h to the corresponding increase in electrical power output tends to zero.

2.2 DERIVATION OF COORDINATION EQUATION

The ELD problem is defined as to minimize the total operating cost of a power system while meeting the total load plus transmission losses within generator limits. Mathematically the problem is defined as (including losses)

Minimize:

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

Subject to (1) the energy balance equation

$$\sum_{i=1}^{NG} P_i = P_D + P_L \qquad (2.3)$$

(2) the inequality constraints

$$P_{i(min)} \leq P_i \leq P_{i(max)} \qquad (2.4)$$

Where,

a_i, b_i, c_i : cost coefficients

P_D : load demand

P_i : real power generation

P_L : power transmission loss

NG : number of generation busses

One of the most important, simple but approximate method of expressing transmission loss as function of generator powers is through B-coefficients. This method uses the fact that under normal operating condition, the transmission loss is quadratic in the injected bus real power.The general form of the loss formula using B-coefficient is

$$P_L = \sum_{i=1}^{NG}\sum_{j=1}^{NG} P_i B_{ij} P_j \quad \text{MW} \qquad (2.5)$$

where,

P_i, P_j : real power injection at the ith, jth buses

B_{ij} : loss coefficients which are constant under certain assumed conditions.

The above loss formula is known as the George's formula. Another more accurate form of transmission loss expression, frequently known as the Kron's loss formula is

$$P_L = B_{00} + \sum_{i=1}^{NG} B_{i0} P_i + \sum_{i=1}^{NG}\sum_{j=1}^{NG} P_i B_{ij} P_j \quad \text{MW} \qquad (2.6)$$

where, B_{00}, B_{i0}, and B_{ij} are the loss coefficient which are constant under certain assumed conditions.

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

Using Lagrange multipliers, an augmentes function is defined as

$$L(P_i,\lambda) = F(P_i) + \lambda\left(P_D - P_L - \sum_{i=1}^{NG} P_i\right) \qquad (2.7)$$

Where, λ is the Lagrangian multiplier.

Necessary conditions for the optimization problem are

$$\frac{\partial L\,(P_i,\,\lambda)}{\partial P_i} = \frac{\partial F(P_i)}{\partial P_i} + \lambda\left(\frac{\partial P_L}{\partial P_i} - 1\right) = 0 \qquad ,(i = 1, 2, 3 \ldots\ldots\ldots\ldots NG) \qquad (2.8)$$

Rearranging the above equation

$$\partial F(P_i)/\,\partial P = \lambda_i\left(1 - \frac{\partial P_L}{\partial P_i}\right) \qquad , (i = 1,2, \ldots\ldots\ldots NG) \qquad (2.9)$$

Where,

$\frac{\partial F(P_i)}{\partial P_i}$: Incremental cost of the i^{th} generator (Rs/MW h)

$\frac{\partial P_L}{\partial P_i}$: Incremental transmission losses.

Equation 2.9 is known as the exact coordination equation and

$$\frac{\partial L(P_i,\lambda)}{\partial \lambda} = P_D + P_L - \sum_{i=1}^{NG} P_i = 0 \qquad (2.10)$$

By differentiating the transmission loss Eq. 2.5 with respect to P_i, the increamental transmission loss can be obtained as

$$\frac{\partial P_L}{\partial P_i} = \sum_{j=1}^{NG} 2\, B_{ij} P_j \qquad (2.11)$$

and by differentiating the cost function of Eq. 2.1 with respect to P_i , the incremental cost can be obtained as

$$\frac{\partial F(P_i)}{\partial P_i} = 2a_i P_i + b_i \qquad (i = 1, 2, \dots \dots, NG) \qquad (2.12)$$

to find the solution

$$2a_i P_i + b_i = \lambda\left(1 - 2\sum_{i=1}^{NG} B_{ij} P_j\right) \qquad (i = 1, 2, \dots \dots, NG) \qquad (2.13)$$

Rearranging the equation 2.13 to get P_i, i.e.

$$2a_i \mathbf{P_i} + b_i = \lambda(1 - 2b_{ij} P_i - \sum_{j=1}^{NG} 2B_{ij} P_j) \qquad (i = 1,2, \dots \dots NG) \text{ or}$$

$$2\left(a_i + \lambda B_{ij}\right)\mathbf{P_i} + 2\sum_{j=1}^{NG} B_{ij} P_i = (\lambda - b_i) \qquad (i = 1, 2, \dots \dots, NG) \qquad (2.14)$$

The above linear equation 2.14 can be solved to obtain the value of P_i if λ is known.

2.3 LAMBDA ITERATION METHOD FOR SOLUTION OF THE ELD PROBLEM

The lambda-iteration method (LIM) is, so far, the most popular method for the solution of the economic load dispatch **[6]**. It gives a decentralized solution to the ELD problem by equating the marginal cost of generation of each thermal unit to the price of electricity, or, equivalently, the marginal revenue of each unit under perfect competition conditions, known as system lambda. LIM is the iterative method by which the optimum lambda value that leads to satisfaction of the system power balance constraint is determined. Using the above economic interpretation of λ, whenever the choice of λ is such that it leads to insufficient generation, λ must be increased so as to make it beneficial for the generators to increase production. Whenever there is excess generation, λ must be decreased to limit generation. There is a number of ways the LIM can be implemented. In this thesis, quadratic cost functions is used for the generating units which is given in Eq. 2.1and a binary search for the optimum value of lambda. That is, the minimum and maximum lambda values are initially computed,

$$\lambda_{min} = \min_{i=1,n}\left\{\frac{dF_i(P_{i,min})}{dP_i}\right\} \tag{2.15}$$

$$\lambda_{min} = \min_{i=1,n}\left\{\frac{dF_i(P_{i,min})}{dP_i}\right\} \tag{2.16}$$

The initial value chosen for lambda is the mid-point of the interval (λ_{min}, λ_{max}), i.e,

$$\lambda = \frac{\lambda_{min} + \lambda_{max}}{2} \tag{2.17}$$

If it leads to insufficient generation, the minimum lambda value is updated to $\lambda_{min} = \lambda$. If it leads to excess generation, the maximum lambda value is updated to $\lambda_{max} = \lambda$. The new search interval (λ_{min}, λ_{max}), is then bisected until the lambda value that leads to the satisfaction of the power balance constraint is determined.

2.3.1 ALGORITHM OF LAMBDA ITERATION METHOD FOR SOLVING ELD:

1. Read data, namely cost coefficients, a_i, b_i, and c_i, (i=1,2, …, NG; j=1,2, …,NG), B-coefficients convergence tolerance, ε, step size α, and maximum iterations allowed, IT_{max}, etc.
2. Compute the initial values of P_i (i=1, 2……., NG) and λ by assuming that the transmission losses are zero, i.e. P_L=0. Then the problem can be stated by equations

$$F(P_i) = \sum_{i=1}^{NG} F_i(P_i), \qquad \text{and} \qquad \sum_{i=1}^{NG} P_i = P_D$$

and the solution can be obtained directly using equations $P_{gi}=(\lambda-b_i)/2a_i$, (i=1, 2, NG), and

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

3. Assume no generator has been fixed at either lower limit or upper limit.
4. Set iteration counter, IT = 1.
5. Compute P_{gi} (i=1,2, …, NG) of generators which are not fixed at either upper or lower limits.
6. Compute transmission losses using equation 2.6
7. Compute $\Delta P = P_D + P_L - \sum P_{gi}$
8. Check $|\Delta P| \leq \varepsilon$, if yes then GOTO step 11. Check $IT \geq IT_{max}$, if yes then GOTO step 11.
9. Modify $\lambda^{new} = \lambda + \alpha\, \Delta P$, where α is the step size used to increase or decrease the value of λ in order to meet step 7.
10. IT = IT +1, $\lambda = \lambda^{new}$ and GOTO step 5 and repeat.
11. Check the limits of generators, if no more violation then GOTO step 13, else fix as following. If $P_{gi} < P_{gi}^{min}$ then $P_{gi} = P_{gi}^{min}$
    If $P_{gi} > P_{gi}^{max}$, then $P_{gi} = P_{gi}^{max}$
12. GOTO step 4.
13. Compute the optimal total cost and losses from equation 2.1 and 2.6
14. STOP

2.3.2 FLOW CHART OF LIM

The flow chart of LIM for solving the ELD problem is shown in fig 2.3

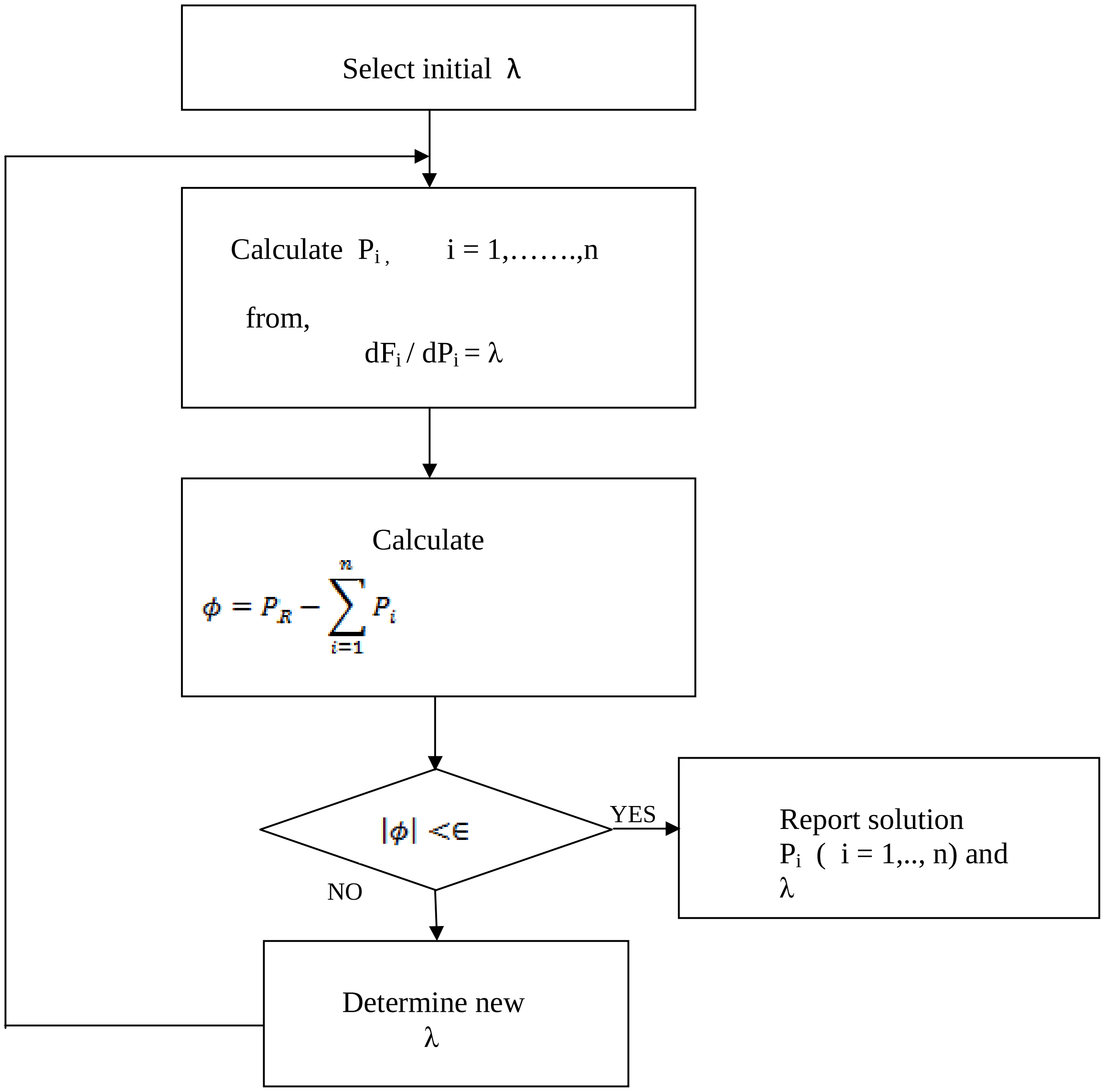

Fig 2.3 Lambda Iteration flow chart to solve ELD

CHAPTER 3

GENETIC ALGORITHM AND FUZZY LOGIC

3.1 GENETIC ALGORITHM

Genetic Algorithm (GA) was first introduced by john Holland of Michigan university in 1970's.The GA is a stochastic global search method that mimics the metaphor of natural biological evolution such as selection, crossover, and mutation **[24-25]**. The GA's combines an artificial principal with genetic operation. The artificial principal is the Darwinians survival of fittest principal and the genetic operation is abstracted from nature to form a robust mechanism that is very effective at finding optimal solutions to complex real world problems.

