FUEL COST FUNCTION ESTIMATION FOR ECONOMIC LOAD DISPATCH USING EVOLUTIONARY PROGRAMMING

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

Master of Engineering
in
Power Systems & Electric Drives

Submitted By
Cherry Mittal
(Roll No. 800941006)

Under the supervision of:
Mrs. Manbir Kaur
Assistant Professor (EIED)

ELECTRICAL & INSTRUMENTATION ENGINEERING DEPARTMENT
THAPAR UNIVERSITY, PATIALA – 147004
June 2011

Certificate

I hereby certify that the work which is being presented in the thesis entitled **"Fuel Cost Function estimation for Economic Load Dispatch using Evolutionary Programming"** in partial fulfillment of award of degree of **Master of Engineering** in **Power Systems and Electric Drives** submitted in Electrical and Instrumentation Engineering department, Thapar University, Patiala is an authentic record of my own work carried under the supervision of **Mrs. Manbir Kaur**, Assistant Professor, Electrical and Instrumentation Engineering department, Thapar University, Patiala, Punjab.

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

Cherry

Cherry Mittal
(800941006)

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

Manbir Kaur
5-7-2011

Mrs. Manbir Kaur
Assistant Professor,
Electrical and Instrumentation Engineering department
Thapar University
Patiala

Countersigned By:

S. Ghosh
14/7/11

(Dr. Smarajit Ghosh)
Professor & Head,
Electrical and Instrumentation Engineering department
Thapar University
Patiala

(Dr. S.K Mohapatra)
Dean of Academic Affairs
Thapar University
Patiala

ACKNOWLEDGEMENT

This dissertation would not have been completed without the help of many people who contributed directly or indirectly through their encouragement in the evolution and preparation of this work.

First and foremost, I thank the almighty, who gave me the opportunity, courage and patience to carry out this work.

I am grateful to my supervisor, Mrs Manbir Kaur, Assistant Professor, Electrical and Instrumentation engineering department, for her active role and leadership that was instrumental in directing this work to a successful conclusion. I also acknowledge her for her excellent supervision, time, motivation and unlimited support and continuous encouragement throughout my thesis work.

I would also like to offer my sincere thanks to Dr. Smarajit Ghosh, Professor & Head, Electrical and Instrumentation engineering department for his ever supporting and encouraging nature.

I am also thankful to the staff of department for their cooperation throughout my thesis work.

Last, but not the least a very special thanks to my parents and friends for their constant encouragement and best wishes, their patience and undertaking, without which this work would have not been in this present form.

Cherry Mittal
(800941006)

ABSTRACT

The input–output characteristics of thermal power plants are affected by many factors such as the ambient operating temperature and aging of generating units. Thus, periodical estimation of fuel cost function is very crucial to improve the overall operational and economical practices. The higher the accuracy of the estimated coefficients of the cost function, the more accurate the results obtained from the economic dispatch and optimal power flow calculations.

Different models that describe the input–output relationship of thermal units are considered, including the one that accounts for the valve loading point. The goal is to minimize the total estimation error such that the selected model follows the actual data measurements as closely as possible. Evolutionary Programming (EP) is employed to minimize the error associated with the estimated parameters. It relieves some of the mathematical restrictions typically imposed on system modelling, since it does not require convexity or differentiability, as in the case of many conventional estimation techniques. Results obtained are compared to those computed by the least error square method and particle swarm optimization.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Overview

Optimal power flow and economic dispatch are important tasks in power system planning and operation. Both tasks can be formulated mathematically as optimization problems with the objective of minimizing the fuel cost function. It is obvious that the final accuracy of the process depends on many factors such as the selected model, the data and parameters used, as well as the technique or the tool used for solving the formulated problem. The input–output characteristics of thermal power plants are affected by many factors such as the ambient operating temperature and aging of generating units. Thus, periodical estimation of power plant characteristics is very crucial to improve the overall operational and economical practices. Many utilities are using outdated information to describe the input–output relationship of generating units [1]. One of the most important issues in solving the economic dispatch problem is to have an accurate estimate of the thermal unit input–output curve coefficients. The accuracy of the estimated coefficients adversely affects the final accuracy of the dispatch process. Therefore, it is very important to use an accurate and reliable estimation technique for estimating the input–output curve coefficients. There are different mathematical models commonly used for fuel cost function representation. These models can be grouped into two main categories. The first category includes smooth models, while the second includes non-smooth models. Both models, smooth and non-smooth, are presented and discussed.

Research efforts in the area of parameter estimation have been continually carried out. Several methods have been proposed to solve estimation problems in power systems. Some of these techniques are based on static estimation techniques, such as least error square (LES) and least absolute value; others are based on dynamic estimation techniques such as the Kalman filter. While the LES technique has been the most famous static estimation technique and in use for a long time as the preferred technique for optimum estimation in general, some limitations and disadvantages are associated with this approach. For example, when the data set is contaminated with bad measurements, the estimates may be inaccurate unless a large number of data points are used [2, 3].

Methods based on artificial intelligence, such as artificial neural networks (ANNs), expert systems, and genetic algorithms, have also been used in power system state estimation. In general, ANNs are superior when the process model is not well defined mathematically. These classes of methods can cope with non-smooth optimization problems, considering non-convex or discontinuity functions like the fuel-cost function with valve point effects. The practical determination of accurate real break points is not easy; however ANN-based methods suffer from many problems. One of the drawbacks of such techniques is the huge amount of data required for network training, which may not be available in some cases [7, 8].

Thus we use Evolutionary Programming (EP) for estimating the thermal unit input–output curve coefficients .The estimation problem is formulated as an optimization one. The goal is to minimize the total error in the estimated state parameters. In this study, both smooth and non smooth functions are presented. An EP based algorithm is used to find the optimum parameter estimation of the formulated optimization problem.

1.2 Literature review

Optimal power flow and economic dispatch are important tasks in power system planning and operation. Both tasks can be formulated mathematically as optimization problems with the objective of minimizing the fuel-cost function. The final accuracy of the process depends on many factors such as the selected model, the data and parameters used, as well as the technique or the tool used for solving the formulated problem.

Alrashidi *et. al*. [1] discussed that one of the most important issues in solving the economic dispatch problem is to have an accurate estimate of the thermal unit input–output curve coefficients. The accuracy of the estimated coefficients adversely affects the final accuracy of the dispatch process as proposed by Chen and Schweppe [2, 3] Therefore, it is very important to use an accurate and reliable estimation technique for estimating the input–output curve coefficients.

Soliman *et. al.*[4] presented an algorithm for estimating the coefficients of a fuel cost I/O curve used for economic dispatch calculation. A special constraint equation is included to ensure that the incremental cost curve exhibits monotonically increasing characteristics. The number of linear segments and their corresponding «break points» are determined in an optimal manner.

Kandari and Duran [5, 6] presented the application of Kalman filtering algorithm for online identification of the input/output curves for thermal power plants. The advantage of the approach is the capability to keep tracking the state of the thermal unit by making the coefficients of I/O curve continuously adaptive to the real characteristic of the unit. Different models for the I/O curve are used, including polynomial models of different orders as well as non linear models are presented by Attaviriyanupa and Liu [7, 8].

Any types of unit characteristics cost curve may be used with adjustments to the objective function are proposed by Hawary *et. al*. [9-11]. Quadratic input-output curves were used initially to develop the software and to verify that optimal solutions are obtained with this algorithm.

Fink and Walter [12, 13] proposed a genetic based algorithm to solve economic dispatch problem for valve point discontinuities. They compared the program based on GA using two different encoding techniques.

Eberhart and Kennedy [14, 15] introduced a method for optimization of continuous non linear function.

Eberhart et al [16-20] proposed four algorithms for estimation of the parameters of model used in optimal economic operation of electric power systems.

Shout and Mead [21] and suggested a very practical and efficient method of estimating the coefficients of heat rate curve represented by a cubic order polynomial. Jayabharti *et al* [22] proposed that this method required matrix inversion and is only suitable for off line usage rather than online application.

A simple and efficient algorithm based on evolutionary strategies is proposed by Neto *et. al*.[23] for the solution of the economic dispatch (ED) problem with non continuous and non smooth/non convex cost functions and with generator constraints being considered.

Evolutionary programming has emerged as a useful optimization tool for handling nonlinear programming problems. Various modifications to the basic method have been proposed with a view to enhance speed and robustness. The performance of evolutionary programs on ELD problems is examined and presented by Sinha *et al*.[24-26].

A novel evolutionary programming (EP) algorithm by Goldberg and Mori [27, 28] is proposed to solve the ED problem for units with non-smooth fuel cost units. The stochastic mechanics, which combine offspring creation based on the performance of current trial solutions and competition & selection based on successive generations, form a considerably robust scheme for large-scale real-valued combinatorial optimization.

In the self adaptation scheme of EP, Yao *et al*. [29] proposed a Cauchy-mutation-based parameter is not prefixed rather it is evolved along with the objective variables EP, called fast EP (FEP), which demonstrated much better performance than Classical EP in converging to a near-global optimum point on some benchmark mathematical functions, Chellapilla and Fogel [30] proposed a fast Evolutionary Programming using the weighted mean of Gaussian and Cauchy mutations with an objective of having a step size greater than Gaussian mutation and smaller than Cauchy mutation so that advantages of both Gaussian as well as Cauchy mutations can be exploited.

Yao *et al.* [31] proposed an improved FEP (IFEP) that uses both Gaussian and Cauchy mutations to create offspring from the same parent and better ones are chosen for next generation. This method outperforms the other EP methods in almost all the benchmark functions they studied.

Wolpert and Macready [32] have shown that, under certain assumptions, no search algorithm is best for all problems. Thus, the success and quality of a solution achieved by an EP algorithm is dependent on the step size, the method of updating, and the nature and number of suboptimal points, as well as their distances with respect to the globally optimum point and other factors.

1.3 Objective of work

The objectives of the thesis work are summarised as follows –

1. To estimate the fuel cost function Parameters of thermal power plants using smooth and non-smooth model using least error square method.

2. Use global search techniques like Least Error Square / Evolutionary Programming to find the optimal settings.

3. Investigate the effectiveness of these methods for finding the fuel cost function parameters of smooth and non smooth functions and to compare the results obtained from the Evolutionary Programming with the results obtained from Least Error Square Method and PSO.

1.4. Organization of Thesis

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

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

The **chapter-2** highlights modelling of fuel cost curves and also gives the overview of Least Error Square method .Also estimation of the Fuel cost parameters using LES method are discussed in this chapter.

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

The **chapter-4** outlines the simulated results using LES and EP for smooth and non smooth models. Comparison of the results from these techniques is done.

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

CHAPTER **2**

MODELLING OF FUEL COST FUNCTION

2.1 Smooth Fuel-cost Function

In this case, the fuel-cost curve for a thermal generating unit (i) as a function of the MW output power can be modelled by a polynomial function in an idealized form as:

$$F_i(P_{ti}) = a_{0i} + \sum_{J=1}^{L} a_{ji} \, P_{ti}^{j} + r_i \,, \qquad i = 1, 2, \ldots \ldots \ldots . N \qquad (2.1)$$

Where

F_i is the fuel-cost function of the i th generating unit.

P_{ti} is the power generated by the i th thermal generating unit.

a_{oi}, a_{ji} are the curve constants of the i th generating unit.

r_i is the error associated with the i th equation.

N is the number of thermal generating units.

L is the equation order (i.e., for the linear model it is 1, for the second order it is 2)

L is a power set as 1, 2, 3 or more.

2.2 Non-smooth Fuel-cost Function

In large steam turbine generating units, the input–output characteristics are neither always smooth nor differentiable everywhere. Large steam turbine generating units tend to have a number of steam admission valves that are opened in sequence to obtain an ever increasing output of the unit. In general, the valve point effects produce ripple, such as heat rate curve. The number of ripples is proportional to the degree of non-convexity in the shape of the objective function, and it increases the difficulty of detecting the global solution.

