Optimization of economic load dispatch with non-smooth cost functions using modified particle swarm optimization

A Project Report

Submitted by

PRABHU.G.V - 71205415005

in partial fulfillment for the award of the degree
of

Master of Engineering
in

Power Electronics and Drives

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE – 641 006

ANNA UNIVERSITY: CHENNAI 600 025

JULY 2007

BONAFIDE CERTIFICATE

Certified that this project report entitled **"Optimization of economic load dispatch with non-smooth cost functions using modified particle swarm optimization"** is the bonafide work of

Mr. Prabhu.G.V Register No. 71205415005

Who carried out the project work under my supervision.

Signature of the Head of the Department

Prof.K.Regupathy Subramanian

Signature of the Guide

Mrs.Mahalakshmi.R

Certified that the candidate with university Register No: 71205415005 was examined in Project viva- voce examination held on 05.07.07

Internal Examiner

5.7.07

External Examiner

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE - 641 006

MAHENDRA ENGINEERING COLLEGE

FS68172

Mahendhirapuri, Mallasamudram (West), Vadugapalayam (P.O), Thiruchengode (Tk.), Namakkal Dist. - 637 503, Tamil Nadu, India.

CERTIFICATE

Second National Conference on Recent Trends in Electrical, Electronics, Instrumentation & Communication Engineering (RETEEICOM - 07)

9th March 2007

This is to certify that ~~Ms.~~ / Mr. / ~~Dr.~~ .... G. V. PRABHU .... *of* KUMARAGURU COLLEGE OF TECHNOLOGY, COIMBATORE ...... *participated / presented a paper titled* Optimisation of Economic Load Dispatch with Non-smooth Cost Functions using MPSO .... *at RETEEICOM - 07 held on 9th March 2007, organized by the departments of* [illegible]

CONVENOR

PRINCIPAL

ABSTRACT

In the present scenario, various types of power generating stations are available using various renewable and non-renewable resources for generating electrical power. Power generating companies have turned their attention towards renewable resources like wind and sunlight because of the decrease in the availability of non-renewable resources like coal and oil. The companies who are generating power by any method turned their attention towards generating the power with minimum cost due to the drastic increase in demand and competition in the generating market. This led to the development of studies for economic load dispatch (ELD). All literature studies of ELD frequently use smooth cost functions with equality and inequality constraints that make the problem of finding the global optimum difficult using any mathematical approaches This project presents a new approach to optimize economic dispatch problems with non-smooth cost functions. A Modified Particle Swarm Optimization (MPSO) mechanism is suggested to deal with the equality and inequality constraints in the ELD problems. Moreover, a dynamic search-space reduction strategy is devised to accelerate the optimization process. The proposed MPSO is applied to test ELD problems with smooth cost functions and with non-smooth cost functions considering valve-point effects and multi-fuel problems, and the performance is compared with the conventional results. The number of iterations for this method is found to be less when compared to the conventional method.

ஆய்வு சுருக்கம்

இன்றைய காலக் கட்டத்தில், மின்சக்தியை உற்பத்தி செய்வதற்குரிய ஆற்றல் மூலங்களாக ஆக்கம் மற்றும் மறுஆக்கம் செய்யக்கூடிய மின் உற்பத்தி மூலங்கள் என இரண்டு வகைப்பட்ட மூலங்களை பயன்படுத்தி வருகிறோம் . மறு ஆக்கம் செய்ய முடியாத மின்உற்பத்தி ஆற்றல் மூலங்களுடைய அளவு குறைந்து வருவதால் மின் உற்பத்தியாளர்கள், (Generators) மறு ஆக்கம் செய்யக்கூடியதும் , நிறந்தரமாக உள்ளவையுமான இயற்கை சக்திகளான காற்று மற்றும் சூரிய ஒளி ஆகிய ஆற்றல் மூலங்களை பயன்படுத்தி வருகின்றனர்.