GA's operate on string structures. The string is binary digits representing a coding of control parameters for a given problem. The each parameter of the given problem is coded with strings of bits. The individual bit is called 'gene' and the content of the each gene is called 'allele'. The total strings of such genes of all parameters written in a sequence is called a 'chromosome' so there exits a chromosome for each point in the search space. Here we have to know about search space.

In this approach, a GA candidate solution is represented as a linear string analogous to a biological chromosome. The general scheme of GAs starts from a population of randomly generated candidate solutions (chromosomes). Each chromosome is then evaluated and given a value which corresponds to a fitness level in objective function space. In each generation, chromosomes are chosen based on their fitness to reproduce offspring. Chromosomes with a high level of fitness are more likely to be retained while the ones with low fitness tend to be discarded. This process is called selection. After selection, offspring chromosomes are constructed from parent chromosomes using operators that resemble crossover and mutation mechanisms in evolutionary biology. The crossover operator, sometimes called recombination, produces new offspring chromosomes that inherit information from both sides of parents by combining partial sets of elements from them. The mutation operator randomly hanges elements of a chromosome with a low probability. Over multiple generations, chromosomes with higher fitness values are left based on the survival of the fittest **[26]**.

Search space: If we are solving some problem, we work towards the some solution which is the best among the others. The space for all possible feasible solution is called search space.

A set of search points selected and used for processing is called population i.e. population is a set of chromosomes. The number of chromosome in a population is called population size and the number of gene's in each string is called string length. The population is processed and evaluated through various operators of GA to generate a new population and this process is carried out till global optimum points are reached.

3.2 GA'S VERSUS TRADITIONAL METHODS:

From the above discussion, it can be seen that the GA differs substantially from more traditional search and optimization methods. The four most significant differences are:

1. GA's search a population of points in parallel, not a single point.
2. GA's do not require derivative information or other auxiliary knowledge.
3. GA's are probabilistic transition rules, not deterministic ones.
4. GA's work on an encoding of the parameter set rather than the parameter set itself except in where real-valued individuals are used.

3.3 OBEJECTIVE FUCTION AND FITNESS FUCTION:

The objective function is used to provide a measure of how individuals have performed in the problem domain. In the case of a minimization problem, the most fit individual will have the lowest numerical value of the associated objective function. This raw measure of fitness is usually only used an intermediate stage in determining the relative performance of individuals in a GA.

Another function, the fitness function is normally used to transform the objective function value into a measure of relative fitness, thus

$$F(x) = g(\ f(x)\ ) \qquad (3.1)$$

Where ‘f’ is the objective function, ‘g’ transform the value of the objective function to a non-negative number and ‘F’ is resulting relative fitness. In many cases, the number of offspring’s that an individual can except to produce in the next generation.

This simple genetic procedure constantly produces even fitter offspring through successive generations. This process gradually leads the search towards a global optimum solution. A flowchart for a GA is shown in fig.3.1

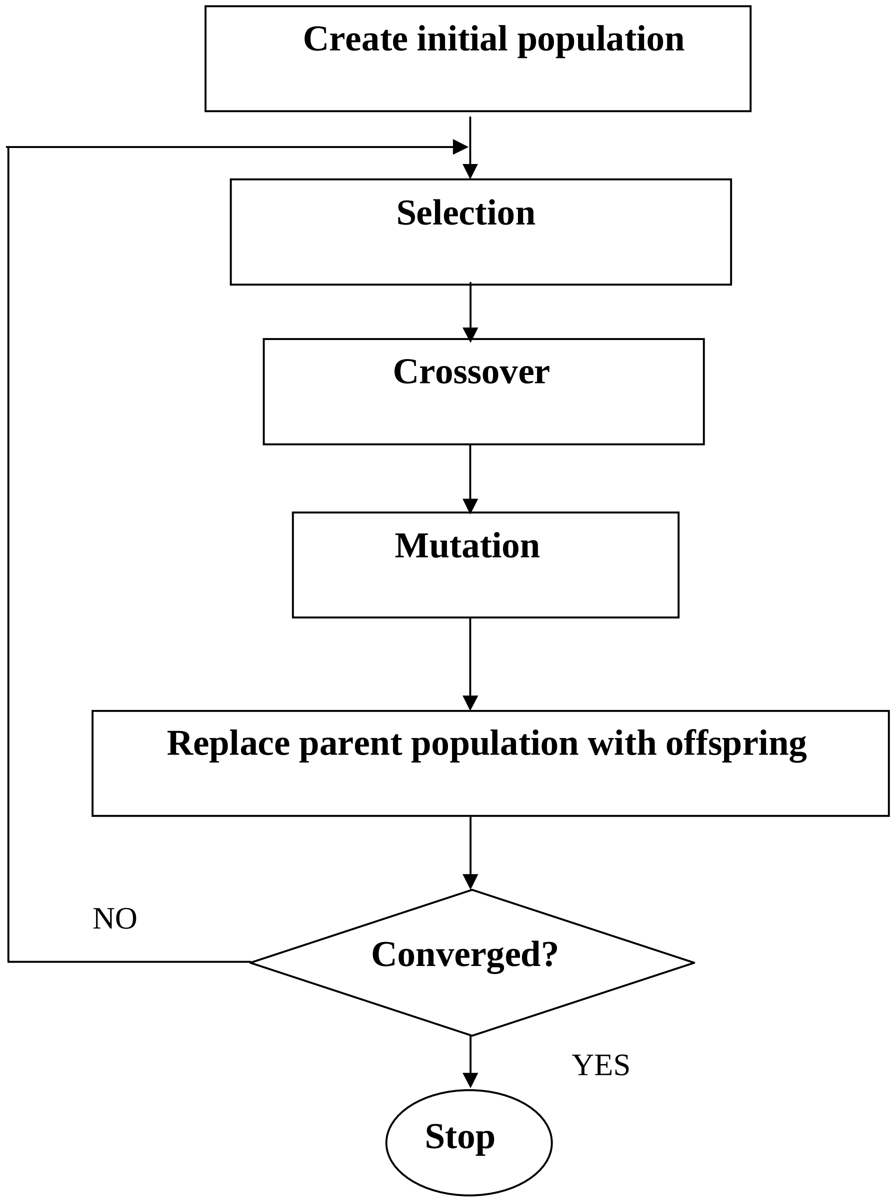

Fig 3.1 Typical flowchart of genetic algorithm

It involves nothing more than swapping of genes and string cloning. This allows GA to produce good results in circumstances which are hard to achieve through many conventional methods. The further attraction to such an algorithm is that it is extremely robust with respect to the complexity of the problem.

3.4 PHASES IN GENETIC ALGORITHM

Typically, the genetic algorithms have three phases

1. Initialization
2. Evaluation
3. Genetic operation

3.4.1 INITIALIZATION

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 **[26].** The number of chromosomes in a population is usually selected to be between 30 and 100 **[11].**

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.

3.4.2 EVALUTAION:

In this phase, we determine the suitability of the solutions 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 fitness function for the maximization problem is

$$f(x) = F(x) \quad (3.2)$$

and for the minimization problem

$$f(x) = 1/(1+F(x)) \tag{3.3}$$

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

3.4.3 GENETIC OPERATION

In this phase we generate a new population from the previous population using genetic operators. They are

- Reproduction
- Crossover
- Mutation

3.4.3.1 REPRODUCTION

This is the operator used to copy the old chromosome into matting pool according to its fittest valve. Higher the fitness of the chromosome more is number of the copies in the next generation chromosome. Chromosomes are selected from the population to be parents to crossover and produce offspring. According to Darwin's fittest principle the best one should survive and create new offspring. That's why this operator is called selection operator. The commonly used reproduction operator is proportionate reproduction operator. The i^{th} string in the population is selected with a probability proportional *fi* where f_j is the fitness value for that string. The probability of selecting i^{th} string is

$$P_i = \frac{f_i}{\sum_{j=1}^{n} f_j} \tag{3.4}$$

where 'n' is the population size.

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.

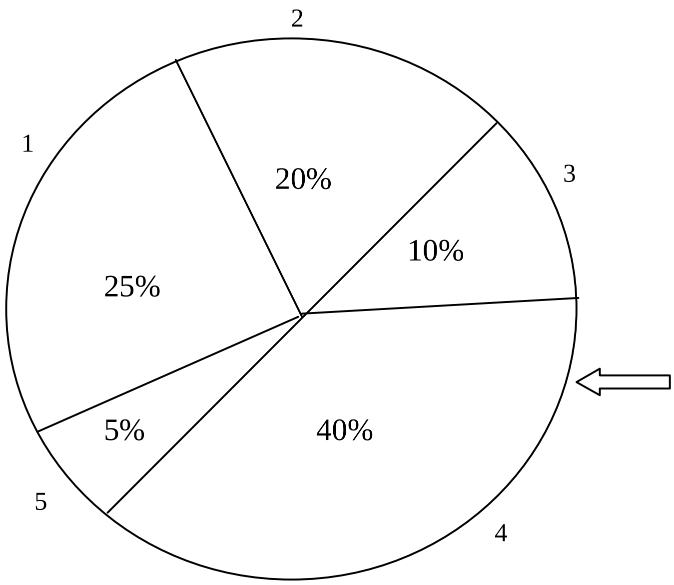

Fig.3.2 A Roulette-wheel is marked for five individuals according to fitness value

The roulette-wheel mechanism is expected to make f_i / fit_{avg} copies of i^{th} string of the mating pool. The average fitness is

$$fit_{avg} = \sum_{j=1}^{n} \frac{f_j}{n} \qquad (3.5)$$

In the figure 3.2 a roulette wheel selection terminology has been given. Since the third individual has a higher fitness value than any other, it is expected that the roulette wheel selection will choose the third individuals more than any other individuals.

This selection method is less noisy and is known as stochastic remainder selection **[32]**. 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.

3.4.3.2 CROSSOVER

The basic operator for producing new chromosome is crossover. In this operator, information is exchanged among strings of matting pool to create new strings. The aim of the crossover operator is to search the parameter space. Crossover is a recombination operator, which proceeds in three steps. First, the reproduction operator selects at random a pair of two individual string for mating, then a crossover site is selected at random along the string length and the position values are swapped between two string following the cross site.

1. Single point crossover
2. Two point crossover
3. Multi point crossover
4. Uniform crossover
5. Matrix 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** }

Fig 3.3 Single point crossover operation

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 fig 3.4. In a multipoint crossover, again there are two cases. One is even no of cross sites and other is odd no of sites. For even no of sites the string is treated as a ring and cross

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}

Fig 3.4 Two point crossover operation

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

3.4.3.3 MUTATION

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

Here, binary mutation flips the value of the bit at the loci selected to be the mutation point. Given that mutation is applied uniformly to an entire population of strings, it is possible that a given string may be mutated at more than one point.