In this situation, the fuel-cost function is modelled by adding a sinusoid term [10, 11].

The new cost function becomes

$$F_i(P_{ti}) = \left[a_{0i} + \sum_{J=1}^{L} a_{ji} \; P_{ti}^{j} + r_i\right] + \left|e_i sin\left(f_i\left(P_{i,min} - P_i\right)\right)\right| \qquad \mathrm{i} = 1,2, \ldots \ldots \ldots \mathrm{N}$$

(2.2)

where e_i and f_i are the fuel-cost coefficients of the *i* th unit with valve point effects.

Although, the use of a non-smooth function increases the accuracy of the economic dispatch results but at the same time it adds to the complexity of computation.

Four different models are considered to examine the performance of the proposed method. The work in this thesis is focussed on the following models.

2.3 Smooth and Non Smooth Models

2.3.1 First-order model: In this case, the fuel-cost function is expressed as a linear function of the unit real power output which is shown in Eq.2.3

$$F_i(P_{ti}) = a_{0i} + a_{1i}\,P_{ti} + r_i \tag{2.3}$$

2.3.2 Second-order model: In this case L is set to 2 and then Eq. (2.1) takes the following quadratic form and is shown in Eq.2.4

$$F_i(P_{ti}) = a_{0i} + a_{1i}\,P_{ti} + a_{2i}P_{ti}^2 + r_i \tag{2.4}$$

2.3.3 Third-order model: In this case L is set to 3 and the equation takes the quadratic form which is represented in Eq.2.5

$$F_i(P_{ti}) = a_{0i} + a_{1i}\,P_{ti} + a_{2i}P_{ti}^2 + a_{3i}P_{ti}^3 + r_i \tag{2.5}$$

2.3.4 Non-smooth model: When accounting for the steam admission valve effects, the fuel cost is typically modelled as a quadratic function augmented with a sinusoidal term, as shown below [13] in Eq.2.6

$$F_i(P_{ti}) = \left[a_{0i} + a_{1i}\,P_{ti} + a_{2i}P_{ti}^2 + \left|e_i sin\left(f_i\left(P_{i,min} - P_i\right)\right)\right|\right] + r_i \tag{2.6}$$

Figure 2.1 shows different input–output characteristics for steam units [1] with the input to the unit on the ordinate and is concerned to be either in terms of heat energy requirements (GJ/hr) or in terms of total cost per hour ($/hr). The output is normally the net electrical power output of the unit (MW).

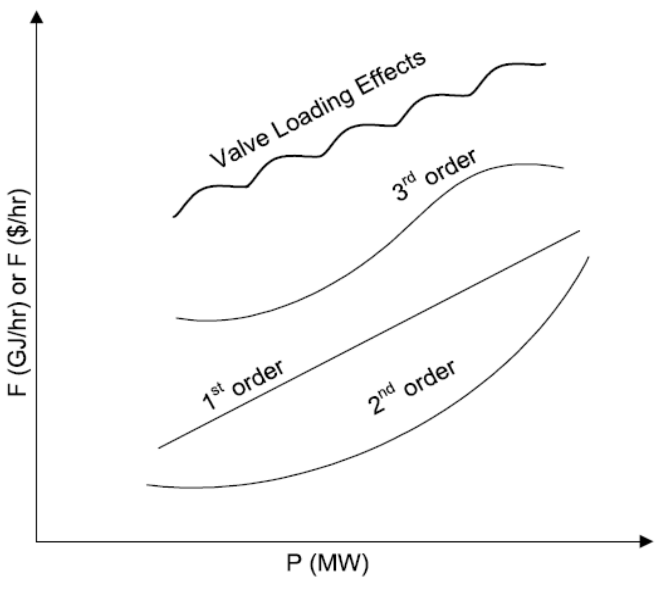


Figure 2.1 Input-output characteristics

The input–output relation (F -P) or (H-P) for each thermal generating unit for any of the four models, a discrete system of equations in state space form can then be written as

$$Z_i = f_i(P_i, X_i) + R_i \tag{2.7}$$

Where Zi is a fuel-cost vector for the i th generating unit (measurement vector), Xi are the parameters to be estimated (a, e, f) for the i th generating unit, and Ri is the error vector associated with Zi. The EP-based program is developed to find an estimate for the unknowns (X) in Eq. (2.7). These estimates are used to recalculate the fuel function ($F_{estimated}$) at each time step. Then, the error associated with each measurement can be calculated as mean square error expression.

$$r_i = F_{i(actual)} - F_{i(estimated)} \tag{2.8}$$

Now, the problem formulation can be clearly stated as to find an estimate for the parameter vector X, for any model, that minimizes the error vector r_i.

2.4 Least error square method

The method of least squares is used to solve a set of linear equations having more equations than unknown variables. Since there are more equations than variables, the solution will not be exactly correct for each equation rather, the process minimizes the sum of the squares of the residual errors. The method is very powerful and can be applied to numerous applications.

In the general case, the least-squares method is often used to solve a set of non-linear equations that have been linearized using a first-order Taylor-series expansion. Solving non-linear equations is an iterative process using Newton's method. The speed of convergence is dependent on the quality of an initial guess for the solution. The non-linear least-squares method is often referred to as a bundle adjustment since all of the values of an initial guess of the solution are modified together (adjusted in a bundle). This technique is also occasionally referred to as the Gauss-Newton method.

If the equations are linear, the least-squares process will produce a direct solution for the unknowns. If the equations are not linear, an initial guess of the unknowns is required, and the result is an adjustment to the initial parameters. This is repeated until the results converge (the adjustments become very close to zero). In linear case initial guess of all parameters is adjusted as zero.

2.4.1 Problem Formulation

The process requires a set of equations with the unknowns on one side and some known quantity on the other. Let x_i be the set of unknowns, and let the equations be of the form shown in Eq.2.9.

$$F_i(x_1, x_2, x_3, \dots.) = k_i \tag{2.9}$$

where k_i is the observation (value) whose least-square error will be minimized.

Since there are more equations than unknowns, the solution of the unknowns will not be exact. Using the solution to compute the equation, it will not generate the exact observation value, k_i. The square of the difference between the evaluated equation and the observation is minimized.

There is typically one equation for each observation. In photogrammetric, this might be one equation for each x pixel coordinate and one equation for each y pixel coordinate. Each equation is not required to have all of the unknowns in it. The Jacobian matrix, **J,** is the matrix of the partial differentials of each equation with respect to each unknown. That is,

$$J = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} & \frac{\partial F_1}{\partial x_3} & \cdots \\ \frac{\partial F_2}{\partial x_1} & \frac{\partial F_2}{\partial x_2} & \frac{\partial F_2}{\partial x_3} & \cdots \\ \frac{\partial F_3}{\partial x_1} & \frac{\partial F_3}{\partial x_2} & \frac{\partial F_3}{\partial x_3} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{2.10}$$

In general, the height of the Jacobian matrix will be larger than the width, since there are more equations than unknowns. Furthermore, let the vector **K** be the vector of the residuals. A residual is the difference between the observation and the equation calculated using the initial values. That is

$$K = \begin{vmatrix} k_1 - F_1(x_{10}, x_{20}, x_{30}, \dots) \\ k_1 - F_1(x_{10}, x_{20}, x_{30}, \dots) \\ k_1 - F_1(x_{10}, x_{20}, x_{30}, \dots) \\ \vdots \end{vmatrix} \tag{2.11}$$

One further parameter is a weighting matrix, **W**. This is a matrix which includes the expected confidence of each equation and also includes any dependence of the equations. A larger value in the weighting matrix increases the importance of the corresponding equation (larger values indicate greater confidence). It is a square symmetric matrix with one row per equation. The main diagonal contains the weights of the individual equations, while the off-diagonal entries are the dependencies of equations upon one another. If all of the observations are independent, this will be a diagonal matrix. The cofactor matrix, **Q**, is the inverse of the weighting matrix. ($Q=W^{-1}$)

Let x_{i0} be an initial guess for the unknowns. The initial guesses can have any finite real value, but the system will converge faster if the guesses are close to the solution. Also, let **X** be the vector of these initial guesses. That is

$$X = \begin{bmatrix} x_{10} \\ x_{20} \\ x_{30} \\ \vdots \end{bmatrix} \tag{2.12}$$

It is desirable to solve for the adjustment values Δ X**.** This is the vector of adjustments for the unknowns

$$\Delta X = \begin{vmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \\ \vdots \end{vmatrix} \tag{2.13}$$

Where, based on the initial guess X and an adjustment ΔX is a set of new values are computed

$$X' = X + \Delta X \tag{2.14}$$

To solve for Δ**X**

$$\Delta X = (J^t W J)^{-1}(J^t W J) \tag{2.15}$$

In various texts, the normal matrix, N is defined as

$$N = (J^t W J) \tag{2.16}$$

and the covariance matrix (sometimes referred to as the variance-covariance matrix), Q_{xx} is defined as

$$Q_{xx} = (J^t W J)^{-1} = N^{-1} \tag{2.17}$$

The initial guesses, x_{i0} are updated using the solution of the adjustment matrix, Δ**X** as follows

$$X' = X + \Delta X \tag{2.18}$$

Or

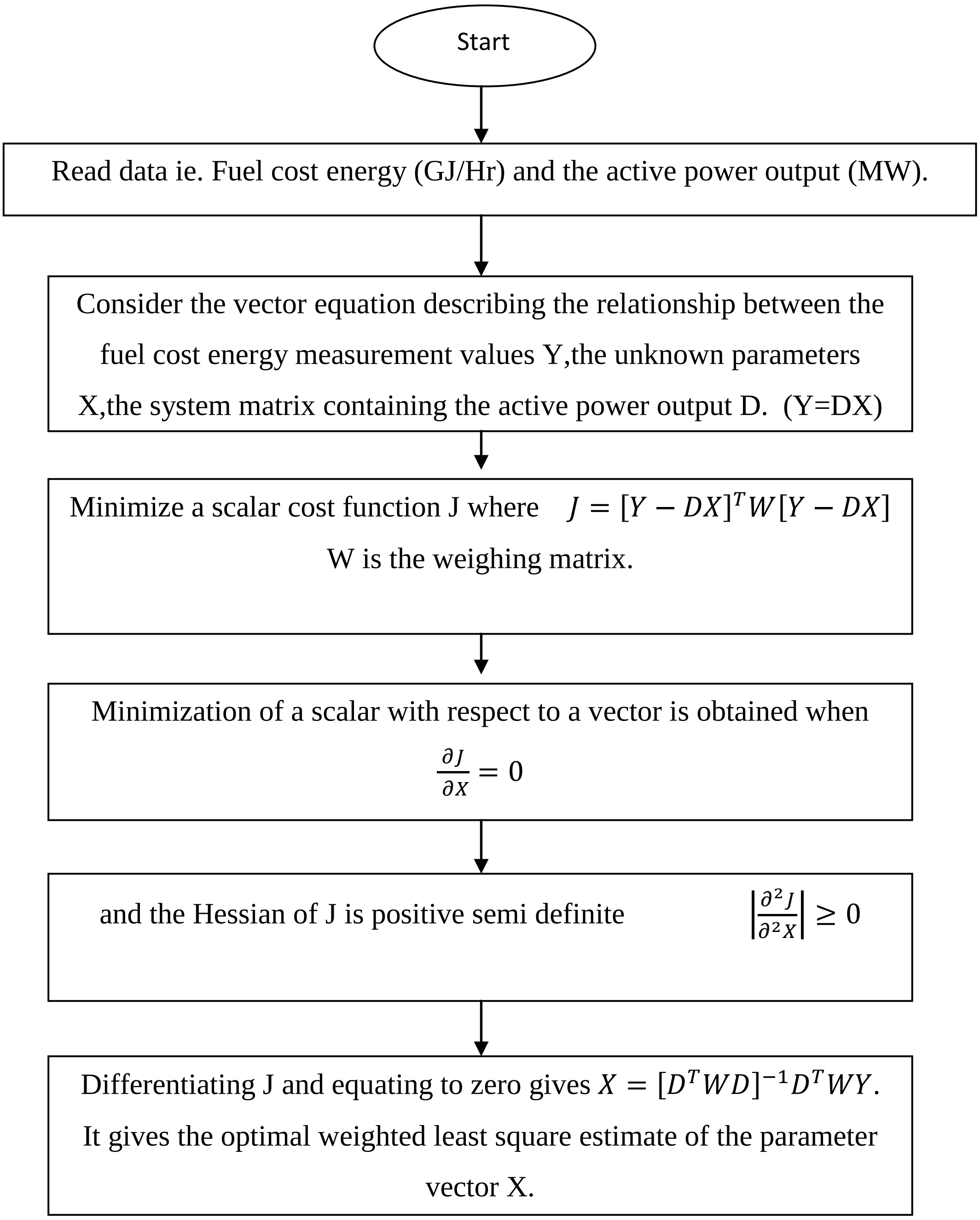
$$x_i' = x_{i0} + \Delta x_i \tag{2.19}$$

The process is repeated using the new values, x_i' as the initial guesses until the adjustments are close to zero.