பெருகிவரும் , மின்சாரத் தேவைகளை நிறைவு செய்ய, மின் உற்பத்தியில் ஈடுபட்டுள்ள நிறுவனங்கள், பல சிக்கனமான முறைகளை கையாள நினைக்கிறார்கள். இதுவே பொருளாதார வழிமுறைகளின் முன்னேற்றத்திற்கு வழிவகுக்கிறது. இந்த ஆய்வுத்திட்டம், பொருளாதார சிக்கன வழிமுறைகளை, புதிய முறையில் அனுகுகிறது.

பொருளாதார செயல்பாடுகளில் ஏற்படும் சமமான, மற்றும் சமன் இல்லாத, தேவைகளினால் ஏற்படும் குறிக்கோள்களை, காரண காரியங்களின் அடிப்படையில் முழு அளவில் கணக்கிடும் வழிமுறைகளை கொண்டுள்ளது.

மேலும் இதன் மூலக்கூறுகளின் வளர்ச்சிகளைக் கணக்கிட்டு, அதன் வேகத்தை சீரான முறையில் அதிகப்படுத்த, முறையான வழிமுறைகளையும் கணக்கிடடு. இந்த (MPSO) திருத்தப்பட்ட மூலக்காரனிகளை ஆய்வு செய்து பொருளாதார பிரிவினைகளாள் ஏற்படும் சிக்கல்களையும் அதன் சமச்சீரான மற்றும் சமச்சீரற்ற விளைவுகளை கணக்கிட ஆய்வு செய்யப்பட்டிருக்கின்றது .

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without acknowledging the people who were responsible for the completion of the project.

I would like to sincerely thank Mrs. **R.Mahalakshmi**, Department of Electrical and Electronics Engineering, who spent her precious time, to guide me to complete our project successfully.

I express my sense of gratitude to **Prof.K.Regupathy Subramaniyam**, Head of the Department of Electrical and Electronics Engineering who gave his valuable suggestions to our project.

I would like to thank our honorable Principal **Dr.Joseph Thanikal** who had encouraged me in all moments.

I express my deepest gratitude to the management who gave their full support to our studies through out the curriculum.

I express my deep sense of gratitude to my friends, parents and all my Department staff members without whom this project would have been virtually impossible.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

List of Symbols

ELD	Economic Load Dispatch
PSO	Particle Swarm Optimization
MPSO	Modified Particle Swarm Optimization
F_T	Total generation cost
F_i	Cost function of generator *i*
a_i ,b_i , c_i,	Cost coefficients of generator i
P_i	*Power* of generator *i*
N	Number of generators.
$\|Vp\|$	voltage magnitude at the pth node
δ_p	phase angle at the pth node.
G	total generation
Pso	prespecified power
n	ratio of transformation
di, ei	Cost coefficients of generator i
V_i^k	velocity of individual *i* at iteration *k*,
ω	weight parameter,
c_1, c_2	weight factors,
$rand_1$, $rand_2$	random numbers between 0 and 1,
X_i^k	Position of individual *i* at iteration *k*,
$Pbest_i^k$	best position of individual *i* until iteration *k*,
$Gbest^k$	best position of the group until iteration *k*.
ω_{max} , ω_{min}	initial, final weights,
$Iter_{max}$	maximum iteration number,
$Iter$	current iteration number
ε	small positive real number
TC_i	function evaluated at the position of individual i

CHAPTER .1
INTRODUCTION

Like water, food and air electrical energy has become an integral part of daily personal and business lives. People have become so accustomed to flicking a switch and having instant light, action or communication that little thought is given to the process that produces the electrical energy or how it gets to where it is used. It is unique in that practically all that is produced is not stored but used instantly in the quantities that are needed. For alternatives to electrical energy, one must go back to the days of gas lamps, oil lamps, candles, and steam or water powered mechanical devices-and work days or leisure time that was limited to daylight hours for the most part.