Offspring x1 : 1 1 **1** 1 0 1 0

New offspring x2 : 1 1 **0** 1 0 1 0

Fig 3.5 Mutation operation

3.5 ALGORITHM FOR ELD USING GA

The step-wise procedure is outlined below:

1. Read data, namely cost coefficients, a_i, b_i, c_i, B-coefficients, B_{ij}(i=1,2,….NG; and j=1,2,….NG), convergence tolerance, error, step size and max allowed iterations, length of string, L, population size, p_c probability of cross over, p_m probability of mutations, lambda min, lambda max.
2. Generate an array of random numbers. Generate the population λ_j (j=1,2,…, L) by flipping coin. The bit is set according to the coin flip as

   $$b_{ij} = \begin{cases} 1, & \text{if } p = 1 \text{ or random } 0 \leq p \\ 0, & \text{otherwise} \end{cases}$$

   Where, p is the probability (0.5)
3. Set generation counter, k=0, BIG (minimum error)=0.1, f_{max}=0.0, and f_{min}=1.0
4. Increment the generation counter k=k+1 and set population counter J=0
5. Increment population counter j=j+1
6. Decode the string using Eq. (3.6)
7. Using gauss elimination method, find P_i
8. Calculate transmission loss using Eq. (2.6)
9. Find e_j and check if (e_j<BIG), then set BIG= e_j
10. Find fitness if ($f_j > f_{max}$) then set $f_{max} = f_j$ and if ($f_j < f_{min}$) then set $f_{min} = f_j$
11. If (j<L) then GOTO step 5 and repeat
12. If (BIG ≤ error) then GOTO step 18
13. Find population with max fitness and average fitness of the population
14. Select the parents for crossover using stochastic remainder roulette wheel selection method.
15. Perform single point crossover for the selected parents
16. Perform mutation
17. If (k<IT_{max}) then GOTO step 4 and repeat
18. STOP

3.6 METHODOLOGY

Here solution methodology which includes the encoding and decoding, constrained generation output are explained.

3.6.1 ENCODING & DECODING

Decoding a binary string into unsigned integer can play very important role in GA implementation. The inequality power limit constraint is performed in such a way that the individual string is normalized over the unit's operating region. The inequality constraints are handled in the manner, which efficiently reduces the searching space, and thus enhances the performance of the system. Binary coded strings having 1's and 0's are used. The equivalent decimal integer of binary string λ is obtained as

$$y_i = \sum_{i=1}^{L} 2^{i-1} b_{ij} \qquad (3.6)$$

Where ,

b_{ij} is the ith binary digit of the jth string and

L is the number of strings or population size

3.6.2 CALCULATION OF GENERATION & TRANSMISSION LOSSES

When the incremental cost λ^j is known for whole population, then the generation can be obtained as

$$2\left(a_i + \lambda^j B_{ii}\right) P_j^i + \lambda^j \sum_{\substack{k=1 \\ k \neq i}}^{NG} 2 B_{ik} P_k^i = (\lambda^j - b_i) \quad , (i = 1, 2, \ldots NG;\ j = 1, 2, \ldots\ldots, L) \qquad (3.8)$$

The above equation can be rewritten as

$$\sum_{k=1}^{NG} A_{ik}^j P_k^j = C_i^j \quad , (i = 1, 2, \ldots., NG; \quad j = 1, 2, \ldots\ldots L) \qquad (3.9)$$

Where,

$$A_{ii}^j = 2\left(a_i + \lambda^j B_{ii}\right) \qquad (3.10)$$

$$A_{ik}^j = 2\lambda^j B_{ik} \quad (i \neq k) \qquad (3.11)$$

$$C_i^j = \left(\lambda^j - b_i\right) \qquad (3.12)$$

Transmission loss for whole population can be obtained as

$$P_L^j = \sum_{i=1}^{NG} \sum_{k=1}^{NG} P_i^j B_{ik} P_k^j \qquad ,(j = 1, 2, \ldots.., L) \qquad (3.13)$$

3.7 FUZZY LOGIC

Logic is the art and science of reasoning which seeks to identify and understand the principles of demonstration and inference. Logic is a branch of philosophy and was part of the classical trivium. As a discipline, logic dates back to Aristotle and remains integral to fields such as mathematics, computer science, and linguistics.

Fuzzy logic is a superset of conventional Boolean logic that has been extended to handle the concept of partial truth-truth-values between "completely true" and "completely false" **[28]**. Kosko Bart presents the theory of fuzzy logic as, "The facts were always fuzzy or vague or inexact... Science treated the gray or fuzzy facts as if they were the black-white facts of math. Yet no one had put forth a single fact about the world that was 100% true or 100% false" **[27]**.

Typically, engineering requires the use of exact or mathematical statements. These statements correspond with precise information, such as "x = 3.0," "$2 \leq u \leq 6$", or "y = 3t + 24." The value of "x = 3" has a grade membership comparable with 100% (= 1); for all other values (2.8, 2.9, 3.1, 3.2), the grades of membership in the solution is zero. In the case of real-world values, however, this grade of membership is not true because of the imprecision of tools, the influence of the observer, and so forth.

Lotfi A. Zadeh **[29]** proposed such a theory to make a rapprochement between the precision of classic mathematics and the imprecise information from the real world.The theory is called fuzzy sets theory and works with grades of membership of x in A, that is, $\mu_A(x)$, taken from the set M, which must be a lattice structure **[30].** Usually, this set M is taken in the interval [0, 1]. Goguem **[31]** proposed a further generalization of the theory using values of the membership taken from the set L ($-\infty$, ∞) or ([-1, 1] for a normalized set). Fuzzy logic is the way the human brain works, and we can mimic this in machines so they will perform somewhat like humans (not to be confused with Artificial Intelligence, where the goal is for machines to perform exactly like humans). Fuzzy logic control and analysis systems may be electro-mechanical in nature, or concerned only with data, for example economic data, in all cases guided by "If-Then rules" stated in human language.

3.7.1 FUZZY SET

Fuzzy sets have membership properties defined between 0 and 1. This means that if we take an attribute say 'red' we can express the colour of any particular apple as a position in this fuzzy set. We may say for example that it is 30% red and thus has a fuzzy truth value (FTV) or membership function of 0.3. The relation of FTV to actual values depends upon the desired mapping from the real world to the normalized range 0 to 1, and this is arbitrary.

Fuzzy sets support a flexible sense of membership of elements to a set. While in crisp set theory, an element either belongs to or does not belong to a set, in fuzzy set theory many degrees of membership (between 0 and 1) are allowed. Thus a membership function $\mu_{\tilde{A}}^{(x)}$ is associated with a fuzzy set Ã such that the function maps every element of the universe of discourse X (or the reference set) to the interval [0, 1]. Formally the mapping is written as $\mu_{\tilde{A}}^{(x)} : X \rightarrow [0, 1]$. A fuzzy set is defined as fallows:

If X is a universe of discourse and x is a particular element of X, then a fuzzy set a defined on X may be written as a collection of ordered pairs

$$A= \{(x, \mu_{\tilde{A}}^{(x)}), x \in X \}$$

Where each pair $(x, \mu_{\tilde{A}}^{(x)})$ is called a single tone. In crisp sets, $\mu_{\tilde{A}}^{(x)}$ is dropped **[32]**. Fuzzy logic is reasoning with fuzzy sets. Operations on fuzzy sets are similar to those of standard logic but are differently defined. Let us assume two FTVs to illustrate, A(0.4) and B(0.7) **[33].** Fuzzy Set theory involves the following Operations

i. Union

ii. Intersection

iii. Complement

iv. De Morgan’s Law

v. Associativity

vi. Commutativity

vii. Distributivity

Union (the joined boundarys of the values):

A OR B = Maximum of the fuzzy truth values (FTVs) i.e., 0.7

Intersection (the commonality between the values):

A AND B = Minimum of the FTVs i.e., 0.4

(again reducing to bivalent logic in the extremes)

Negation (the opposite of the value)

NOT A = 1 – FTV A i.e., 0.6

(Once more this is simply an extension of normal logic)

3.7.2 MEMBERSHIP FUNCTION

The membership function values need not always be described by discrete values. Quite often, these turn out to be as described by a continuous function. The membership function is a graphical representation of the magnitude of participation of each input. It associates a weighting with each of the inputs that are processed, define functional overlap between inputs, and ultimately determines an output response **[33]**.

The fuzzy membership function **[32]** for the fuzzy linguistic term “cool” relating to temperature may turn out to be as illustrated in fig 3.6

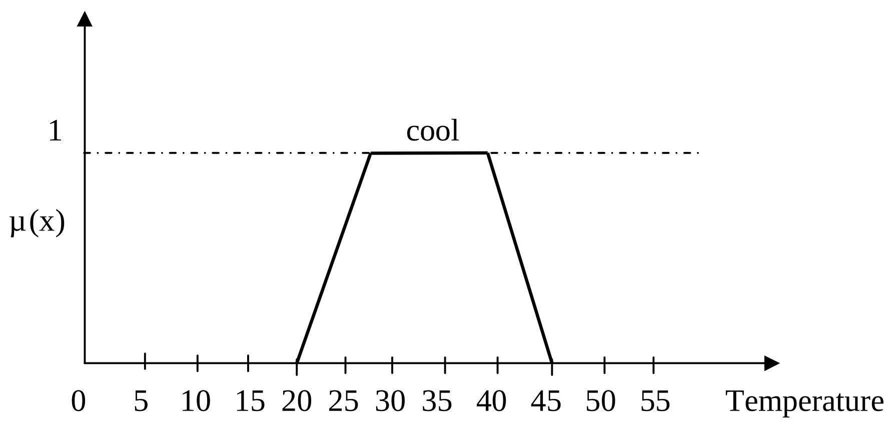

Fig 3.6 Continuous membership functions for “cool”.

A membership function can also be given mathmetically as

$$\mu_{\tilde{A}}(x) = 1/(1+x)^2 \tag{3.14}$$

the graph is as shown in fig 3.7

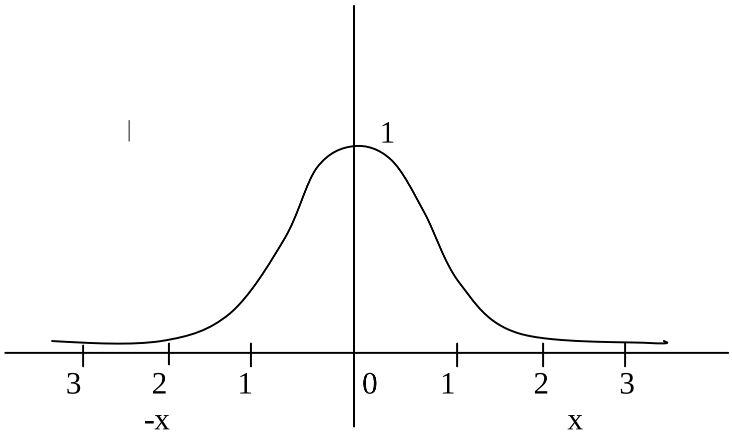

Fig 3.7 Continuous membership function dictated by a mathmetical function

Different shapes of membership functions exit. The commonly used shape to describe the membership function is triangular, but bell, trapezoidal and exponential can also be used as shown in figure 3.8 with their variation

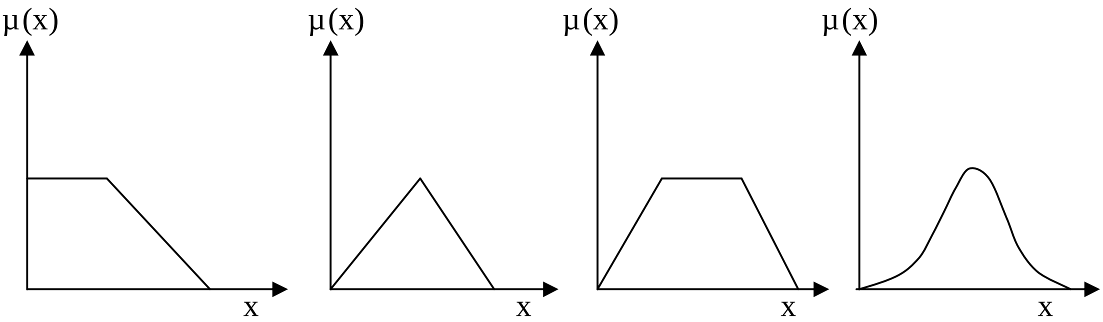

Fig 3.8 Different shapes of membership function graphs

Human beings make decisions based on rules. Although, we may not be aware of it, but whatever decisions are made are all based on computer like if-then statements. For example, if the weather is fine, then we may decide to go out. If the forecast says the weather will be bad today, but fine tomorrow, then we make a decision not to go today, and postpone it till tomorrow. Rules associate ideas and relate one event to another **[34]**.