There are conditions where the solution will not converge or will converge to undesirable values. This process finds a local minimum. As such, there may be a better solution than the one found. A solution is dependent on the equations $F_i(x_1, x_2, x_3, \dots.)$ being continuous in

$(x_1, x_2, x_3, \ldots.)$. The first and second derivatives do not need to be continuous, but if the equations are not continuous, there is no guaranty that the process will converge. Also, in certain circumstances, even if the equations are continuous, the solution may not converge. This can happen when the first and second derivatives of the equations have significantly different values at the initial values than at the solution. In any case where the solution does not converge, a solution may still be able to be obtained if different starting values, x_{i0} are used.

The flowchart for estimating the fuel cost function parameters for thermal power plants for smooth and non-smooth functions is shown in figure 2.2.



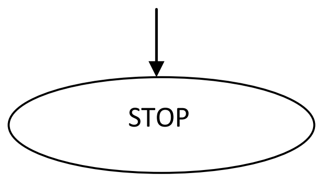


Figure 2.2 Flowchart for Least Error Square method

CHAPTER 3

ESTIMATING THE FUEL COST FUNCTION USING EVOLUTIONARY PROGRAMMING

3.1 Introduction

Evolutionary programming is based on Finite State Machine (FSM) model in its early stage, which is proposed by L. J. Fogel in 1960's when he studied the artificial intelligence. In 1990’s the evolutionary programming was developed by D. B. Fogel and was made to solve the optimal problems in real space. The evolutionary programming bas been become an optimal tool and is used in many practical problems. Evolutionary programming is a probabilistic, global search technique that starts with a population of randomly generated candidate solutions and evolves towards better solutions over a number of generations or iterations. The main stages of this technique include initialization, mutation, competition and selection. . According to the problem, each step could be modified and configured in order to achieve the optimum result.

3.2 Overview of Evolutionary Programming

Evolutionary Programming (EP) is a probabilistic, global search technique that starts with a population of randomly generated candidate solutions and evolves towards better solutions over a number of generations or iterations.

Optimization by EP can be summarized into two major steps:

1. Mutate the solutions in the current population

2. Select the next generation from the mutated and the current solutions.

These two steps can be regarded as a population-based version of the classical generate-and test method, where mutation is used to generate new solutions (offspring) and selection is used to test which of the newly generated solutions should survive to the next generation.

Formulating EP as a special case of the generate-and-test method establishes a bridge between EP and other search algorithms, such as evolution strategies, genetic algorithms, simulated annealing (SA), tabu search (TS), and others, and thus facilitates cross-fertilization among different research areas. One disadvantage of EP in solving some of the multimodal optimization problems is its slow convergence to a good near-optimum.

Evolutionary programming (EP) is a stochastic optimization strategy, which places emphasis on the behavioural linkage between parents and their offspring. It is a powerful and general optimization method, which does not depend on the first and second derivatives of the objective function and the constraints of the problem.

Evolutionary programming is a probabilistic search technique, which generates the initial parent vectors distributed uniformly in intervals within the limits and obtains global optimum solution over number of iterations. The main stages of this technique are initialization, creation of off – spring vectors by mutation and competition and selection of best vectors to evaluate best fitness solution. As the history of the field suggests there are many different variants of Evolutionary Algorithms. The common underlying idea behind all these techniques is the same: given a population of individuals the environmental pressure causes natural selection (survival of the fittest) and this causes a rise in the fitness of the population.

Given a quality function to be maximized we can randomly create a set of candidate solutions, i.e., elements of the function's domain, and apply the quality function as an abstract fitness measure – the higher the better. Based on this fitness, some of the better candidates are chosen to seed the next generation by applying recombination and/or mutation to them. Recombination is an operator applied to two or more selected candidates (the so called parents) and results one or more new candidates (the children). Mutation is applied to one candidate and results in one new candidate.

Executing recombination and mutation leads to a set of new candidates (the offspring) that compete based on their fitness (and possibly age) – with the old ones for a place in the next generation. This process can be iterated until a candidate with sufficient quality (a solution) is found or a previously set computational limit is reached. In this process there are two fundamental forces that form the basis of evolutionary systems.

- Variation operator (mutation) create the necessary diversity and thereby facilitates novelty, while
- Selection acts as a force pushing quality.

The combined application of variation and selection generally leads to improving fitness values in consecutive populations. It is easy (although somewhat misleading) to see such a process as if the evolution is optimizing, or at least "approximating", by approaching optimal values closer and closer over its course. Alternatively, evolution it is often seen as a process of adaptation. From this perspective, the fitness is not seen as an objective function to be optimized, but as an expression of environmental requirements. Matching these requirements more closely implies an increased viability, reflected in a higher number of offspring. The evolutionary process makes the population adapts to the environment better and better.During selection fitter individuals have a higher chance to be selected than less fit ones, but typically even the weak individuals have a chance to become a parent or to survive.

For mutation, the pieces that will be mutated within a candidate solution and the new pieces replacing them are chosen randomly. The evaluation (fitness) function represents a heuristic estimation of solution quality and the variation and the selection operators drive the search process. The most important advantage of EP is that it uses only the objective function information and hence independent of the nature of the search space such as smoothness, convexity or uni-modality, etc.

EP differs from traditional GAs in two aspects:

- EP uses the control parameters (real values), but not their coding as in traditional GAs
- EP relies primarily on mutation and selection, but not crossover, as in traditional GAs. Hence, considerable computation time may be saved in EP.

3.3 Difference between Evolutionary Programming and Classical Method

- EP has capability of converging into a global optimum. Classical methods are not guaranteed to converge to the global optimum of the general non-convex optimum problem.
- Accuracy in case of EP is less as compare to classical method.
- The search process in EP uses a population of points rather than a single point.
- Classical methods suffer from the difficulty in handling inequality constraints. Whereas EP can handle constraints easily.
- EP requires only the evaluation of the fitness function, which is formed from objective function so that every solution could be given a quality value.
- EP operates on the encoded string of the problem parameters rather than the actual parameters of the problem.

3.4 Applications of Evolutionary Programming

Evolutionary programming is a probabilistic search technique, which generates the initial parent vectors distributed uniformly in intervals within the limits and obtains global optimum solution over number of iterations. Evolutionary Programming is search algorithms based on the mechanics of natural selection. The most important advantage of EP is that it uses only the objective function information and hence independent of the nature of the search space such as smoothness, etc. EP based algorithms are increasingly applied for solving power system optimization problems in recent years.

3.5 Components of Evolutionary Programming

Evolutionary Programming requires certain steps, components, procedures or operators that must be specified. The main components are:

1. Initialization
2. Representation
3. Fitness function
4. Population
5. Parent selection methodology
6. Mutation
7. Survivor selection mechanism (replacement)
8. Termination condition

To obtain a running algorithm the initialization procedure and a termination condition must be also defined. Schematic diagram of EP algorithm is shown in figure 3.1.

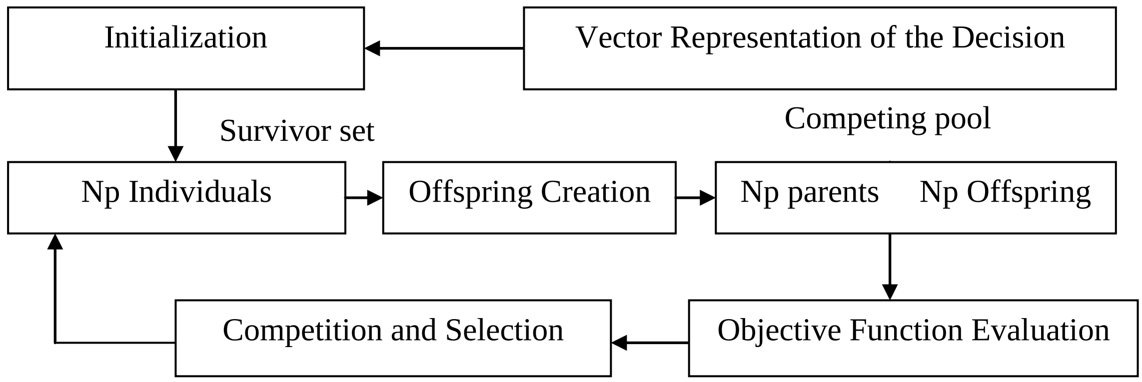


Figure 3.1 Schematic diagram of EP algorithm

3.5.1 Initialization

Initialization is the first step which is applied while using Evolutionary Programming. Here, problem specific heuristics can be used in this step aiming at an initial population with higher fitness. Thus, population is selected based on certain measures. Whether this is worth the extra computational effort or not is very much depending on the application at hard[22]. Here, an initial population of parent individuals (X_i, i = 1, 2, 3 ……k) is generated randomly within a feasible range in each dimension.

3.5.2 Representation

This is the first design step. It amounts to specifying a mapping from the phenotypes onto a set of genotypes that are said to represent these phenotypes which means to link the" real world" to the "EA world", that is to set up a bridge between the original problem context and the problem solving space where evolution will take place. Objects forming possible solutions within the original problem context are referred to as phenotypes, their encoding, the individuals within the EA, are called genotypes. For instance, given an optimization problem on integers, the given set of integers would form the set of phenotypes. Then one could decide to represent them by their binary code, hence 18 would be seen as a phenotype and 10010 as a genotype representing it. It is necessary to understand that the phenotype space can be very different from the genotype space, and that the whole evolutionary search takes place in the genotype space. A solution of a good phenotype is obtained by decoding the best genotype after termination.

The inverse or reverse mapping from genotypes to phenotypes is usually called decoding and it is required that the representation be invertible: to each genotype there has to be at most one corresponding phenotype.

3.5.3 Fitness Function

Evaluation function represents the task to solve in the evolutionary context. Technically, it is a function or procedure that assigns a quality measure to genotypes. The role of the evaluation function is to represent the requirements to adapt to. It forms the basis for selection, and thereby it facilitates improvements.

3.5.4 Population

Population means how many individuals have in it i.e. setting the population size. As opposed to variation operator, that act on one or two parent individuals, the selection operators (the parent selection and survival selection) work at population level. In general, the take the whole population into account and choices are always made relative to what we have. For instance, the best individual of the given population is chosen to be replaced by a new one. In almost all EPs applications, the population size is constant, not changing during the evolutionary search. The objective of population is to hold (the representation of) possible solutions. A population is a multi set of genotypes and each individual forms the unit of evolution but does not change itself.

The diversity of a population is a measure of the number of different solutions present. No single measure for diversity exists; typically people might refer to the number of different fitness values present, the number of phenotypes present, or the number of different genotypes.