The electric utility is the basic supplier of electrical energy and is perhaps unique in that almost everyone does business with it and is universally dependent on its product. Many people are unaware that a utility is a business enterprise and must meet costs or exceed them to survive. Unlike other enterprises producing commodities or services, it is obligated to have electrical energy available to meet all the customer demands when they are needed and that its prices are not entirely under its control.

The regulation of utilities by government agencies leads to the perception that utilities are in fact monopolies. People have alternatives in almost every other product they use such as choosing various modes of travel-auto, train, or plane. People can use gas, oil, or coal directly for their own energy needs or use them to generate their own electrical energy. Indeed some people today use sunlight or wind power to supplement their electrical needs. The point is that electrical energy supply from an electric utility is usually much more convenient and economical than producing it individually. Some larger manufacturing firms find it feasible to provide their own electrical energy by using their waste energy (cogeneration) or having their own individual power plants. In some cases legislation makes it mandatory to purchase the excess energy from their sources at rates generally higher than what the utility can produce it for.

The fact remains that utilities must pay for the materials, labor and capital they require and pay taxes just like other businesses. In obtaining these commodities necessary to every business, utilities must compete for them at lower prices generally dictated by the market place, while the prices charged for the product produced-electrical energy are limited by government agencies.

With the dawn of the new era in which the electric incandescent light replaced oil lamps and candles, sources of electric energy had to be found and developed. Gas light companies were giving way to geographically small electric companies. For instance on long island, New York, a company called "Babylon Electric Light Company" was formed in 1886. It would surprise many LI residents today that the low level waterfall on Sumpwarm's Creek in Babylon was used to light up eight stores and three street lights and that the dam still exists. Similar examples can be cited for other communities throughout the world. Most small electric companies started out using hydro power or steam engines to generate their electrical energy.

As the innovation caught on and the electrical energy requirements grew from the use of lights and electrically driven equipment, so did the growth of electric power generators. The size of generators grew from a few hundred watts to thousands of kilowatts. New sources of fuel needed to power generators led to coal, oil and gas fired boilers. New ways of transmitting electrical energy for some distance was found and led to larger central stations instead of the small local area stations. As AC (alternating current) transmission developed to permit sending power over longer distances, the early small electric companies consolidated their territories and started to interconnect their systems. Planning and development of these early generating stations were not hindered by environmental restrictions or government regulations. Their main concern was rising of enough capital to build the stations and selecting the best site for the fuel to be used and the load to be served.

1.1.Importance of electrical energy:

Energy may be needed as heat, as light, as motive power etc.The present day advances in science and technology has made it possible to convert electrical energy in to any desirable form. This has given electrical energy a place of pride in the modern world. The survival of individual undertakings and our social structures depends primarily upon low cost and uninterrupted supply of electrical energy. In fact, the advancement of a country is measured in terms of per capita consumption of electrical energy.

1.2 Comparison of energy sources:

The chief sources of energy used for the generation of electrical energy are water, fuels and nuclear energy. Below is given their comparison in tabular form:

Table 1.1 Comparison of energy sources

S.No	Particular	Water-power	Fuels	Nuclear energy
1.	Initial cost	High	Low	Highest
2.	Running cost	Less	High	Least
3.	Reserves	Permanent	Exhaustible	Inexhaustible
4.	Cleanliness	Cleanest	Dirtiest	Clean
5.	Simplicity	Simplest	Complex	Most complex
6.	Reliability	Most reliable	Less reliable	More reliable

1.3 Objective:

The proposed project deals with the Optimization of Economic Load Dispatch Problem using Modified Particle Swarm Optimization and the performance is compared with the conventional methods.

1.4 Organization of the report:

Chapter 1

This chapter deals with the general concepts and the importance of electrical energy and the comparison of energy sources. The generation of electrical energy by various methods, its individuals' merits and demerits are tabulated in table 1.1

Chapter 2

The brief explanation of the economic load dispatch with the essential equality and inequality constraints is stated in this chapter.