Fuzzy machines, which always tend to mimic the behaviour of man, also work in the same way. However, the decision and the means of choosing that decision are replaced by fuzzy sets and the rules are replaced by fuzzy rules. Fuzzy rules also operate using a series of if-then statements. For instance, if X is positive then A, and if Y is negative then B, where A and B are all sets of X and Y. Fuzzy rules define fuzzy patches, which is the key idea in fuzzy logic.

3.7.3 CRISP (CLASSICAL) VERSUS FUZZY SET

Crisp set requires a deep understanding of a system, exact equation, and precise numeric value, Fuzzy logic represents an alternative way of thinking, which allows modeling complex system using a higher level of abstraction originating from our knowledge and experience. Fuzzy Logic allows expressing this knowledge with subjective concepts such as very hot, bright red, and a long time which are mapped into exact numeric ranges.

Consider the query “Is water colourless?” The answer to this is a definite yes/true or no/false as warranted by the situation. If “yes/true” is accorded a value of 1 and “no/false” is accorded value of 0, this statement results in a 0/1 type situation. Such a logic which demands a binary (0/1) type of handling is termed crisp in the domain of fuzzy set theory. Thus statement such as “temperature is 32^0 C”, “the running time of program is 4 seconds” are examples of crisp situations.

On the other hand consider the statement, “is Ram honest?” The answer to this query need not to be definite “yes” or “no”. considering the degree to which one know Ram, a variety of answers spanning a range such as “extremely honest”, “extremely dishonest”, “honest at times”, “very honest” could be generated. If for instance, “extremely honest” were to be accorded a value of 1, at the high end of spectrum of value “extremely dishonest” a value of 0 at the low end of the spectrum then “honest at the times” and “very honest” could be assigned value of 0.4 and 0.85 respectively. So the situation is that it can accept values between 0 and 1. Such a situation is termed fuzzy **[32]**. The differenses between crisp and fuzzy may be easily understand by figure 3.9

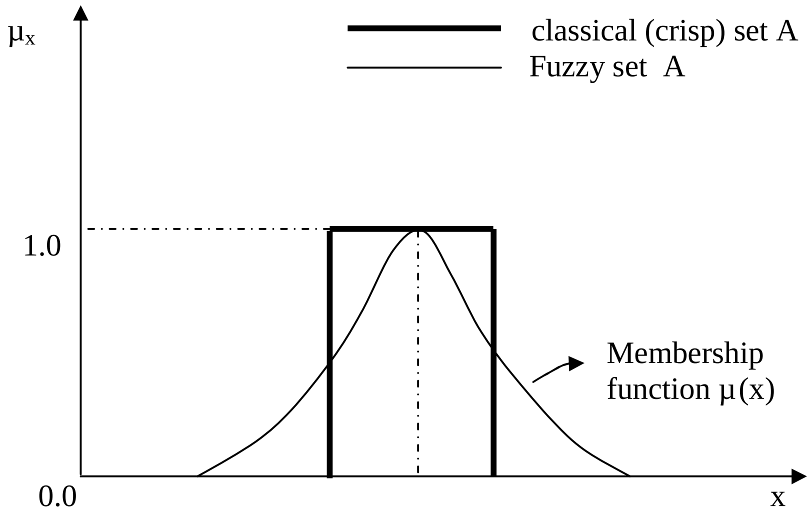

Fig 3.9 Shapes of crisp and fuzzy

3.8 FEATURES OF FUZZY LOGIC

Fuzzy Logic offers several unique features that make it a particularly good choice for many control problems.

1. It is inherently robust since it does not require precise, noise-free inputs and can be programmed to fail safely if a feedback sensor quits or is destroyed. The output control is a smooth control function despite a wide range of input variations.

2. Since the fuzzy logic controller processes user-defined rules governing the target control system, it can be modified easily to improve or drastically alter system performance. New sensors can easily be incorporated into the system simply by generating appropriate governing rules.

3. Fuzzy logic is not limited to a few feedback inputs and one or two control outputs, nor is itnecessary to measure or compute rate-of-change parameters in order for it to be 9 implemented. Any sensor data that provides some indication of a system's actions and reactions is sufficient. This allows the sensors to be inexpensive and imprecise thus keeping the overall system cost and complexity low.

4. Because of the rule-based operation, any reasonable number of inputs can be processed (1-8 or more) and numerous outputs (1-4 or more) generated, although defining the rule base quickly becomes complex if too many inputs and outputs are chosen for a single implementation, since rules defining their interrelations must also be defined. It would be better to break the control system into smaller chunks and use several smaller fuzzy logic controllers distributed on the system, each with more limited responsibilities.

5. Fuzzy logic can control nonlinear systems that would be difficult or impossible to model mathematically. This opens doors for control systems that would normally be deemed unfeasible for automation.

3.9 FUZZY EXPERT SYSTEM

A fuzzy expert system is an expert system that uses a collection of fuzzy membership functions and rules, instead of Boolean logic, to reason about data. The rules in a fuzzy expert system are usually of a form similar to the following:

if x is low and y is high then z is medium

where x and y are input variables (names for know data values), z is an output variable (a name for a data value to be computed), low is a membership function (fuzzy subset) defined on x, high is a membership function defined on y, and medium is a membership function defined on z. The antecedent (the rule's premise) describes to what degree the rule applies, while the conclusion (the rule's consequent) assigns a membership function to each of one or more output variables. Most tools for working with fuzzy expert systems allow more than one conclusion per rule. The set of rules in a fuzzy expert system is known as the rule base or knowledge base. The general inference process proceeds in three (or four) steps.

1. Under FUZZIFICATION, the membership functions defined on the input variables are applied to their actual values, to determine the degree of truth for each rule premise.

2. Under INFERENCE, the truth value for the premise of each rule is computed, and applied to the conclusion part of each rule. This results in one fuzzy subset to be assigned to each output variable for each rule. Usually only MIN or PRODUCT are used as inference rules. In MIN inferencing, the output membership function is clipped off at a height corresponding to the rule premise's computed degree of truth (fuzzy logic AND). In PRODUCT inferencing, the output membership function is scaled by the rule premise's computed degree of truth.

3. Under COMPOSITION, all of the fuzzy subsets assigned to each output variable are combined together to form a single fuzzy subset for each output variable. Again, usually MAX or SUM are used. In MAX composition, the combined output fuzzy subset is constructed by taking the point wise maximum over all of the fuzzy subsets assigned to variable by the inference rule (fuzzy logic OR). In SUM composition, the combined output fuzzy subset is constructed by taking the point wise sum over all of the fuzzy subsets assigned to the output variable by the inference rule.

4. Finally is the (optional) DEFUZZIFICATION, which is used, when it is useful to convert the fuzzy output set to a crisp number. There are more defuzzification methods in which two of the more common techniques are the CENTROID and MAXIMUM methods. In the CENTROID method, the crisp value of the output variable is computed by finding the variable value of the center of gravity of the membership function for the fuzzy value. In the MAXIMUM method, one of the variable values at which the fuzzy subset has its maximum truth value is chosen as the crisp value for the output variable.

3.10 FUZZY CONTROL

Fuzzy control, which directly uses fuzzy rules, is the most important application in fuzzy theory. Using a procedure originated by Ebrahim Mamdani in the late 70s, three steps are taken to create a fuzzy controlled machine [34, 35]:

1) Fuzzification (Using membership functions to graphically describe a situation)

2) Rule evaluation (Application of fuzzy rules)

3) Defuzzification (Obtaining the crisp or actual results)

Advantages

i. Allows the use of vague linguistic terms in the rules.

ii. Fuzzy logic solutions are easy to verify and optimize.

Disadvantages

i. It is difficult to optimize membership function.

ii. There are many ways of interpreting fuzzy rules, combining the output of several fuzzy rules and defuzzifying the outputs.

CHAPTER 4

ECONOMIC LOAD DISPATCH USING FCGA

In this chapter fuzzy logic controlled genetic algorithm (FCGA) is briefly reviewed. This formulation is applied to ELD, by varying the crossover probability and mutation probability during the GA method.

4.1 FUZZY LOGIC CONTROLLED GENETIC ALGORITHM

This is mainly the hybrid system. Hybrid systems are those which employ integrated technologies to effectively solve problems. Hybrid systems are classified as sequential, auxiliary and embedded hybrids.

Fuzzy-Genetic Hybrid system applicable on fuzzy optimization problems. The system obtains optimal solution to problems with fuzzy constraints and fuzzy variables. The hybrid system has been demonstrated on the problems of optimization of structures (civil/machine tool) and obtain the optimal mix for high performance concrete. The optimal solution is obtained by Genetic Algorithm which crossover and mutation adjusted by fuzzy practical rules.

4.2 FUZZY RULES FOR CROSSOVER AND MUTATION ADJUSTMENT

For better results and to get faster convergence, conventional GA modes have been modified. In recent years various techniques have been studied to achieve this objective, these include **[36]**

- Using advanced string coding.
- Generating initial population with some prior knowledge.
- Establishing some better evaluation function.
- Including new operators such as elitism, multi point or uniform crossover and creep mutation.

A refined GA was used to solve the economic dispatch in **[38]** and a Pyramid Genetic Algorithm (PGA) has been used in **[39]** for voltage profile optimization.

FCGA proposes a flexible Genetic Algorithm which based on fuzzy logic rules with the ability to adjust continuously the crossover and mutation parameters. Figure 4.1 presents the proposed block diagram of a fuzzy logic controlled genetic algorithm.