3.5.5 Parent Selection Methodology

Chromosomes are selected from the population to be parents. The problem is how to select these chromosomes. According to Darwin's evolution theory the best ones should survive and create new offspring. The objective of parent selection or mating selection is to compare among the individuals based on their quality, and to allow the better individuals to become parents of the next generation. An individual is a parent if it has been selected to undergo variation in order to create offspring. Together with the survivor selection mechanism, parent selection is responsible for pushing quality improvements. Thus, high quality individuals get a higher chance to become parents than those with low quality. However, low quality individuals are often given a small, but positive chance; otherwise the whole search would become too greedy and get stuck in a local optimum.

There are many methods for selecting the best chromosomes:

1. Steady-State Selection

2. Roulette Wheel Selection

3. Rank Selection

4. Elitism

3.5.5.1 Steady-State Selection:

This is not a particular method of selecting parents. The main idea of this selection is that big part of chromosomes should survive to the next generation. EP then works in a following way. In every generations a few (good with high fitness) chromosomes are selected for creating a new offspring. Then some (bad with low fitness) chromosomes are removed and then, the new offspring is placed in their place. The rest of population survives to the new generation.

3.5.5.2 Roulette Wheel Selection:

Parents are selected according to their fitness. The better the chromosomes are, the more chances to be selected they have. Imagine a roulette wheel where are placed all chromosomes in the population, everyone has its place big accordingly to its fitness function. Then a marble is thrown there and selects the chromosome. Chromosome with bigger fitness will be selected more times.

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

$$f_{av} = \frac{\sum_{i=1}^{n} f_i}{n} \tag{3.1}$$

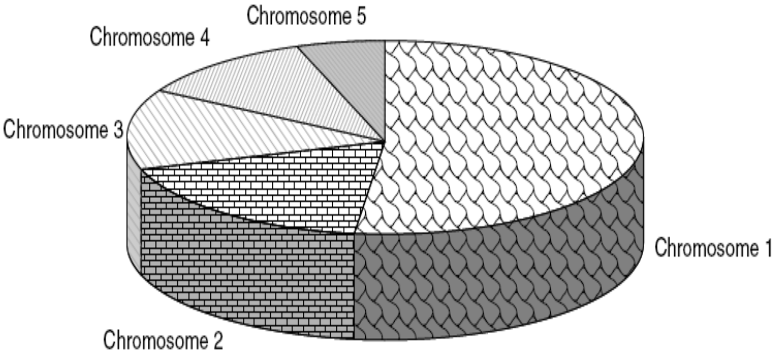


Figure 3.2 Roulette Wheel selection

Since, the first individual has a higher fitness value than any other, it is expected that the RW selection will choose the third individual more than any other individual. This roulette-wheel selection scheme can be simulated easily. Using the fitness value Fi of all the strings, the probability of selecting a string pi can be calculated. Thereafter the cumulative probability (Pi) of each string being copied can be calculated by adding the individual probabilities from the top of the list. Thus, the bottom-most string in the population should have a cumulative probability (P_n) equal to one.

3.5.5.3 Rank Selection

In the roulette wheel selection, the problem is when the fitnesses differ very much. For example, if the best chromosome fitness is 90% of the entire roulette wheel then the other chromosomes will have very few chances to be selected. But, Rank selection first ranks the population and then every chromosome receives fitness from this ranking. The worst will have fitness 1, second worst will have 2 etc. and the best will have fitness N (number of chromosomes in population). After this all the chromosomes have a chance to be selected. But this method can lead to slower convergence, because the best chromosomes do not differ so much from other ones.

3.5.5.4 Elitism

Elitisms are quite different. When creating a new population by crossover and mutation, there exists a high probability of losing the best chromosome. Elitism is the name of a method which first copies the best chromosome (or a few best chromosomes) to the new population.

3.5.6 Mutation

It is also known as background operator .It plays dominant role in the evolutionary process and consists of randomly selecting a mutation point. It is applied to one genotype and delivers a (slightly) modified mutant, the child or offspring of it. A mutation operator is always stochastic: its output- the child- depends on the outcome of a series of random choices. A problem specific heuristic operator acting on one individual could be termed as mutation.

3.5.7 Survivor Selection Mechanism

The role of survivor selection is to distinguish among individuals based on their quality. In that it is similar to parent selection, but it is used in different stage of evolutionary cycle. The survival selection mechanism is called after having created the offspring of the selected parents. A choice has to be made on which individuals will be allowed in the next generation. This decision is usually based on their fitness values, favouring those with higher quality. Survivor selection is also often called replacement selection or replacement strategy. In many cases, the two terms can be used interchangeably. The choice between the two is often arbitrary. The preference for using replacement can be motivated by the skewed proportion of the number of individuals in the population and the number of newly created children. In particular, if the number of children is quite small with respect to the population size e.g., 2 children and a population of 100. In this case, the survivor selection step is as simple as to choose the two old individuals that are to be deleted to make place for the new ones. In other words, it is more efficient to declare that everybody survives unless deleted and to choose whom to replace. If the proportion is not skewed like this, e.g., 500 children made from a population of 100, and then this is not an option, so using the term survivor selection is appropriate. Each individual in the competing pool is evaluated for its fitness. All individuals compete with each other for selection. The best K individuals with maximum fitness values are retained to be parents of the next generation. The process of creating off spring and selecting those with maximum fitness are repeated until there is no appreciable improvement in the maximum fitness value or it is repeated up to a pre specified number of iterations.

3.5.8 Termination Condition

Termination of the algorithm will be done due to either of the following condition. One if the problem has a known optimal fitness level, probably coming from a known optimum of the given objective function, then reaching this level should be used as a stopping condition or if the maximum number of iterations are reached.

3.6 Problem Formulation for Estimation of Fuel Cost Function

The EP based program is developed to find an estimate for the thermal unit input-output curve coefficients. These estimates are used to recalculate the fuel cost energy function ($F_{estimated}$) at each time step. Then, the error associated with each measurement is minimized. The problem formulation can be clearly stated as to find an estimate for the fuel cost parameters, for any model, that minimizes the error r_i .

$$r_i = F_{i(actual)} - F_{i(estimated)} \qquad (3.2)$$

$$where\ F_{i(estimated)} = \sum_{i=1}^{n} FC_i(P_i) \qquad (3.3)$$

$FC_i(P_i)$ is the fuel cost energy function of the ith unit and P_i is the power generated by the ith unit.

Subjected to power balance constraints:

$$P_D = \sum_{i=1}^{n} P_i - P_L \qquad Here\ \ P_L = 0 \qquad (3.4)$$

Where P_D is the system load demand.

The inequality constraint:

$$C_{imin} \le C_i \le C_{imax} \qquad i = 1,2, \dots n \qquad (3.5)$$

Where C_{imin} and C_{imax} $a$re the minimum and maximum limits of the fuel cost coefficients of the unit.

The fuel-cost function without valve-point loadings of the generating units is given by

$$F_i(P_{ti}) = \ a_{0i} + a_{1i}\,P_{ti} + r_i \qquad (3.6)$$

$$F_i(P_{ti}) = \ a_{0i} + a_{1i}\,P_{ti} + a_{2i}P_{ti}^2 + r_i \qquad (3.7)$$

$$F_i(P_{ti}) = \ a_{0i} + a_{1i}\,P_{ti} + a_{2i}P_{ti}^2 + a_{3i}P_{ti}^3 + r_i \qquad (3.8)$$

The fuel-cost function considering valve-point loadings of the generating units is given by

$$F_i(P_{ti}) = \ \left[a_{0i} + a_{1i}\,P_{ti} + a_{2i}P_{ti}^2 + \left|e_i sin\left(f_i\left(P_{i,min} - P_i\right)\right)\right|\,\right] + r_i \qquad (3.9)$$

Where $a_{0i}, a_{1i}\,, a_{2i}$ and are the fuel-cost coefficients of the unit, and e_i and f_i are the fuel cost-coefficients of the unit with valve-point effects.

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

3.6.1 Parameter Initialization

The key parameters that control the Evolutionary Programming are population size (L), boundary constraints of optimization variables (CG), scaling factor (β), empirical rate (τ), and the stopping criterion of maximum number of iterations (generations) $iter_{max}$. The set of fuel cost parameters of all generators are represented as the population. For a system with NG generators, the population is represented as a vector of length CG.

3.6.2 Initialization

The initial population comprises of only the candidate cost coefficient solutions, which satisfy all the constraints. It consists of C_i^t (i=1, 2 ….CG; j=1, 2…….L) trial parent individuals. The elements of a parent are the recombination of the cost coefficients of the generating units randomly chosen by a random number ranging over $[C_i^{min}, C_i^{max}]$.

$$C_i^j = C_i^{min} + rand()\left(C_i^{max} - C_i^{min}\right) \quad (\mathrm{i} = 1, 2 \; \ldots . \mathrm{CG}; \; \mathrm{j} = 1, 2 \ldots \ldots . \mathrm{L}) \tag{3.11}$$

where rand () is uniform random number ranging over [0, 1]

The elements of parent/offspring C_i^j may violate constraint. This violation is corrected by fixing them at lower or upper limits as described in Eq.

$$C_i^j = \begin{cases} C_i^{min} \; ; \left(C_i^j < C_i^{min}\right) \\ C_i^{max} \; ; \left(C_i^j > C_i^{min}\right) \\ C_i^j ; \; C_i^{max} \geq C_i^j \geq C_i^{min} \end{cases}$$

$$(\mathrm{i} = 1, 2 \; \ldots . \mathrm{CG}; \; \mathrm{j} = 1, 2 \ldots \ldots . \mathrm{L}) \tag{3.12}$$

3.6.3 Mutation

Mutation is often implemented by adding a random number or a vector from a certain distribution [e.g. a Gaussian distribution in the case of classical EP (CEP)] to a parent. The degree of variation of the Gaussian mutation is controlled by its standard deviation, which is also known as a 'strategy parameter" in evolutionary search. Initial solutions are usually far from the global optimum and hence larger step sizes may prove to be beneficial. But as the evolution progresses, later solutions may be nearer to the global point and the step size should be reduced gradually to enable quick convergence. Unfortunately, there is no way of predicting the transition point for the change of step size from a larger to smaller one. As no unique step size can give desired results for all types of functions.

The four alternative methods for creating offspring are explained below:

3.6.3.1 Classical Evolutionary Programming

Using Gaussian mutation, an offspring vector is created from each parent by adding to each component of, a Gaussian random variable with a zero mean and a standard deviation proportional to the scaled cost values of the parent trial solution.

$$C_i^{j+L} = C_i^j + \sigma_i^j \, N(0,1) \qquad (i = 1,2 \dots \dots CG; j = 1,2 \dots . . L) \qquad (3.13)$$

Where $N(0,1)$ represents a normally distributed random variable with zero mean and unit variance.

3.6.3.2 Fast Evolutionary Programming

Using Cauchy mutation (in FEP), an offspring is created by

$$C_i^{j+L} = C_i^j + \sigma_i^j \, Cr_i^j(0,1) \qquad (i = 1,2 \dots \dots CG; j = 1,2 \dots . . L) \qquad (3.14)$$

Where $Cr_i^j(0,1)$ is a Cauchy random variable with scale parameter centered at zero that is generated anew for each value of j.

3.6.3.3 Mean of the Gaussian and Cauchy mutations

Using the mean of the Gaussian and Cauchy mutations (in MFEP), an offspring is created in this method as

$$C_i^{j+L} = C_i^j + \frac{1}{2}\sigma_i^j (N(0,1) + \, Cr_i^j(0,1)) \qquad (i = 1,2 \dots \dots CG; I \neq d; j = 1,2 \dots . . L) \qquad (3.15)$$

Where N_j and Cr_j are Gaussian and Cauchy random variables, respectively, to be generated anew for each value of j.

3.6.3.4 Better of the Gaussian and Cauchy mutations

Using the method of choosing the better from two offspring generated from each parent, one by Gaussian mutation and the other by Cauchy mutation (in IFEP).

$$C_i^{j+L} = C_i^j + \sigma_i^j (N(0,1) \qquad (i = 1,2 \dots \dots CG; j = 1,2 \dots . . L)$$

$$C_i^{j+L} = C_i^j + \frac{1}{2}\sigma_i^j \, Cr_i^j(0,1) \qquad (i = 1,2 \dots \dots CG; j = 1,2 \dots . . L) \qquad (3.16)$$

The values of the objective function value of both offspring are evaluated, compared, and better individuals are chosen as parents for next generation.