Chapter 3

This chapter deals with the overview of Particle Swarm Optimization. Modified Particle Swarm Optimization methodology is explained in detail, which is used to solve the economic dispatch.

Chapter 4

The program for the economic load dispatch using Modified Particle Swarm Optimization algorithm, implemented using MATLAB is presented in this chapter.

Chapter 5

The results are given in this chapter.

Chapter 6

The conclusion and the future scope is presented in this chapter.

TABLE 1.1. COMPARISON OF POWER GENERATION METHODS

S.No	Item	Steam power station	Hydro-electric power plant	Diesel power plant	Nuclear power plant
1.	Site	Such plants are located at a place where ample supply of water and coal is available, transportation are adequate	Such plants are located where large reservoirs can be obtained by constructing a dam e.g., in hilly areas.	Such plants can be located at any place because they require less space and small quantity of water.	These plants are located away from thickly populated areas to avoid radio-active pollution.
2	Initial cost	Initial cost is lower than those of hydroelectric and nuclear power plants.	Initial cost is very high because of dam construction and excavation work.	Initial cost is less as compared to other plants	Initial cost is highest because of huge investment on building a nuclear reactor.
3	Running cost	Higher than hydroelectric and nuclear plant because of the requirement of large amount of coal.	Practically nil because no fuel is required.	Highest among all plants because of high price of diesel.	Except the hydro electric plant, it has the minimum running cost because small amount of fuel can produce relatively large amount of power.
4	Limit of source of power	Coal is the source of power which has limited reserves all over the world.	Water is the source of power which is not dependable because of wide variations in the rainfall every year.	Diesel is the source of power which is not available in huge quantities due to limited reserves.	The source of power is the nuclear fuel which is available in sufficient quantity. It is because small amount of fuel can produce huge power

5	Cost of fuel transportation	Maximum because huge amount of coal is transported to the plant site	Practically nil.	Higher than hydro and nuclear power plants.	Minimum because small quantity of fuel is required.
6	Cleanliness and simplicity	Least clean as atmosphere is polluted due to smoke.	Most simple and clean.	More clean than steam power and nuclear power plants	Less cleaner than hydro-electric and diesel power plants.
7	Overall efficiency	Least efficient. Overall efficiency is about 25%	Most efficient. Overall efficiency is about 85%	More efficient than steam power station. Efficiency is about 35%	More efficient than steam power plants.
8	Starting	Requires a lot of time for starting	Can be started instantly.	Can be started quickly	Can be started easily
9	Space required	These plants need sufficient space because of boilers and other auxiliaries.	Requires very large area because of the reservoir	Requires less space.	These require minimum space as compared to any other plant of equivalent capacity
10	Maintenance cost	Quite high as skilled operating staff is required	Quite low	Less	Very high as highly trained personnel are required to handle the plant.
11	Transmission and distribution cost	Quite low as they are generally located near the load centers.	Quite high as these are located quite away from the load centers	Least as they are generally located at the centre of gravity of the load.	Quite low as these are located near load centres
12	Standby losses	Maximum as the boiler remains in operation even when the turbine is not working.	No standby losses	Less standby losses	less

CHAPTER.2

FORMULATION OF ECONOMIC DISPATCH

The classical economic dispatch problem is an optimization problem that determines the power output level of each online generator that will result in a least cost system operating state. The objective of the classical economic dispatch is to minimize the total system cost, where the total system cost is a function determined by the sum of the cost functions of each online generator. This power allocation procedure is carried out considering the power balance between generation and loads, as well as the feasible operating region for each generating unit.

2.1. Basic Economic Dispatch Formulation

Economic dispatch is one of the most important problems to be solved in the operation and planning of a power system. The objective of the economic dispatch problem is to minimize the total fuel cost of thermal power plants subjected to the operating constraints of a power system. In general, it can be formulated mathematically with an objective function.