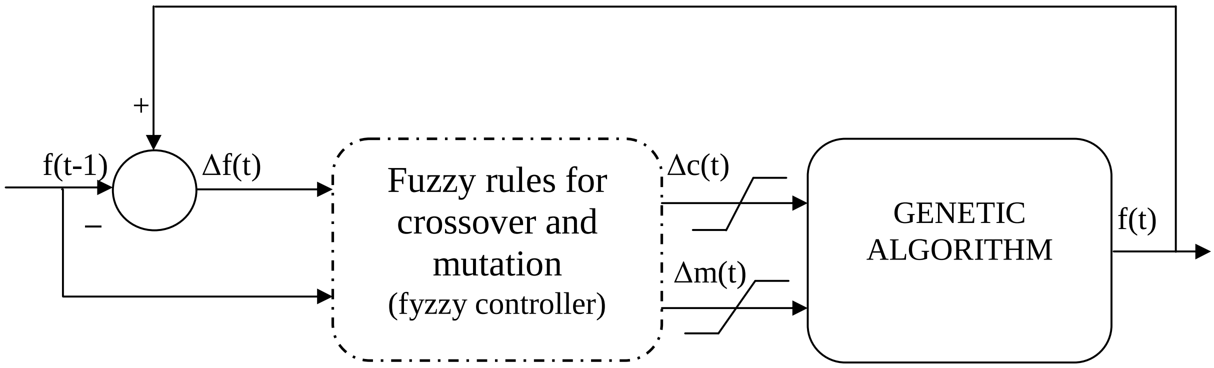

Fig 4.1 Global block diagram of the genetic parameters adjustment.

Crossover and Mutation are considered critical for GA convergence. A suitable value for mutation provides balance between global and local exploration abilities and consequently results in a reduction of the number of iterations required to locate the optimum solution. Experimental results based in application of GA to any practical networks at normal and abnormal conditions with load incrementation indicated, that it is better to adjust dynamically the value of the two parameters, crossover and mutation. It is intuitive that for a small variation in the chromosomes in a particular population, the effect of crossover during this critical stage becomes insignificant therefore, creating diversity in the population is required by increasing mutation (High value) probability of the chromosome and reducing (Low value) the value of crossover, note that the terms, small and high are linguistic. The proposed approach employs practical rules interpreted in fuzzy logic rules to adjust dynamically the two parameters (crossover and mutation) during execution of the GA standard algorithm **[57]**.

4.3 MEMBERSHIP FUNCTION DESIGN

The variables chosen for this controller are change in average fitness (Δf(t)), change in average fitness in last iteration (Δf(t-1)), change in crossover probability (Δc(t)) and change in mutation probability ($\Delta m(t)$). In this, $\Delta f(t)$ and $\Delta f(t-1)$ are the input variables and $\Delta c(t)$ is the crossover output and $\Delta m(t)$ is the mutation output variable. The number of linguistic variables describing the fuzzy subsets of a variable varies according to the application. Usually an odd number is used. However, increasing the number of fuzzy subsets results in a corresponding increase in the number of rules. Each linguistic variable has its fuzzy membership function. The membership function maps the crisp values into fuzzy variables. The triangular membership functions are used to define the degree of membership. It is important to note that the degree of membership plays an important role in designing a fuzzy controller. Each of the input and output fuzzy variables is assigned nine linguistic fuzzy subsets varying from negative larger (NL) to positive larger (PL). Each subset is associated with a triangular membership function to form a set of nine membership functions for each fuzzy variable given in table 4.1

Table 4.1 Membership functions for fuzzy variables

NL	NEGATIVE LARGER
NR	NEGATIVE LARGE
NM	NEGATIVE MEDIUM
NS	NEGATIVE SMALL
ZE	ZERO
PS	POSITIVE SMALL
PM	POSITIVE MEDIUM
PR	POSITIVE LARGE
PL	POSITIVE LARGER

The inputs to the crossover fuzzy logic controller are changes in fitness at two consecutive steps, i.e. $\Delta f(t - l)$, $\Delta f(t)$, and the output of which is change in crossover $\Delta c(t)$. Membership functions of fuzzy input and output linguistic variables are shown in Fig 4.2. $\Delta f(t-1)$, $\Delta f(t)$ are normalised into the range of [-1.0, 1.0], and $\Delta c(t)$ is normalised into the range of [-0.1, 0.1] according to their corresponding maximum values.

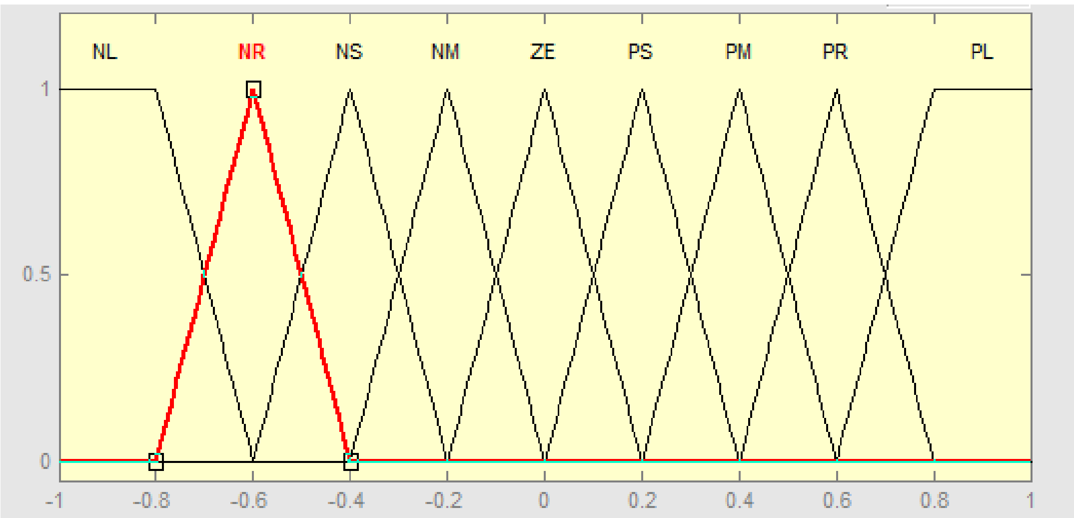

Fig. 4.2(a) Membership functions for Δf(t)

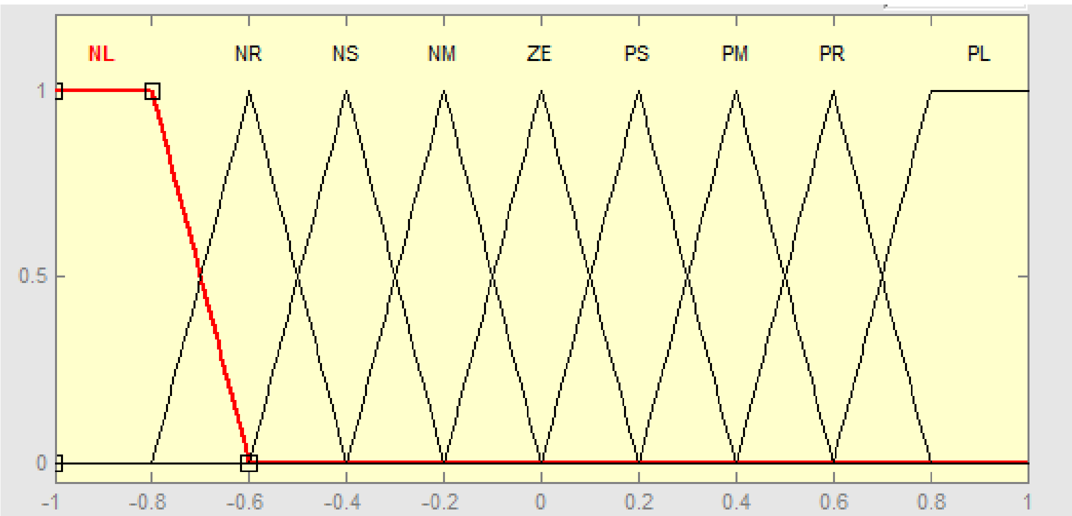

Fig. 4.2(b) Membership functions for Δf(t-1)

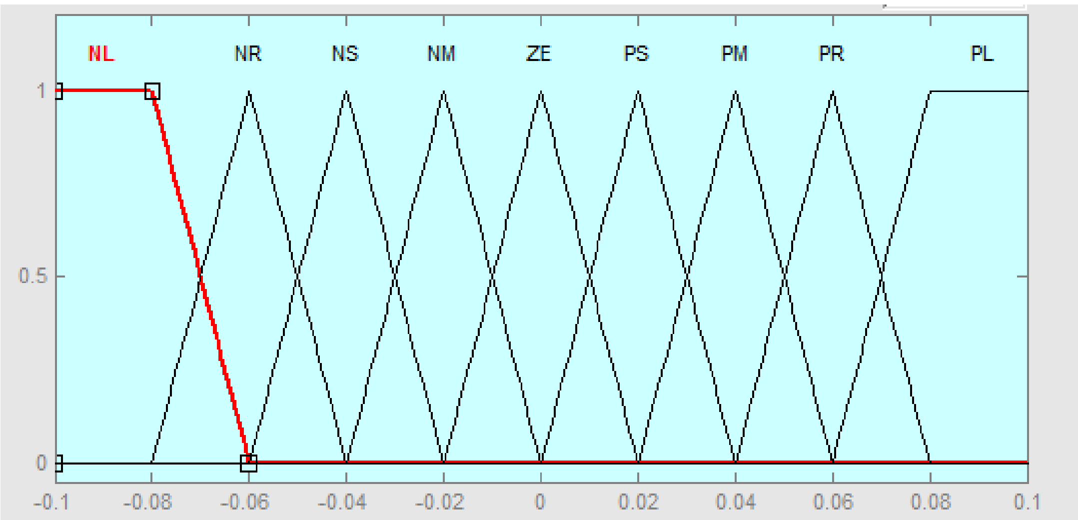

Fig. 4.2(c) Membership functions for $\Delta c(t)$

The mutation operation is determined by the flip function with mutation probability rate, and the mutate bit is randomly performed. The mutation probability rate is automatically modified during the optimization process based on the fuzzy logic controller. The heuristic information for adjusting the mutation probability rate is if the change in average fitness is very small in consecutive generations, then the mutation probability rate should be increased until the average fitness begins to increase in consecutive generations.

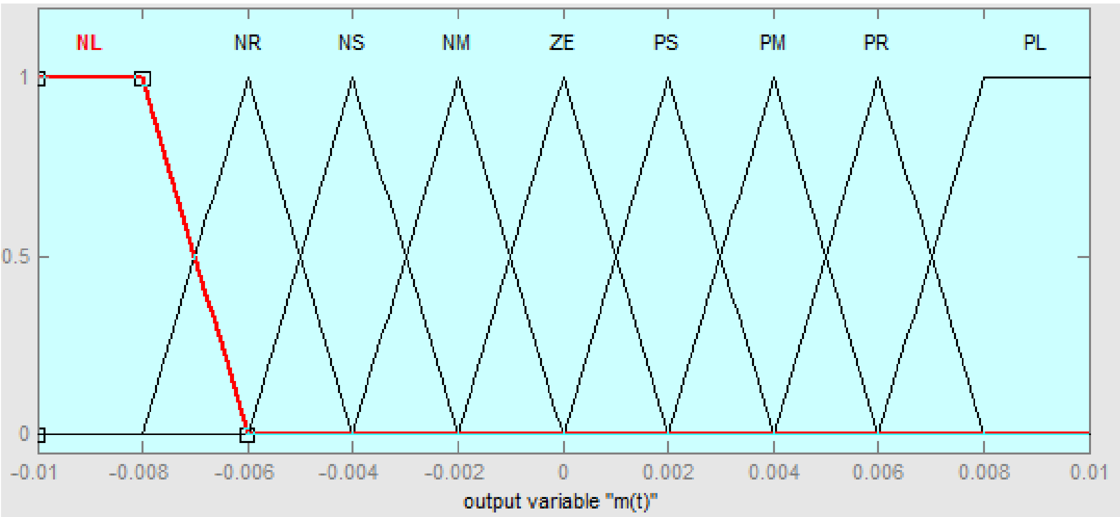

Fig. 4.3 Membership functions for $\Delta m(t)$

If the average fitness decreases the mutation probability rate should be decreased. Only the change in output of mutation operator membership function which is shown in fig 4.3. The inputs to the mutation fuzzy controller are the same as those of the crossover fuzzy controller, and the output of which is the change in mutation $\Delta m(t)$. The design of the membership function for both input and rule base table for the fuzzy mutation controller is similar to these for the fuzzy crossover controller.