In each method mentioned above, the standard deviation is given by the expression:

$$\sigma_i^j = \beta \frac{f^j}{f^{min}} \left(C_i^{max} - C_i^{min} \right) \qquad (3.17)$$

Where f^{min} is the minimum cost value among n the trial solutions.

β is the scaling factor.

$f_i = f(C_i)$ is the value of the objective function associated with the trial vector C_i .

Here σ is adapted based on its scaled values of the parent trial solution; if f_i is relatively low, the step size will be small and the offspring trial solution will be created near the current solution on average, and if f_i relatively high, the step size will be larger and next trial solution will be searched within a wider range on average thereby providing an adaptive effect based on the fitness (cost) value. Also, as f^{min} keeps updating toward a globally optimum point, adaptation will also be toward the global point.

The EP technique considers different kinds of mutations viz. Gaussian, Cauchy and combined Gaussian-Cauchy mutation. To create offspring the following scheme is adopted for self adaptation of strategy parameter based on empirical learning rates τ_1 *and* τ_2.

$$\sigma_i^{j+L} = \sigma_i^j \exp\{\tau_1 N(0,1) + \tau_2 N^j(0,1)\} \tag{3.18}$$

Where

$$\tau_1 = \left(\sqrt{(2 * (CG - 1))}\right)^{-1}$$

$$\tau_2 = \left(\sqrt{\left(2\sqrt{(CG - 1)}\right)}\right)^{-1}$$

3.6.4 Competition and Selection

Selection is the procedure whereby better offspring are produced. To decide whether the vector C_i^{t+1} should be a member of the population comprising the next generation, it is compared with the corresponding vector C_i^t. Thus, if f denotes the error under minimization, then

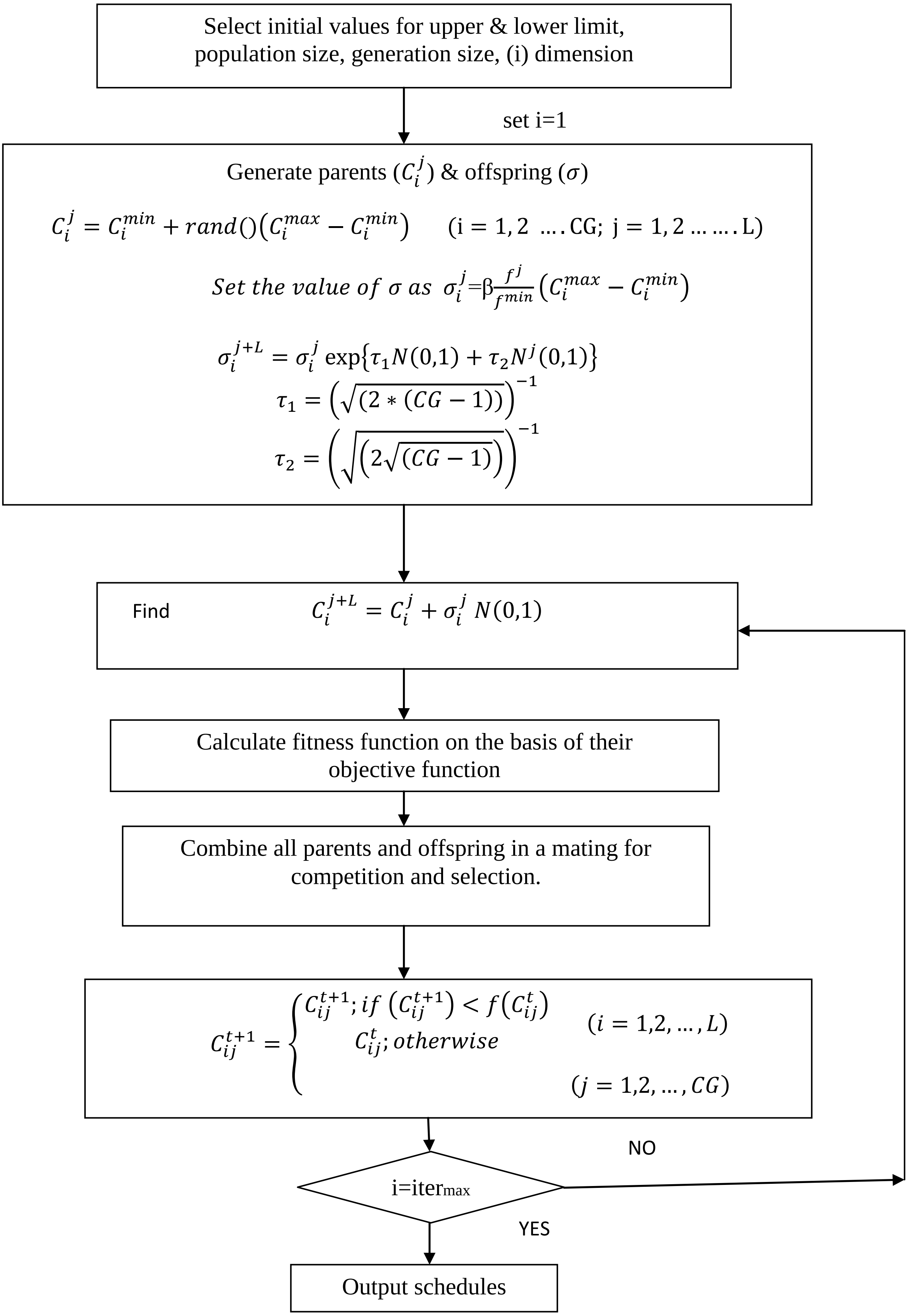
$$C_i^{j(t+1)} = \begin{cases} C_i^{j(t+1)};\ if\ \left(C_i^{j(t+1)}\right) < f\left(C_i^{j(t)}\right) & (i = 1,2,\dots,L) \\ C_i^{j(t)};\ otherwise & \\ & (j = 1,2,\dots,CG) \end{cases} \tag{3.19}$$

In this case, the cost of each of trial vector $C_i^{j(t+1)}$ is compared with that of its parent target vector $C_i^{j(t)}$. If the error f, of the target vector $C_i^{j(t)}$, is lower than that of the trial vector, the target is allowed to advance to the next generation. Otherwise, a trial vector replaces the target vector in the next generation.

3.6.5 Verification of the stopping criterion

Set the generation number for t=t+1. Repeat mutation, competition and selection operation until a stopping criterion is met, usually a maximum number of iterations (generations), $t_{\max}$. The stopping criterion depends on the type of problem.

3.7 Flowchart of Classical Evolutionary Programming



CHAPTER 4

RESULTS AND DISCUSSIONS

Evolutionary Programming method is tested using the thermal data. The field data are used to estimate the parameters of the smooth and non-smooth models. These measurements are for three thermal plants using different fuels (coal, oil, and gas). The main objective is to minimize the error obtained from the difference between the actual fuel cost energy given and the estimated fuel cost energy obtained from these fuel cost models. The results of Estimating Fuel-cost function Parameters after the implementation of EP are discussed. The algorithms are implemented in MATLAB (10) to solve the problem. The performance evaluated using the first, second and a third order model is referred to as Test Problem I, II and III respectively. The performance of the Non Smooth model with valve point loading is referred to as Test Problem IV.

Test Problem I: First order model

Test Problem II: Second order model

Test Problem III: Third order model

Test Problem IV: Non Smooth model with valve point loading

4.1. Test Problem I- First order model

In this test, data for the three units are used to estimate the first-order model parameters. Recalculated fuel-cost energy function along with the data is calculated using evolutionary programming and the comparison of absolute error in the estimation process is presented. The same parameters are estimated using the Least Error Square estimation, and outcomes from both estimation methods along with the Particle Swarm Optimization are presented and compared. The thermal generation data for the smooth models is given in table 4.1. [20]

Table-4.1 Thermal generation data

Output Power P_G (MW)	**Fuel Cost energy for Unit 1 Coal F_S (GJ/hr)**	**Fuel Cost energy for Unit 2 Oil F_S (GJ/hr)**	**Fuel Cost energy for Unit 3 Gas F_S (GJ/hr)**
10.0	176.62	184.75	187.87
20.0	256.40	268.20	272.80
30.0	361.50	377.70	384.30

40.0	467.60	488.80	497.20
50.0	579.50	606.00	616.50
$\sum$ Fs	**1841.62**	**1925.45**	**1958.00**

4.1.1 Optimum solution for Test Problem I using Least Error Square

Developed program gives the estimated fuel cost coefficients, estimated fuel cost energy and the absolute error for three units. The results are shown in the table 4.2(a).

Table-4.2(a) Estimated Coefficients using Least Error Square Method

Unit	Fuel cost Coefficients	Fuel cost Coefficients by LES
Unit-1 Coal	a_0 a_1	10.1696 63.2360
Unit-2 Oil	a_0 a_1	10.6470 65.8400
Unit-3 Gas	a_0 a_1	10.8300 66.7000

4.1.2 Problem Formulation by Evolutionary Programming

The EP based program is developed to find an estimate for the unknowns (X) in Eq. (2.7). These estimates are used to recalculate the fuel cost energy function ($F_{estimated}$) at each time step. Then, the error associated with each measurement is minimized. The problem formulation can be clearly stated as to find an estimate for the fuel cost coefficients, for any model, that minimizes the error r_i .

$$Min\ r_i = F_{i(actual)} - F_{i(estimated)} \tag{4.1}$$

$$where\ F_{i(estimated)} = \sum_{i=1}^{n} FC_i(P_i) \tag{4.2}$$

$FC_i(P_i)$ is the fuel cost energy function of the i^{jth} unit and P_i is the power generated by the i^{jth} unit.

Subjected to power balance constraints:

$$P_D = \sum_{i=1}^{n} P_i - P_L \qquad Here\ \ P_L = 0 \tag{4.3}$$

Where P_D is the system load demand

And the inequality constraint

$$C_{imin} \le C_i \le C_{imax} \qquad i = 1,2, \dots n \tag{4.4}$$

where C_{imin} and C_{imax} are the minimum and maximum limits of the fuel cost coefficients of the unit.

The fuel-cost function without valve-point loadings of the generating units is given by

$$Case\ I - \ F_i(P_{ti}) = a_{0i} + a_{1i}\,P_{ti} + r_i \tag{4.5}$$

$$Case\ II - \ F_i(P_{ti}) = a_{0i} + a_{1i}\,P_{ti} + a_{2i}P_{ti}^2 + r_i \tag{4.6}$$

$$Case\ III - \ F_i(P_{ti}) = a_{0i} + a_{1i}\,P_{ti} + a_{2i}P_{ti}^2 + a_{3i}P_{ti}^3 + r_i \tag{4.7}$$

The fuel-cost function considering valve-point loadings of the generating units is given by

$$Case\ IV - \ F_i(P_{ti}) = \left[a_{0i} + a_{1i}\,P_{ti} + a_{2i}P_{ti}^2 + \left|e_i sin\left(f_i\left(P_{i,min} - P_i\right)\right)\right|\right] + r_i \tag{4.8}$$

where $a_{0i}, a_{1i}\,, a_{2i}$ and are the fuel-cost coefficients of the unit, and e_i and f_i are the fuel cost-coefficients of the unit with valve-point effects.