The objective of the classical economic dispatch is to minimize the total system cost (2.1) by adjusting the power output of each of the generators connected to the grid. The total system cost is modeled as the sum of the cost function of each generator. [1][5][2]

$$F_T = \sum_{i=1}^{N} F_i\left(P_i\right) \tag{2.1}$$

The cost function of each generator establishes the relationship between the power injected to the system by the generator and the cost incurred to load the machine to that capacity. Generators are typically modeled by smooth quadratic functions such as (2.2), in order to simplify the corresponding optimization problem, as well as to facilitate the application of classical techniques.

$$F_i\left(P_i\right) = a_i + b_i P_i + c_i P_i^2 \tag{2.2}$$

Where,

F_T = Total generation cost,

F_i = cost function of generator *i*,

a_i ,b_i , c_i = Cost coefficients of generator *i*,

P_i = *Power* of generator i,

N = Number of generators.

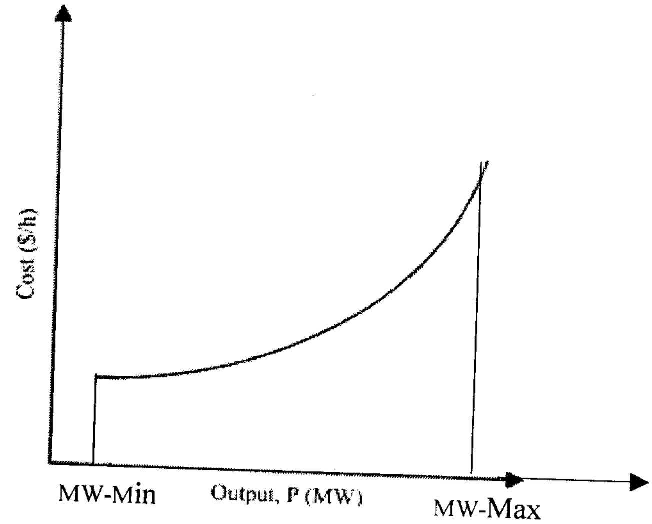

Figure 2.1 Typical fuel cost function of a thermal generating unit

Where MW-Min is the minimum loading limit below which it is uneconomical to operate the unit and MW- Max is the maximum output limit. The input-output curve has discontinuities at steam value openings which have not been indicated in the figure (2.1)

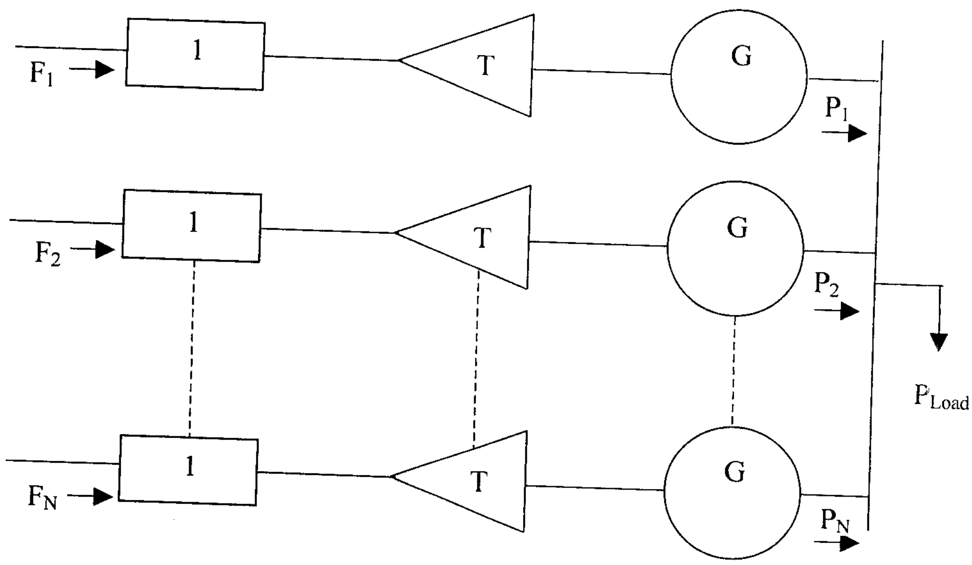

Figure 2.2 N Generating units committed to serve a load of P_{load}

2.2 CONSTRAINTS:

2.2.1 Equality constraints

The important constraint in the generating system is that the generated power should be equal to the summation of the power to be delivered to the customer with the losses while sending the same. [13]