4.4 FUZZY RULE BASE FOR SOLVING THE ELD

A set of rules which define the relation between the input and output of fuzzy controller can be found using the available knowledge in the area of designing ELD. These rules are defined using the linguistic variables. The two inputs, $\Delta f(t)$ and $\Delta f(t-1)$, result in 81 rules. The typical rules are having the following structure:

Rule 1**:** If $\Delta f(t)$ is NL (negative larger) AND $\Delta f(t-1)$ is NM (negative medium) then $\Delta c(t)$ (output of fuzzy crossover) is NR (negative large).
Rule 2**:** If $\Delta f(t)$ is PM (positive medium) AND $\Delta f(t-1)$ is NL (negative larger) then $\Delta c(t)$ (output of fuzzy crossover) is NS (negative small).
Rule 3**:** If $\Delta f(t)$ is ZE (zero) AND $\Delta f(t-1)$ is PR (positive large) then $\Delta c(t)$ (output of fuzzy crossover) is PM (positive medium).. And so on….

All the 81 rules governing the mechanism are explained in Table 4.2 where all the symbols are defined in the basic fuzzy logic terminology

Table 4.2 Rule base of fuzzy logic controller

Δc (t) / Δf(t)	Δf(t-1)								
	NL	**NR**	**NM**	**NS**	**ZE**	**PS**	**PM**	**PR**	**PL**
NL	NL	NR	NR	NM	NM	NS	NS	ZE	ZE
NR	NR	NR	NM	NM	NS	NS	ZE	ZE	PS
NM	NR	NM	NM	NS	NS	ZE	ZE	PS	PS
NS	NM	NM	NS	NS	ZE	ZE	PS	PS	PM
ZE	NM	NS	NS	ZE	ZE	PS	PS	PM	PM
PS	NS	NS	ZE	ZE	PS	PS	PM	PM	PR
PM	NS	ZE	ZE	PS	PS	PM	PM	PR	PR
PR	ZE	ZE	PS	PS	PM	PM	PR	PR	PL
PL	ZE	PS	PS	PM	PM	PR	PR	PL	PL

4.5 ALGORITHM OF FCGA:

The step-wise procedure is outlined below:

1. Read data, namely cost coefficients, a_i, b_i, c_i, B-coefficients, B_{ij}(i=1,2,….NG; J=1,2,….NG), convergence tolerance, error, step size and max allowed iterations, length of string, L, population size, pc probability of cross over, pm probability of mutations, lambda min, lambda max.
2. Generate an array of random numbers. Generate the population λ_j (j=1,2,…, L) by flipping coin. The bit is set according to the coin flip as

$$b_{ij} = \begin{cases} 1 \ , \text{if } p = 1 \text{ or random } o \leq p \\ 0 \ , \text{if otherwise} \end{cases}$$, Where p is the probability (0.5)

3. Set generation counter, k=0, BIG(min error)=0.1, f_{max}=0.0, and f_{min}=1.0
4. Increment the generation counter k=k+1 and set population counter j=0
5. Increment population counter j=j+1

6. Decode the string using Eq. (3.6)
7. Using gauss elimination method, find P_i
8. Calculate transmission loss using Eq. (2.6)
9. Find e_j and check if (e_j<BIG), then set BIG= e_j
10. Find fitness if ($f_j > f_{max}$) then set $f_{max} = f_j$ and if ($f_j<f_{min}$) then set $f_{min}=f_j$
11. If (j<L) then GOTO step 5 and repeat
12. If (BIG $\leq$ error) then GOTO step 22
13. Find population with max fitness and average fitness (fit_{avg})of the population
14. Select the parents for crossover using stochastic remainder roulette wheel selection method.
15. If the change in $fit_{avg} > 0$, and keeps the same sign in consecitive generations, P_c rate should be increased, otherwise P_c rate should be decreased.
16. Generate a random number (R_n), if Rn<Pc then GOTO step 17 otherwise GOTO step 18
17. Perform single point crossover for the selected parents
18. If the change in fit_{avg} is very small then P_m rate should be increased until the fit_{avg} begins to increage. And if fit_{avg} decreases the P_m rate should be decreased.
19. Generate a random number (R_n), if $R_n<P_m$, then GOTO step 20 otherwise GOTO step 21
20. Perform mutation operation for the selected parents
21. If ($k<IT_{max}$) then GOTO step 4 and repeat
22. STOP

4.6 FLOW CHART OF FCGA

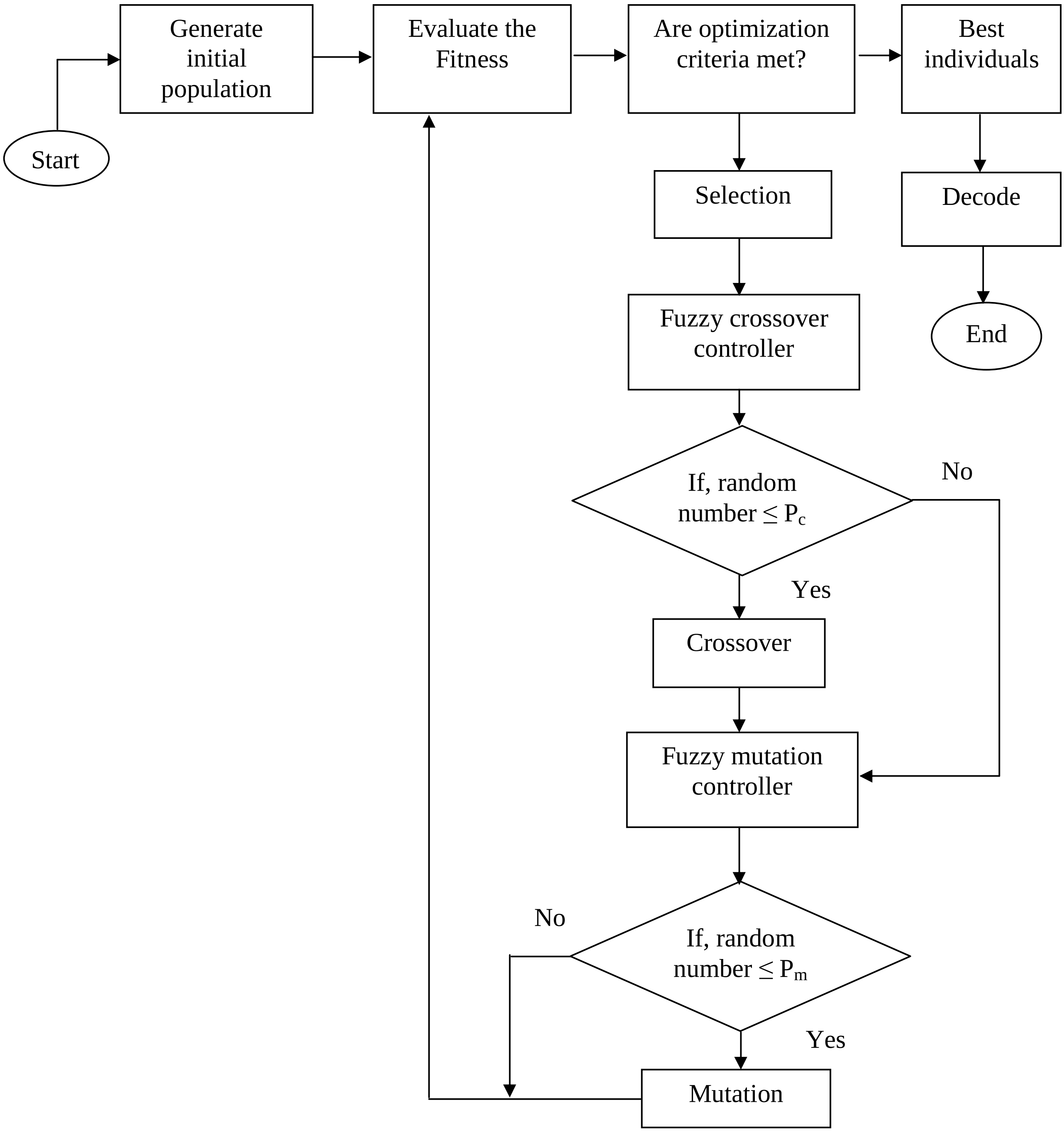

Fig 4.4 Flow chart of FCGA to solve ELD

CHAPTER 5

RESULTS AND DISCUSSIONS

The results of ELD after the implementation of proposed fuzzy logic controlled genetic algorithm are discussed and compared with the GA and classical method (lambda iteration). The algorithms are implemented in MATLAB to solve ELD problem. The main objective is to minimize the cost of generation of thermal plants using FCGA, GA and classical Lambda Iteratin Method. The performance is evaluated with losses for two set generator data, which are referred as Problem I and Problem II

Problem I: Three generator test systems **[58]**

Problem II: Ten generator test systems **[4]**

5.1 PROBLEM I: 3 GENERATOR TEST SYSTEMS

The specifications of three generator test system are detailed in table5.1. The coefficients of fuel cost are given below in Tables 5.1(a). Maximum and minimum power limits are given in Table 5.1(b). The power demand is considered to be 300MW. Transmission loss coefficients are given in Table 5.1(c). The results corresponding to Lambda Iteration method, GA and FCGA are detailed in section 5.1.1, 5.1.2 and 5.1.3 respectively.

Table 5.1 (a)

Unit no	a_i	b_i	c_i
1	0.00525	8.663	328.13
2	0.00609	10.040	136.91
3	0.00592	9.760	59.16

Table 5.1(b)

Limits	Max	Min
P_1	250	50
P_2	150	5
P_3	100	15

Table 5.1 (c)

0.000136	0.0000175	0.000184
0.0000175	0.000154	0.000283
0.000184	0.000283	0.000161

TABLE-5.1. Specifications of 3-generator test
(a) Cost coefficients
(b) Power generation limits
(c) Loss coefficients

Length of the string, l = 16

Population of string, pop = 20

Crossover probability, p_c = 0.8

Mutation probability, p_m = 0.01

The min and max value of incremental cost are assumed as

$$\lambda_{min} = 10$$

$$\lambda_{max} = 12.5$$

5.1.1 OPTIMUM SOLUTION USING LIM FOR PROBLEM-I

Developed program returns the generated power, the total cost, total losses and error. The simulated results for LIM are shown in table 5.2,

Table 5.2. Results using LIM for three generator test system

Total cost	Power			Loss	Error
	P_1	P_2	P_3		
3615.11	202.49	81.0267	27.0149	10.5311	0.000652

5.1.2 OPTIMUM SOLUTION USING GA FOR PROBLEM-I

The simulated results of ELD using GA are shown in table 5.3, the results of GA are better than the results of lambda iteration method,