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

4.1.2.1 Algorithm for estimating the Fuel Cost Function

1. Read the thermal generation data,actual fuel cost energy $F_{i(actual)}$, Power output, Population size (L)=10, boundary constraints of optimization variables (NG)=5,scaling factor (β)=0.01,imperical rate (τ), iter=50 and the $C_i^{max}\ and\ C_i^{min}$ for (i=1,2,…CG) are the limits of the fuel cost function coefficients.
2. Generate an array of (L*CG) size of uniform random numbers.
3. Set population counter, for i=0 and increment population counter, i=i+1.
4. Set generation counter, for j=0 and increment generation counter, j=j+1.
5. Calculate the fuel cost function parameters for each unit using equation

$$C_i^j = C_i^{min} + rand()\left(C_i^{max} - C_i^{min}\right) \qquad (\mathrm{i} = 1, 2\ \ldots.\mathrm{CG};\ \mathrm{j} = 1, 2 \ldots \ldots.\mathrm{L}) \tag{4.9}$$

6. Check the limit of the cost coefficient of individual generator i.e.

$$C_i^{max} \geq \ C_i^j \geq C_i^{min} \tag{4.10}$$

7. If limit is violated than upgrade the cost coefficient of the unit using equation

$$C_i^j = \begin{cases} C_i^{min} \ ;\ \left(C_i^j < C_i^{min}\right) \\ C_i^{max} \ ;\ \left(C_i^j > C_i^{min}\right) \end{cases}$$
$$(\mathrm{i} = 1, 2\ \ldots.\mathrm{CG};\ \mathrm{j} = 1, 2 \ldots \ldots.\mathrm{L}) \tag{4.11}$$

8. Calculate estimated cost function using equation

$$F_i(P_{ti}) = a_{0i} + a_{1i}\,P_{ti} + r_i \tag{4.12}$$

$$F_i(P_{ti}) = a_{0i} + a_{1i}\,P_{ti} + a_{2i}P_{ti}^2 + r_i \tag{4.13}$$

$$F_i(P_{ti}) = a_{0i} + a_{1i}\,P_{ti} + a_{2i}P_{ti}^2 + a_{3i}P_{ti}^3 + r_i \tag{4.14}$$

The fuel-cost function considering valve-point loadings of the generating units is given by

$$F_i(P_{ti}) = \left[a_{0i} + a_{1i}\, P_{ti} + a_{2i}P_{ti}^2 + \left|e_i sin\left(f_i\left(P_{i,min} - P_i\right)\right)\right|\right] + r_i \tag{4.15}$$

9. The error associated with each measurement can be calculated as

$$r_i = F_{i(actual)} - F_{i(estimated)} \tag{4.16}$$

10. End of the evaluation operation

11. Start of the mutation operation

12. Calculate the Standard deviation σ_i^j of the cost coefficients obtained from the evaluation process by the formula:

$$\sigma_i^j = \beta \frac{f^j}{f^{min}} \left(C_i^{max} - C_i^{min}\right) \tag{4.17}$$

13. Now calculate the Standard deviation of the next L populations from the standard deviation of the first L populations by:

$$\sigma_i^{j+L} = \sigma_i^j \exp\{\tau_1 N(0,1) + \tau_2 N^j(0,1)\} \tag{4.18}$$

Where $$\tau_1 = \left(\sqrt{(2 * (CG - 1))}\right)^{-1}$$

$$\tau_2 = \left(\sqrt{\left(2\sqrt{(CG - 1)}\right)}\right)^{-1}$$

14. Repeat the steps 5-7

15. Calculate the fuel cost function coefficients using equation

$$C_i^{j+L} = C_i^j + \sigma_i^j\, N(0,1) \qquad (i = 1,2 \ldots \ldots CG; I \neq d; j = 1,2 \ldots \ldots L) \tag{4.19}$$

16. Repeat steps 8-10

17. End of the mutation operation

18. Start of the selection process

19. Set population counter, for j=0 and increment population counter, j=j+1

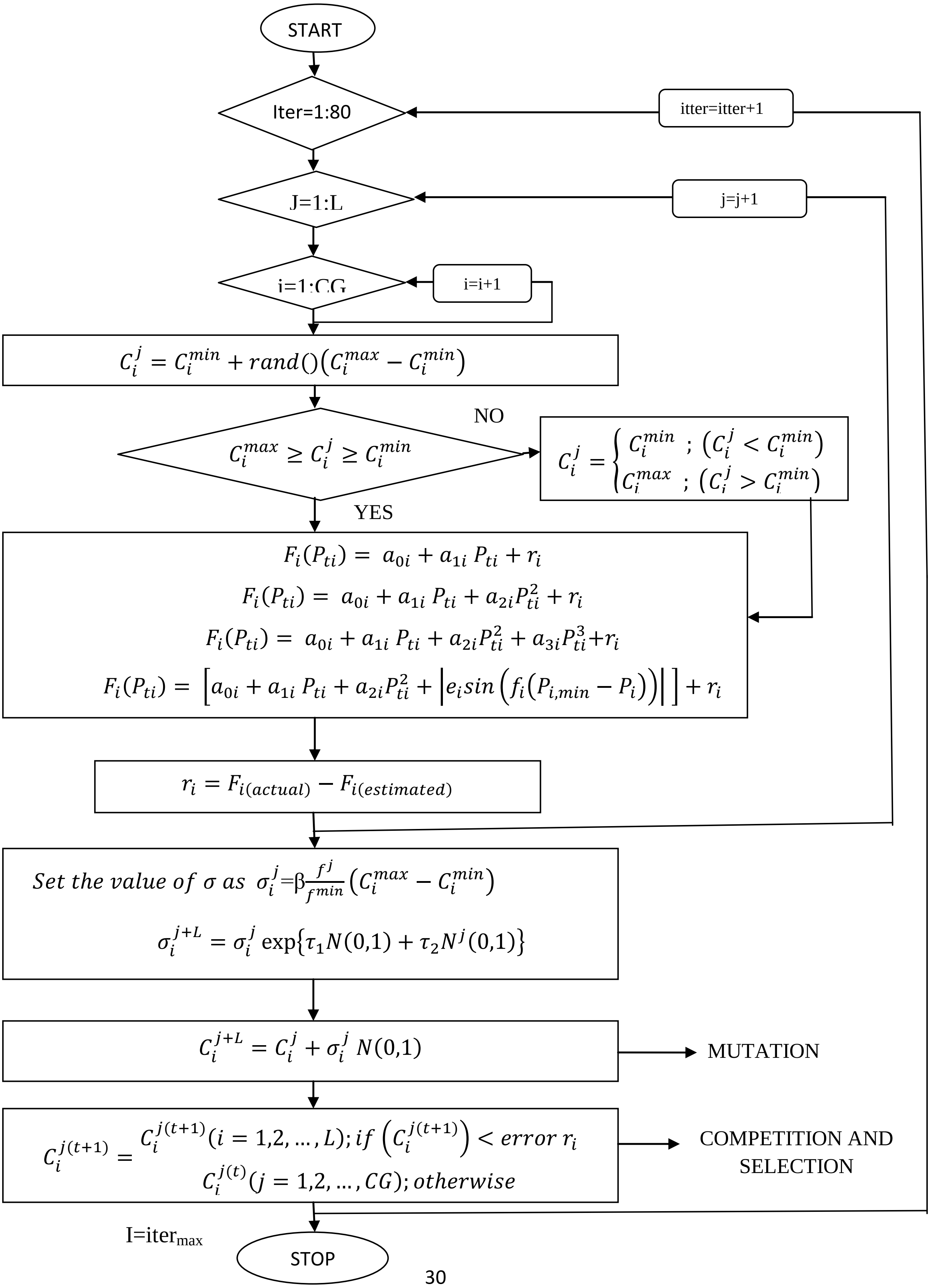
$$C_i^{j(t+1)} = \begin{cases} C_i^{j(t+1)}; if\ \left(C_i^{j(t+1)}\right) < f\left(C_i^{j(t)}\right) \\ C_i^{j(t)}; otherwise \end{cases} \qquad \begin{matrix}(i = 1,2, \ldots, L) \\ (j = 1,2, \ldots, CG)\end{matrix} \tag{4.20}$$

20. Select the least error among no of populations.

21. Repeat steps for increment each iteration, steps 3-21.

22. Stop

4.1.2.2 Flowchart for estimating the Fuel Cost function using Evolutionary Programming



4.1.3 Optimum solution for Test Problem I using Evolutionary Programming

The description of the results obtained by utilizing Evolutionary Programming is considered is detail. The population size has been assumed as 20. Therefore, population we have taken is 20 x 4. Results of the first 20 iterations for unit 1 are shown below in table 4.3(a). Best solutions after 50 iterations with different population size and different scaling factor for Unit 1 are given in table 4.3(b).The simulated results for Test problem I are shown in table 4.3(c). Actual fuel cost energy is 1841.62 GH/hr [1]

Table-4.3(a) Results of first 20 iterations

Cost parameter (a_1)	**Cost parameter (a_0)**	**Estimated Fuel cost energy $F_{estimated}$ (GJ/hr)**	**Error (r_i)**
1.0e+003 *			
0.0101	0.0103	0.0101	0.0102
0.0651	0.0615	0.0655	0.0614
1.8448	1.8453	1.8494	1.8361
0.0032	0.0037	0.0077	0.0055
0.0102	0.0103	0.0102	0.0103
0.0618	0.0610	0.0618	0.0625
1.8423	1.8538	1.8409	1.8504
0.0007	0.0121	0.0007	0.0087
0.0101	0.0101	0.0101	0.0102
0.0653	0.0641	0.0651	0.0636
1.8425	1.8310	1.8408	1.8470
0.0009	0.0106	0.0008	0.0053
0.0103	0.0102	0.0102	0.0101
0.0603	0.0619	0.0636	0.0653
1.8442	1.8466	1.8444	1.8438
0.0026	0.0049	0.0028	0.0022
0.0101	0.0102	0.0101	0.0101
0.0660	0.0650	0.0646	0.0657
1.8454	1.8508	1.8448	1.8465
0.0038	0.0092	0.0032	0.0049

Table-4.3(b) Best Solutions

Population size L	**Scaling factor β**	**Fuel Cost Coefficient a_1**	**Fuel Cost Coefficient a_0**	**Estimated Fuel cost energy (GJ/hr) $F_{estimated}$**	**Error (r_i)**
10	0.01	10.2682	60.2731	1841.6	0.0246
20	0.01	10.1474	63.9045	1841.6	0.0129
30	0.01	10.0719	66.1727	1841.6	0.0120

Table-4.3(c) Simulation results for Test Problem I

		Coefficients based on			Error		
Unit	**Fuel Cost coefficients**	**PSO[1]**	**LES**	**EP**	**PSO[1]**	**LES**	**EP**
Unit 1	a_0 a_1	**60.0060** **10.1900**	**63.2360** **10.1696**	**63.9045** **10.2682**	**36.322**	**38.956**	**18.1305**
Unit 2	a_0 a_1	**66.0010** **10.5700**	**65.8400** **10.6470**	**69.1320** **10.5322**	**39.151**	**41.140**	**0.04**
Unit 3	a_0 a_1	**66.0020** **10.7800**	**66.7000** **10.8300**	**65.3015** **10.8767**	**38.801**	**41.000**	**0.0155**

It can be observed from table 4.3(c) that the recalculated fuel cost energy function values (festimated) using the Evolutionary Programming (EP) methods are very close to the actual values with very less value of absolute error in comparison to the other two methods namely PSO and LES.The evolutionary programming convergence characteristics are presented in the form of a figure as shown below.