Table 5.3 ELD Result using GA for three generator test system

S.no	**Fitness**	**Losses**	**Power (P_i)**			**TC**	**Error**
			P_1	**P_2**	**P_3**		
1	0.99941	10.5493	202.56	81.0595	27.1071	3617.15	0.177194
2	0.997438	10.5926	202.859	81.3118	27.1921	3624.02	0.770561
3	0.999849	10.5331	202.448	80.9649	27.0752	3614.58	0.045392
4	0.999756	10.531	202.434	80.953	27.0713	3614.26	0.0732181
5	0.999169	10.5182	202.345	80.8781	27.046	3612.22	0.249464
6	0.999185	10.5185	202.347	80.8801	27.0467	3612.27	0.244826
7	0.999827	10.5402	202.497	81.0063	27.0892	3615.71	0.0519943
8	0.999132	10.5554	202.602	81.095	27.1191	3618.12	0.260653
9	0.999009	10.5581	202.621	81.1108	27.1244	3618.55	0.297745
10	0.999802	10.532	202.441	80.959	27.0733	3614.42	0.0593049
11	0.999703	10.5429	202.515	81.022	27.0945	3614.13	0.0890918
12	0.999734	10.5422	202.511	81.0181	27.0932	3616.03	0.0798175
13	0.999827	10.5402	202.497	81.0063	27.0892	3615.71	0.0519943
14	0.99988	10.5337	202.452	80.9688	27.0766	3614.69	0.0361168
15	0.977177	11.0533	202.013	83.9701	28.0775	3696.41	7.00691
16	0.993734	106747	203.425	81.7888	27.3522	3637.01	1.89162
17	0.999564	10.5459	202.536	81.0398	27.1005	3616.62	0.130825
18	**0.999957**	**10.5354**	**202.464**	**80.9787**	**27.0799**	**3614.95**	**0.0129291**
19	0.999178	10.5544	202.595	81.0891	27.1171	3617.96	0.246744
20	0.999849	10.5331	202.448	80.9649	27.0752	3614.58	0.045392

In 18th row we obtained the optimum solution which shown in table 5.4

Table 5.4 Optimum Result using GA for three generator test system

Fitness	**Losses**	**Power (P_i)**			**TC**	**Error**
		P_1	**P_2**	**P_3**		
0.999957	10.5354	202.464	80.9787	27.0799	3614.95	0.0129291

5.1.3 OPTIMUM SOLUTION USING FCGA FOR PROBLEM-I

The number of generations that evolve depends on whether an acceptable solution is reached or a set number of iterations is exceeded. After a while all the chromosomes and associated costs would become the same if it were not for mutations. At this point, the algorithm should be stopped.The results for final iteration using FCGA are shown in table 5.5 and the optimum result for three generators are shown in table 5.6

Table 5.5 ELD Result using FCGA for three generator test system

S no.	Fitness	Losses	Power (P_i)			TC	Error
			P_1	P_2	P_3		
1	0.999841	10.5351	202.462	80.9767	27.0792	3614.9	0.047566
2	0.998146	10.577	202.752	81.2211	27.1616	3621.56	0.557353
3	0.99988	10.5388	202.487	80.9984	27.0865	3615.49	0.0334451
4	0.999873	10.5392	202.49	81.0004	27.0872	3615.54	0.0380824
5	0.99907	10.5562	202.611	81.1029	27.1217	3618.34	0.279199
6	0.99978	10.5412	202.504	81.0122	27.0912	3615.87	0.065906
7	0.999725	10.5304	202.429	`80.9491	27.0699	3614.15	0.0824935
8	0.999595	10.5452	202.532	81.0358	27.0992	3616.51	0.121551
9	0.997228	10.4757	202.05	80.6297	26.9621	3605.45	0.834038
10	**0.999911**	**10.5344**	**202.457**	**80.9728**	**27.0779**	**3614.79**	**0.0268417**
11	0.999116	10.5554	202.604	81.097	27.1198	3618.17	0.26529
12	0.998885	10.5608	202.639	81.1265	27.1297	3618.98	0.334835
13	0.999756	10.531	202.434	80.953	27.0713	3614.26	0.0732181
14	0.999873	10.5392	202.49	81.0004	27.0872	3615.54	0.0380824
15	0.999509	10.5256	202.396	80.9215	27.0606	3613.4	0.147424
16	0.997884	10.5828	203.791	81.2547	27.1728	3622.47	0.636151
17	0.999571	10.527	202.405	80.9294	27.0633	3613.61	0.128872
18	0.998901	10.5605	202.637	81.1246	27.129	3618.93	0.330199
19	0.999641	10.5442	202.525	81.0299	27.0972	3616.35	0.10764
20	0.999734	10.5422	202.511	81.018	27.0932	3614.03	0.0798175

Table 5.6 Optimum Result using FCGA

Fitness	Losses	Power (P_i)			TC	Error
		P_1	P_2	P_3		
0.999911	10.5344	202.457	80.9728	27.0779	3614.79	0.0268417

5.2 PROBLEM II: TEN GENERATOR TEST SYSTEM

Again the proposed technique has been performed on a sample system which consists of ten generator system. The power demand is considered to be 1440MW. The coefficients of fuel cost are given below in Tables 5.7(a). Maximum and minimum power limits are given in Table 5.7(b). Transmission loss coefficients are taken from **[4].**

Table 5.7 (a)

Unit no	a_i	b_i	c_i
1	0.001220	7.92	630
2	0.004700	7.91	190
3	0.001320	7.93	625
4	0.001153	7.92	723
5	0.001154	7.93	717
6	0.001562	7.92	561
7	0.001153	7.92	723
8	0.001321	7.91	618
9	0.001319	7.00	561
10	0.001530	7.00	561

Table 5.7 (b)

Limits	Max	Min
P_1	700	160
P_2	300	65
P_3	680	150
P_4	780	170
P_5	750	160
P_6	600	130
P_7	780	170
P_8	670	145
P_9	640	140
P_{10}	590	120

TABLE-5.7 Specifications of 10-generator test
(a) Cost coefficients
(b) Power generation limits

The simulated results are presented for various problem using 10-generator sample test system. The results corresponding to Lambda Iteration method, GA and FCGA are detailed in section 5.2.1, 5.2.2 and 5.2.3 respectively.

5.2.1 OPTIMUM SOLUTION USING LIM FOR PROBLEM-II

TABLE-5.8 Results of ELD using LIM for ten generator test system

TC	Power (P_i)										Losses
	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	
17608.4	160	65	150	170	160	130	170	145	140	163.926	13.9357

5.2.2 OPTIMUM SOLUTION USING GA FOR PROBLEM-II

The results of the above said problem has also been obtained by utilizing GA. The length of string, population of string, crossover probability p_c, and mutation probability p_m are same as problem-I. The results of ELD obtained by applying the GA is shown in table 5.9 and the optimum result is shown in table 5.10. For this optimal result the power distribution to the ten generators is shown in table 5.11

Table 5.9 ELD Result using GA for ten generator test system

S.no	Fitness	Losses	Total cost	Error
1	0.999954	13.9364	17609	0.0664733
2	0.998255	13.961	17627.6	2.51773
3	0.998328	13.912	17590.2	2.41143
4	0.998487	13.9142	17591.9	2.18148
5	**0.999976**	**13.9361**	**17608.7**	**0.0345486**
6	0.999378	13.9447	17615.3	0.896448
7	0.999895	13.9343	17607.3	0.150618
8	0.999537	13.9424	17613.5	0.666622
9	0.999289	13.946	17616.2	1.02413
10	0.999254	13.9465	17616.6	1.0752
11	0.999957	13.9351	17608	0.0612264
12	0.999972	13.9362	17608.8	0.0409336
13	0.999369	13.9448	17615.4	0.909216
14	0.999263	13.9463	17616.5	1.06243
15	0.999272	13.9462	17616.4	1.04966
16	0.999922	13.9346	17607.6	0.112307
17	0.999945	13.9365	17609.1	0.0792431
18	0.999378	13.9447	17615.3	0.896448
19	0.999839	13.9381	17610.2	0.232478
20	0.999949	13.9365	17609	0.0728582

Table 5.10 Optimal Result using GA

Fitness	Losses	Total cost	Error
0.999976	13.9361	17608.7	0.0345486

Table 5.11 Power distribution of optimum result using GA

P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	TP
160	65	150	170	160	130	170	145	140	163.971	1453.971

5.2.3 OPTIMUM SOLUTION USING FCGA FOR PROBLEM-II

The final iteration results of FCGA for ten generators are shown in table 5.12, and the optimum result is shown in table 5.13

Table 5.12 ELD Results using FCGA for ten generators test system

S. No	Fitness	Losses	Total cost	Error
1	0.999972	13.9362	17608.8	0.0409336
2	0.999723	13.9318	17605.4	0.399643
3	0.999669	13.931	17604.9	0.476271
4	0.999935	13.9348	17607.8	0.0931518
5	0.999365	13.9449	17615.4	0.9156
6	0.999351	13.9451	17615.6	0.934751
7	0.999356	13.945	17651.5	0.934851
8	0.999953	13.9351	17608	0.0676114
9	0.999918	13.9346	17607.6	0.118692
10	0.9978	13.9676	17632.5	3.17504
11	0.997818	13.9673	17632.3	3.14951
12	**0.999989**	**13.9356**	**17608.3**	**0.0165312**
13	0.995327	14.0043	17659.8	6.76013
14	0.997615	13.9703	17634.6	3.44304
15	0.999966	13.9353	17608.1	0.0484563
16	0.999949	13.9365	17609	0.0728582
17	0.998647	13.9165	17593.7	1.95153
18	0.998638	13.9164	17593.6	1.96431
19	0.998633	13.9163	17593.5	1.97069
20	0.998288	13.9114	17589.8	2.46892

Table 5.13 Optimum result using FCGA

Fitness	**Losses**	**Total cost**	**Error**
0.999989	13.9356	17608.3	0.0165312

For this optimum result the power distribution to the ten generators are shown in table 5.14

Table 5.14 Power distribution of optimum result using FCGA

P_1	**P_2**	**P_3**	**P_4**	**P_5**	**P_6**	**P_7**	**P_8**	**P_9**	**P_{10}**	**TP**
160	65	150	170	160	130	170	145	140	163.919	1453.919

CHAPTER-6

CONCLUSIONS AND FUTURE SCOPE

6.1 CONCLUSION

In this project, Fuzzy Logic Controlled Genetic Algorithm has been successfully introduced to obtain the optimum solution of ELD. Power system has large variation in load from time to time and it is not possible to have the load dispatch for every possible load demand.As there is no general procedure for finding out the optimum solution of economical load dispatch. This is where FCGA plays an important role to find out the optimum solution in a fraction of second. In the proposed method two fuzzy controllers have been designed to adaptively adjust the crossover probability and mutation rate during the optimization process based on some heuristics.

For the testing of proposed algorithm, three generators and ten generators test systems are used. The results obtained from proposed method are also compared with conventional GA and the Lambda Iteration Method.

It is found that FCGA is giving better results than GA and LIM. i.e. FCGA proves their fast algorithm and yields true optimum generations of both operating costs and transmission line losses of the power system.