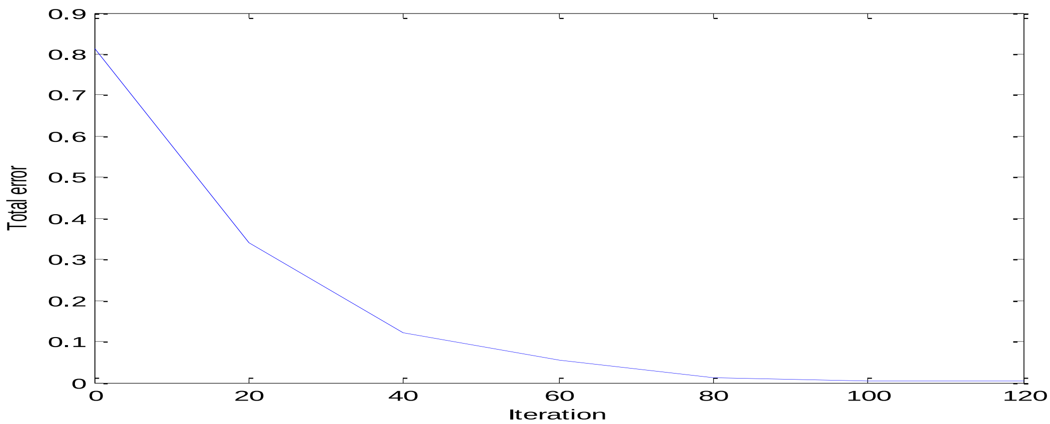


Figure-4.2 EP convergence characteristics for unit 1 (linear model)

The convergence pattern of EP is shown in Figure 3.1 for a sample case. It is evident that the proposed approach converged to a good solution within the first 80 iterations before it gradually settled to the final results.

4.2. Test Problem II- Second order model

In this test, the same data is used to estimate the parameters for the second-order model.

$$F_i(P_{ti}) = a_{0i} + a_{1i} P_{ti} + a_{2i} P_{ti}^2 + r_i$$

4.2.1 Optimum solution using Least Error Square

Developed program gives the estimated fuel cost coefficients, estimated fuel cost energy and the absolute error for three units. The results are shown in the table 4.4(a).

Table-4.4(a) Estimated Coefficients using Least Error Square Method

Unit	Fuel cost Coefficients	Fuel cost Coefficients by LES
Unit 1 Coal	a_0 a_1 a_2	0.0466 7.3736 95.8560
Unit 2 Oil	a_0 a_1 a_2	0.0505 7.6170 101.1900
Unit 3 Gas	a_0 a_1 a_2	0.0491 7.8814 101.1000

4.2.2 Optimum solution using Evolutionary Programming

The simulated results for Test Problem II are shown in table 4.4(b).[1]

Table-4.4(b) Simulation results for Test Problem II

	Coefficients based on				Error		
Unit	**Cost coefficients**	**PSO**	**LES**	**EP**	**PSO**	**LES**	**EP**
Unit 1	a_0 a_1 a_2	**96.279** **7.592** **0.042**	**95.856** **7.373** **0.046**	**106.0951** **7.6410** **0.0512**	**10.117**	**14.2080**	**15.87**
Unit 2	a_0 a_1 a_2	**101.000** **7.800** **0.046**	**101.190** **7.617** **0.050**	**101.2339** **7.5419** **0.0542**	**11.850**	**14.0400**	**7.7**

Unit 3	**a_0** **a_1** **a_2**	**102.000** **7.900** **0.048**	**101.100** **7.881** **0.049**	**105.8016** **7.7076** **0.0496**	**12.741**	**14.2286**	**1.2**

Table 4.4(b) shows that for the three units, estimates are close to the actual values as compared with the first order model by using LES and PSO methods. The absolute error for both the PSO and LES methods are lower than those obtained with the linear model. This indicates that this model is better suited for such data presentation. But in the case of EP the fuel cost estimation is too close to the actual value irrespective of the model used. And the absolute error obtained by using EP is very small as compared to PSO and LES methods.

The evolutionary programming convergence characteristics for unit 1 of second order model are presented in the form of a figure as shown below.

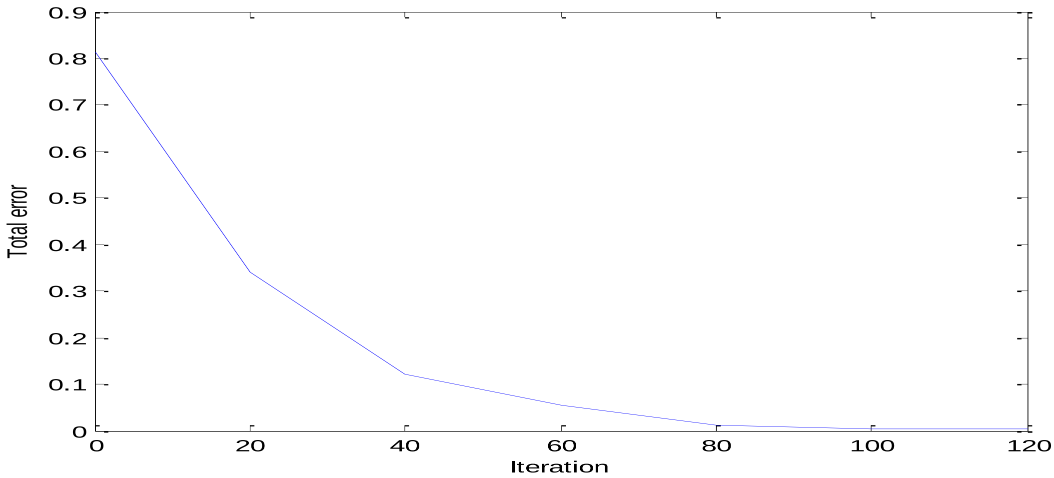


Figure-4.3 EP convergence characteristics for unit 1 (second order model)

The convergence pattern of EP is shown in Figure 3.2 for a sample case. It is evident that the proposed approach converged to a good solution within the first 80 iterations before it gradually settled to the final results.

4.3. Test Problem III- Third order model

The coefficients of the third-order model are estimated using the same set of thermal generation data. The third order model equation is given by the following equation.

$$F_i(P_{ti}) = a_{0i} + a_{1i} P_{ti} + a_{2i} P_{ti}^2 + a_{3i} P_{ti}^3 + r_i$$

4.3.1 Optimum solution using Least Error Square

Developed program gives the estimated fuel cost coefficients, estimated fuel cost energy and the absolute error for three units. The results are shown in the table 4.5(a).

Table-4.5(a) Estimated Coefficients using Least Error Square Method

Unit	Fuel cost Coefficients	Fuel cost Coefficients by LES
Unit 1 Coal	a_0 a_1 a_2 a_3	-0.0016 0.1930 3.5347 123.1840
Unit 2 Oil	a_0 a_1 a_2 a_3	-0.0016 0.1941 3.8508 128.0000
Unit 3 Gas	a_0 a_1 a_2 a_3	-0.0016 0.1954 4.0464 128.4000

4.3.2 Optimum solution using Evolutionary Programming

The simulated results for Test Case III are shown in table 4.5(c).

Table-4.5(b) Simulation results for Test Problem III

	Coefficients based on				Error		
Unit	**Cost coefficients**	**PSO[1]**	**LES**	**EP**	**PSO[1]**	**LES**	**EP**
Unit 1	a_0 a_1 a_2 a_3	**120.241 3.979 2.0184 -0.002**	**123.1840 3.5347 0.1930 -0.0016**	**117.8033 3.6747 0.1938 -0.0016**	**8.8641**	**6.6560**	**4.5**
Unit 2	a_0 a_1 a_2 a_3	**130.278 3.542 0.200 -0.002**	**128.000 3.8508 0.1941 -0.0016**	**133.2364 3.6946 0.1944 -0.0016**	**5.547**	**6.8000**	**0.7**
Unit 3	a_0 a_1 a_2 a_3	**128.376 4.146 0.188 -0.002**	**128.4000 4.0464 0.1954 -0.0016**	**123.6271 4.1036 0.1943 -0.0015**	**5.799**	**6.7429**	**4.7**

The errors for the three units are much lower than those reported in cases I and II by using LES and PSO approach. This means that the third-order model fits the data more precisely than the previous models. Therefore, this is the most appropriate model that describes the data. Visual inspection of the data could help in choosing the best suitable model. Indeed, error analysis would help more in discussing the model adequacy. Industry practice shows that the third-order model is usually the best model that fits the thermal unit's input–output characteristics for modelling generating unit fuel-cost function [21]. In this case, there is no need for further investigations, since the resultant error is acceptable.

Since our selection criterion is based on the overall error of the estimation process, we can conclude that the EP is better than the LES and PSO method, regardless of the estimation model. The evolutionary programming convergence characteristics for unit 1 of third order model are presented in the form of a figure as shown below.

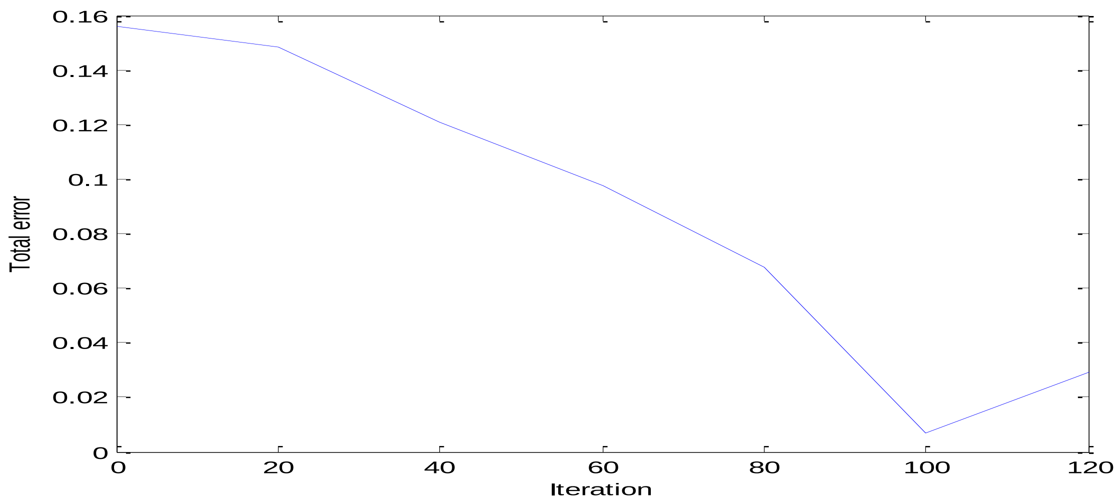


Figure-4.4 EP convergence characteristics for unit 1 (third order model)

The convergence pattern of EP is shown in Figure 3.3 for a sample case. It is evident that the proposed approach converged to a good solution within the first 100 iterations and after that the error started increasing. It means that the global solution is found at the 100^{th} iteration for the third order model.

4.4. Test Problem IV- Non-Smooth model with valve point loading

In this study, the EP approach is used to estimate fuel-cost function coefficients while considering the valve loading point effects. Two large thermal units are used in this study with their power output ranging from 0–500 MW for unit 1 and 0–360 MW for unit 2. This set of data covers a wide range of possible loading conditions, including all rippling effects, for the given generating units. The thermal generation data for valve point loading effects is given in table 4.6(a). The simulated results for Test Case IV are shown in table 4.6(b).[1,20]

Table-4.6(a) Thermal generation data

Unit 1		**Unit 2**	
Power output P (MW)	**Fuel cost energy F_{actual} (GJ/hr)**	**Power output P (MW)**	**Fuel cost energy F_{actual} (GJ/hr)**
0	550.00	0	309.00
25	982.94	18	592.18
50	1250.90	36	800.98
75	1307.25	54	901.36
100	1468.03	72	918.57
125	1849.96	90	1161.72
150	2028.98	108	1387.23
175	2023.333	126	1505.826
200	2378.296	144	1533.617
225	2686.609	162	1735.414
250	2779.917	180	1976.564
275	2858.341	198	2113.785
300	3269.109	216	2153.829
325	3490.738	234	2313.54
350	3512.636	252	2569.061
375	3785.882	270	2725.078
400	4131.982	288	2778.892
425	4265.053	306	2896.388
450	4264.307	324	3164.832
475	4698.816	342	3339.577
500	4962.688	360	3408.509
$\sum F_{actual}$	**58518.767**	$\sum F_{actual}$	**40285.952**

Table-4.6(b) Simulation results for Test Problem IV

Unit	Coefficients based on			Error	
	Cost coefficients	**PSO**	**EP**	**PSO**	**EP**
Unit 1	**a_0 a_1 a_2 e f**	**548.92129 8.09575 0.00029 301.42437 0.03500**	**559.0141 8.0925 0.000282 312.7233 0.0339**	**11.405**	**0.8523**
Unit 2	**a_0 a_1 a_2 e f**	**308.41028 8.10701 0.00053 200.61149 0.04201**	**321.0258 7.9504 0.00055083 201.6577 0.0404**	**8.618**	**0.4450**

Table 4.6(b) shows the results obtained in estimating the coefficients of the non-smooth fuel-cost functions for both units. Results clearly indicate that the proposed approach is able to accurately capture the impact of the valve point loadings on the fuel-cost function. The total estimated error is quite small for both units, which validates the accuracy of the proposed estimation tool. It is very important to notice that application of LES method is not simple or easy in this case. It is obvious that the sine term needs some mathematical manipulations and approximations in order to obtain the mathematical model that suits the LES method. On the other side, application of the EP method does not need any mathematical manipulation. This more precise mathematical modelling of the input–output curve poses a challenge to the LES and PSO method.EP convergence characteristics are shown below in figure 3.4.