6.2 FUTURE WORK

Having gone through the study of fuzzy logic controlled genetic algorithm for Economic Load Dispatch, the scope of the work has been identified as –

1. extend the fuzzy logic controlled genetic algorithm based ELD to truly the large number of units i.e 30 or even higher units
2. extend the problem of ELD by using the TABU search and also with the hybrid of Fuzzy logic controlled TABU search
3. extend the fuzzy logic controlled genetic algorithm based ELD solution by including the various Facts devices

REFERENCES

1. Prasanna T.S., Somasundaram P., “Fuzzy Mutated Evolutionary Programing based algorithm for combined economic and emission dispatch”. IEEE international conference, TENCON-2008, November 2008, in press.
2. Chowdhury B.H., Rahman Saifur, “A Review of Recent Advances in Economic Dispatch”, IEEE Transactions on Power Systems, Vol. 5, No. 4, November 1990
3. Dhillon J.S., Parti S.C. and Kothari D.P., “Multiobjective Stochastic Optimal thermal power dispatch”, ibid, pp. 136-140, 1987.
4. Roa C.A-Sepulveda, Herrera M., Pavez-Lazo. B, Knight U.G., Coonick A.H., “Economic Dispatch using fuzzy decision trees”, Electric Power Systems Research 66 (2003) 115_ 122, Dec 2002
5. Palanichamy C. and Shrikrishna K. (1991), “Simple algorithm for economic power dispatch” electrical power syst. Res, Vol.21 pp 174-153.
6. Wood Allen.J. & Woolenberg Bruce.F. (1996), “power generation, operation and Control”, johnwilley & sons.
7. King T.D., El-Hawary M.E. and El-Hawary F., “Optimal environmental dispatching of electric power systems via an improved hopfield neural network model.” IEEE Transactions on Power Systems, vol. 10, no. 3, pp.1559-1565, Aug. 1995.
8. “Canada/Nova Scotia Agreement Respecting an Acid Rain Reduction Program Nova Scotia 1991 Annual Report", Nova Scotia Department of the Environment, 1991,67
9. Kermanshahi B.S., Wuch Y., Yasuda K. and Yokoyama R., “Environmental Marginal Cost Evaluation By Non-inferiority Surface,” IEEE Transactions on Power Systems, vol. 5, no.4, pp. 1151-1159, Nov. 1990.
10. Gent M.R. and Lamont J., “Minimum-Emission Dispatch,” IEEE Transactions on Power Apparatus and Systems, vol. PAS-90, pp. 2650-2660, Nov.-Dec. 1971.

11. Hess S.W., Parker D., Alms John E., Le K.D., Day J.T. and Malone M.J., "Planning System Operations to meet NOx constraints", IEEE Computer Applications in Power, vol. 5, no.3, pp. 10-14, July 1992.
12. Hamid Bouzeboudja, Abdelkader Chaker, Ahmed Alali, Bakhta Naama., "Economic Dispatch Solution Using A Real-Coded Genetic Algorithm," Acta Electrotechnica et Enformatica No. 4, Vol. 5, 2005
13. Hopfield J.J. and Tank D.W., "Simple neural optimization networks: an A/D converter, signal decision network, and a linear programming circuit," IEEE Transactions on Circuit and Systems, vol. CAS-33, pp. 533-541, May 1986.
14. Abe S., "Global convergence and suppression of spurious states of the hopfield neural networks," IEEE Transactions on Circuits and Systems, vol. CAS-40, 1993, pp. 246-257.
15. Su C.-T. and Chiou G.-J., "A fast - computation hopfield method to economic dispatch of power system," IEEE Transactions on Power Systems, vol. 12, no. 4, Nov., 1997.
16. Walsh M. P. and Malley M. J, O', "Augmented Hopfield network for unit commitment and economic dispatch," IEEE Transactions on Power Systems, vol. 12, no. 4, pp. 1765-1774, 1997.
17. Gall A. L. and Zissimopoulos V., "Extended Hopfield models for Combinatorial Optimization," IEEE Transactions on Neural Networks, vol. 10, no. 1, Jan.1999.
18. Su C.T. and Lin C.T., "New approach with a Hopfield modeling framework to economic dispatch", IEEE Transactions on Power Systems, vol. 15, 2000.
19. Silva I. N., Arruda L. V. R. and Amaral W. C., "A modified Hopfield architecture for solving nonlinear programming problems with bounded variables, Neural Network World, vol. 4, no. 0, pp. 713-722, 2000.
20. Farooqi M.R., Jain P. and Niazi K.R., "Using Hopfield Neural Network for Economic Dispatch of Power Systems," IEEE Power Engineering Conference,2003. PECon 2003. Proceedings, Bangi, Malaysia, pp. 5-10, 15-16 Dec. 2003.

21. Yalcinoz T. and Altun H., “Comparison of simulation algorithms for the Hopfield neural network: an application of economic dispatch,” Turkish Journal of Electrical Engineering & Computer Sciences, vol. 8, no.1, pp. 67-80, 2000.
22. Park J.H., Kim Y.S., Eom I.K. and Lee K.Y., “Economic Load Dispatch for Piecewise Quadratic Cost Function Using Hopfield Neural Network”, IEEE Transactions on Power Systems, vol. 8, no. 3, pp. 1030-1038, Aug.1993.
23. King T.D., “Optimal Environmental Dispatching of Electric Power Systems Using A New Hopfield Model”. M. A. Sc. Thesis, Technical University of Nova Scotia, Halifax, 1994. 66
24. http:// en.wikipedia.org/wiki/Genetic_algorithm
25. Mitchell M., An Introduction to Genetic Algorithm. MIT Press, 1998.
26. Cai, Xinye., “A Multy-Objective GP-PSO Hybrid Algorithm For Gene Regulatory Network Modeling” May-2009
27. Kosko B., “Bidirectional Associative Memories”, IEEE Transactions on Systems, Man and Cybernetics, vol. SMC-18, no. 1, pp.49-60, 1988.
28. http://www.cs.cmu.edu/Groups/AI/html/faqs/ai/fuzzy/part1/faq-doc-2.html
29. Zadeh LA. Fuzzy sets. Information Control 1965; 8:338–353
30. Birkhoff G.D., “Lattice Theory”. Vol. XXV, 2nd ed. Providence, RI: Colloquium Publications, 1948.
31. Goguen JA. L-fuzzy sets. J Math Anal Appl 1967; 18:145–174.
32. Vijayalakshmi. G.A., Rajsekaran. S, “Neural Networks, Fuzzy Logic, and Genetic Algorithms” synthesis and application
33. E. Cox, “The Fuzzy Systems Handbook”, Morgan Kaufmann Publishers, 1998.
34. George. J Klir/Boyuan, “Fuzzy Sets and Fuzzy Logic”, prentice Hall of India Private Limited, New Delhi - 2000.
35. Mohammad Jamshidi, Vadice Nader and Ross. T.J, “Fuzzy Logic And Control”, New Jersey,1993.
36. Yalcinoz. T, Altun. H, and Uzam. M, “Economic dispatch solution using genetic algorithm based on arithmetic crossover,“in Proc. IEEE Porto Power Tech. Conf., Porto, Portugal, Sep. 2001.

37. Pavej.B-Lazo, Roa .C.A. -Sepulveda, Herrrera. M, Coonick. A.H, “Economic Dispatch using Fuzzy Decision Trees”, electric power system research 66 (2003) 115-122
38. Sheble G.B and Britigg K., “Refined genetic algorithm-economic dispatc exmple”, IEEE Trans. Power Systems, vol. 10, no. 1, pp. 117- 124, Feb.1995.
39. Lee .S.K, Son K. M and Park K. K., “Voltage profile solution of economic dispatch with valve point loading,” In proc. IASP 97, Seoul, Korea, July 6-10, 1997, pp 407- 414.
40. Bakirtzis. A, Petridis.V, and Kazarlis.S, july 1904 “Genetic Algorithm solution to the Economic Dispatch problem”,IEE Proc-Gener, vol.141, No.4,
41. Hazra Jagabondhu and Sinha Avinash, IEEE member, “Application of soft computing methods for Economic Dispatch in Power Systems”, International Journal of Electrical Power and Energy Systems,2009
42. Lee. K.Y, Bai. X and Park. Y.M, “Optimization method for Reactive power Planning by Modified Simple Genetic Algorithm” IEEE Transactions on Power Systems Vol.10, No.4, November1995
43. Song Y.H., Wang G.S., Wang P.Y. and Johns A.T., “Environmental economic dispatch using fuzzy logic controlled genetic algorithms,” IEE Proc. Generation Transmission Distribution, vol. 144, pp. 377-382, July 1997.
44. Kazuko Y., “Genetic Algorithms under Dynamic Environments,” Fourth International Conference on knowledge – based Intelligent Engineering Systems and Allied technologies,” vol.101, no.66, pp.55-62, 2001.
45. 45. Goodman E. D., “Introduction to Genetic Algorithm,” GECCO 2007 Tutorial, pp. 1-19.
46. John R. K., “Genetic Programming,” Encyclopaedia of Computer Science and Technology, version 2, 1997.
47. Sadati N. and Taheri J., “Genetic Algorithm in Robot Path Planning Problem in Crisp and Fuzzified Environments,” IEEE International Conference on Industrial Technology, Bangkok, Thailand, IEEE ICIT '02. vol. 1, pp. 175- 180, 2002.

48. Yalcinoz T. and Altun H., “Environmentally constrained economic dispatch via a genetic algorithm with Arithmetic crossover,” IEEE 6th Africon Conference in Africa 2002, vol. 2, pp. 923 – 928, 2002.
49. AL-Kandari A.M. and El-Naggar K.M., “A genetic-based algorithm for optimal estimation of input-output curve parameters of thermal power plants,” Springer Berlin / Heidelberg Journal of Electrical Engineering, vol. 89, no. 8, pp. 585-590, Sept., 2007.
50. Coelho L.S. and Lee C.-S., “Solving economic load dispatch problems in power systems using chaotic and Gaussian particle swarm optimization approaches,” Electrical Power and Energy Systems, vol. 30, pp. 297-307, 2008.
51. Wang L. and Singh C., “Stochastic economic emission load dispatch through a modified particle swarm optimisation algorithm,” Electric Power Systems Research, vol. 78, pp. 1466-1476, 2008.
52. . Park J.-B., Lee K.-S., Shin J.-R. and Lee K.Y., “A Particle Swarm Optimisation for Economic Dispatch with Nonsmooth cost functions,” IEEE Transactions on Power Systems, vol. 20, no.1, Feb. 2005.
53. Thitithamrongchai C. and Eva-Arporn B., “Hybrid Self-Adaptive Differential Evolution Method with Augmented Lagrange Multiplier for Power Economic Dispatch of units with Valve-Point Effects and Multiple Fuels,” IEEE Power Systems Conference and Exposition, 2006. PSCE '06. 2006 IEEE PES, 2006. pp. 908-914, Oct. 29 - Nov. 1, 2006.
54. Ouyang Z. and Shahidehpour, S.M., “A hybrid artificial neural networkdynamic approach to unit commitment,” IEEE Transactions on Power Systems, pp. 236-242, 1992.
55. Benasla.L, Belmadant.A and Rahli M., “Hooke-Jeeves’Method Applied to a New Economic Dispatch Problem Formulation” journal of information science and engineering 24, 907-917 (2008)
56. Sushil K., Naresh R., “Non Convex Economic Load Dispatch using an Efficient Real Coded Genetic Algorithm”, elsevier, 2008

57. Bouktir.T, Mahdad.B, and Srairi.K., "Fuzzy Controlled Genetic Algorithm For Envirenmental/Economic Dispatch with Shunt FACTS Devices" IEEE press, pp.201, 2003
58. Kothari D.P., Dhillon J.S., "Power System Optimization", Prentice-Hall of India
59. J.Nanda & R.Badri Naryan, "Application of Genetic Algorithm to Economic Load Dispatch with line flow constraints" from elsevier.com nov 2001
60. Zhuang F. and Galiana F.D., "Unit Commitment by Simulated Annealing," IEEE Transactions. on Power Systems, vol. 5, no.1, pp. 311-318, 1990.
61. D.B. Fogel, and K. Chellapilla, "Revisiting Evolutionary Programming", roc. Of Applications and Science of Computional Intelligence. Orlando, pp. 2-11, 1998.
62. D.B. Fogel, Evolutionary Computation: "Toward a New Philosophy of Machine Intelligence", second ed., IEEE Press, Piscataway, NJ, 2000.
63. Salami, and G. Cain, "Multiple Genetic Algorithm Processor for The Economic Power Dispatch Problem", Proceeding of the Genetic Algorithm in Engineering Systems: Innovations and Applications, Conference Publication, No. 414, IEE, 1995, pp 188-193.