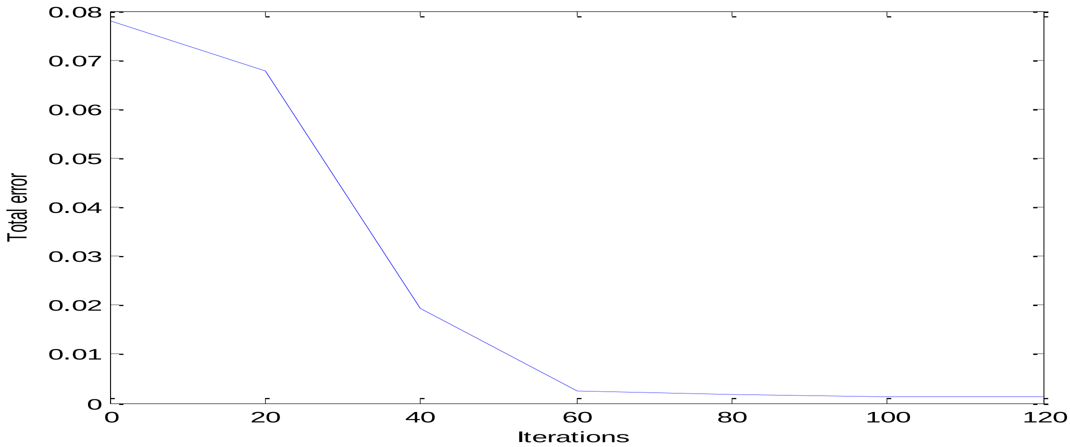


Figure-4.5 EP convergence characteristics for unit 1 (non-smooth model)

The convergence pattern of EP is shown in Figure 3.4 for a sample case. It is evident that the proposed approach converged to a good solution within the first 80 iterations before it gradually settled to the final results.

It can be concluded that Evolutionary programming can be used as a very accurate tool for estimating the input–output curve parameters as the results obtained are very accurate in comparison with Least Error Square method and Particle Swarm Optimization. Since our selection criterion is based on the overall error of the estimation process, we also conclude that the EP is better than the LES and PSO method, regardless of the estimation model.

CHAPTER **5**

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

An evolutionary programming based approach has been proposed and demonstrated to estimate the parameters of the input–output curve of thermal units. The formulated estimation problem is presented in state-space form. The objective is to minimize the total estimation error. Therefore, the problem is handled as an optimization problem rather than an estimation one. Numerical results show that highly near optimal solutions can be obtained by EP. The effectiveness of the developed program is tested successfully using different mathematical models with smooth and non-smooth characteristics. Different fuel function data are used to test the robustness of the developed approach. Convergence to an optimal or near-optimal solution is achieved with randomly chosen initial populations.

Since the encoding and decoding schemes are not needed in the proposed EP approach, a lot of computer memory and computing time can thus be saved. As only the objective function is relied on to guide the search towards the optimal solution in the EP-based algorithm, the generation models of thermal units can be of nonlinear and non smooth functions with prohibited areas. . The very accurate results obtained show that the proposed method can be used as a very accurate tool for estimating the input–output curve parameters.

5.2 Future Work

- An improved fast evolutionary programming (IFEP) technique that uses both Gaussian and Cauchy mutations for creation of off springs from the same parent and better ones are chosen for next generation may be used in place of EP.
- The problem can be extended to find the Economic Load Dispatch solution using the parameters obtained from the Evolutionary Programming method and compare with the solution of the same case using some other coefficients like obtained by PSO.

REFERENCES

[1] Alrashidi, M. R. , El-Naggar, K. M. and Al-Othman, A. K.(2009) , "Particle Swarm Optimization Based Approach for Estimating the Fuel-cost Function Parameters of Thermal Power Plants with Valve Loading Effects", Electric Power Components and Systems, 37: 11, pp.1219 – 1230.

[2] Chen, H. Y. K., and Postel, C. E., "On-line parameter identification of input-output curves for thermal units," IEEE Transactions on Power Systems, Vol. PWRS-1, No. 2, pp. 221–224, 1986.

[3] Schweppe, F. C., and Wildes, J., "Power system static state estimation," IEEE Transactions on Power Apparatus System, Vol. PAS89, pp. 120–130, 1970.

[4] Soliman, S. A., Emam, S. A., and Christensen, G. S., "Optimization of the optimal coefficients of non-monotonically increasing incremental cost curves," Electrical Power System , Vol. 21,pp. 99–106, 1991.

[5] Soliman, S. A., and Al-Kandari, A. M., "Kalman filtering algorithm for on-line parameter identification of input-output curves for thermal units," 8th Mediterranean Electro technical Conference, pp. 1588–1593, Bari, Italy, 13–16 May 1996.

[6] Duran-Paz, J. I., Perez-Hidalgo, F., and Duran-Martinez, M. J., "Bad data detection of unequal magnitudes in state estimation of power systems," IEEE Power Engineering Revolution, Vol. 1, pp. 57–60, 2002.

[7] Attaviriyanupa, P., Kita, H., Tanaka, E., and Hasegawa, J., "A hybrid EP and SQP for dynamic economic dispatch with non smooth fuel cost function," IEEE Transactions on Power System, Vol. 17, No. 2, pp. 411–416, 2002.

[8] Liu, D., and Cai, Y., "Taguchi method for solving the economic dispatch problem with non smooth fuel cost functions," IEEE Transactions on Power System, Vol. 20, No. 4, pp. 2006–2014, 2005.

[9] AlRashidi, M. R., and El-Hawary, M. E., "A survey of particle swarm optimization applications in electric power systems," IEEE Transactions on Evolutionary Computation, Vol. 13, No. 4, pp. 913–918, 2009.

[10] Chiang, Ch., "Improved genetic algorithm for power economic dispatch of units with valve point effects and multiple fuels," IEEE Transactions on Power System, Vol. 20, No. 4, pp. 1690–1699, 2005.

[11] AlRashidi, M. R., and El-Hawary, M. E., "Hybrid particle swarm optimization approach for solving the discrete OPF problem considering the valve loading effects," IEEE Transactions on Power System, Vol. 22, No. 4, pp. 2030–2038, 2007.

[12] Fink, L. H., Kwatny, H. G., and Mcdonald, J. P., "Economic dispatch of generation via valve-point loading," IEEE Transactions on Power Apparatus System, Vol. PAS-88, No. 6, pp. 805–811, 1969.

[13] Walters, D. C., and Sheble, G. B., "Genetic algorithm solution of economic dispatch with valve point loading," IEEE Transactions on Power System, Vol. 8, No. 3, pp. 1325–1332, 1993.

[14] Eberhart, R., and Kennedy, J., "A new optimizer using particle swarm theory," Proceedings of the Sixth International Symposium on Micro Machine and Human Science, pp. 39–43, Nagoya, Japan, 4–6 October 1995.

[15] Kennedy, J., and Eberhart, R., "Particle swarm optimization," IEEE International Conference on Neural Networks, Vol. 4, pp. 1942–1948, Perth, Australia, 1995.

[16] Hu, X., Shi, Y., and Eberhart, R., "Recent advances in particle swarm," Proceedings of 2004 Conference on Evolutionary Computation, Vol. 1, pp. 90-97, Portland, 2004.

[17] Eberhart, R. C., and Shi, Y., "Guest editorial special issue on particle swarm optimization," IEEE Transactions on Evolutionary Computation, Vol. 8, No. 3, pp. 201–203, 2004.

[18] Kennedy, J., and Eberhart, R. C., Swarm Intelligence, San Francisco: Morgan Kaufmann,2001.

[19] Shi, Y., and Eberhart, R., "A modified particle swarm optimizer," IEEE World Congress on Computational Intelligence, pp. 69–73, Anchorage, Alaska, 4–9 May 1998.

[20] El-Hawary, M. E., and Mansour, S. Y., "Performance evaluation of parameter estimation algorithms for economic operation of power systems," IEEE Transactions on Power Apparatus Syst., Vol. PAS-101, No. 3, pp. 574–582, 1982.

[21] Shoults, R. R., and Mead, M. M., "Optimal estimation of piece-wise linear incremental cost curves for EDC," IEEE Transactions on Power Apparatus System, Vol. PAS-103, No. 6, pp. 1432–1438, 1984.

[22] Jayabharti,Jayaprakash T., Jeyakumar D.N.,and Raghunathan T., "Evolutionary Programming Techniques for Different kinds of Economic dispatch Problems",Electric Power Systems Research,Vol.73(2),pp.169-176,August 2005.

[23] Periera Neto A., Unishuay C.,and Saveedra O.R., "Efficient Evolutionary strategy optimization procedure to solve the non convex economic dispatch problem with generator constraints",IEE Proceedings-Generation,Transmission and Distribution,Vol.153(5),pp.653-660,2005.

[24] Sinha, Chakrabarti N.,R.,and Chattopadhyay P.K., " Evolutionary Programming Techniques for Economic load Dispatch",IEEE Transactions on Evolutionary Computation,Vol.7(1),pp.83-94,2003.

[25] Yang H.T. and Yang P.C, Huang C.L., “Evolutionary Programming based Economic dispatch for units with non-smooth fuel cost functions”, IEEE Transactions on Power Systems, Vol.11 (1), pp.112-118, 1996.

[26] Yao,X.,Yong Liu and Guangming Liu, “Evolutionary Programming mode faster”, IEEE Transactions on Evolutionary Computation,Vol.3(2),pp.82-102,1999.

[27] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning. Reading, MA: Addison–Wesley, 1989.

[28] Mori H., and Horiguchi T., “A genetic algorithm based approach to economic load dispatching,” in Proc. on 2nd International Forum Applications on Neural Networks to Power Systems (ANNPS ’93) , Yokohama, Japan, Apr. 1993, pp. 145–150

[29] Yao X.,and Y. Liu, Fogel L. J., Back T., and Angeline P. J. , “Fast evolutionary programming,” in Proc. 5th Annual Conference on Evolutionary Programming, Cambridge, MA, 1996, pp. 451–460.

[30] Chellapilla K. , and Fogel D. B., “Two new mutation operators for enhanced search and optimization in evolutionary programming,” in SPIE International Symp. Optical Science and Engineering Instrum. Conference, 3165: Applications on Soft Compututing. Bellingham, WA: SPIE Press, pp. 260–269

[31] Yao X., Liu Y., and Lin G., “Evolutionary programming made faster,” IEEE Transactions on Evolutinary Computation, vol. 3, pp. 82–102, July 1999.

[32] Wolpert D. H. and Macready W. G., “No free lunch theorems for optimization,” IEEE Transactions on Evolutionary Computation, vol. 1, pp. 67–82, 1997.

[33] Wood A., Wollenberg B., “Power Generation, Operation and Control,” New York: Wiley, 1996.

[34] Kothari D.P., Dhillon J.S., “Power System Optimization”, Prentice Hall of India Private Limited, New Delhi, 2004.