Generated power = Power to be delivered + Losses

2.2.2 Inequality constraints:

(a) Generator constraints:

The KVA loading on a generator is given by

$\sqrt{P_p^2 + Q_p^2}$ and this should not exceed a prespecified value Cp because of the temperature raise conditions, i.e,

$$P_p^2 + Q_p^2 \le C_p^2 \qquad (2.3)$$

The maximum active power generation of a source is limited again by thermal considerations and also minimum power generation is limited by the flame instability of a boiler. If the power output of a generator for optimum operation of the system is less than a prespecified value P_{min} , the unit is not put on the bus bar because it is not possible to generate that low value of power from that unit. Hence the generated power Pp cannot be outside the range stated by the inequality, i.e,

$$P_{p\min} \le P_p \le P_{p\max} \qquad (2.4)$$

Similarly the maximum and minimum reactive power generations of the source are limited. The maximum reactive power is limited because of overheating of the rotor and minimum is limited because of the stability limit of the machine. Hence the generator reactive power Q_p cannot be outside the range stated by the inequality, i.e.,

$$Q_{p\min} \le Q_p \le Q_{p\max} \qquad (2.5)$$

(b) Voltage constraints:

It is essential that the voltage magnitudes and phase angles at various nodes should vary within certain limits. The voltage magnitudes should vary within certain limits because otherwise most of the equipment connected to the system will not operate satisfactorily or additional use of voltage regulating devices will make the system uneconomical. Thus

$$|V_{p\min}| \le |V_p| \le |V_{p\max}|$$

$$\delta_{p\min} \le \delta_p \le \delta_{p\max} \tag{2.6}$$

Where |Vp| and δ_p stand for the voltage magnitude and phase angle at the pth node.

The normal operating angle of transmission line lies between 30° to 45o for transient stability reasons; therefore a higher limit is imposed on angle δ. A lower use of assures proper utilization of transmission facility.

(c) Running spare capacity constraints:

These constraints are required to meet: (i) the forced outages of one or more alternators on the system, and (ii) the unexpected load on the system

The total generation should be such that in addition to meeting load demand and losses a minimum spare capacity should be available. i.e,

$$G \ge P_p + P_{so} \tag{2.7}$$

Where G is the total generation and Pso is prespecified power. A well planed system is one in which this spare capacity P_{so} is minimum.

(d) Transformer tap settings:

If an autotransformer is used, the minimum tap setting could be zero and the maximum one, i.e,

$$0 \le t \le 1.0$$

Similarly for a two winding transformer if tapings are provided on the secondary side;

$$0 \le t \le n$$

Where n is the ratio of transformation. Phase shift limits of a phase shifting transformer

$$\theta_{p\min} \le \theta_p \le \theta_{p\max} \tag{2.8}$$

2.3 NON-SMOOTH COST FUNCTIONS

2.3.1 Valve-point effects

The generator cost function is obtained from data points taken during 'heat run' tests, when input and output is measured as the unit is slowly varied in the operating region. But in real system the operating region may not be a smooth and continuous function as the system contains a number of different steam unit characteristics. Large steam turbine generators will have a number of steam admission valves that are opened in sequence to obtain ever–increasing output of the unit. As the unit loading increases the input to the unit increases and the incremental heat rate decreases between the opening points for any two valves, however, when a valve is first opened, the throttling losses increases rapidly and the incremental heat rate rises suddenly. This is "valve point" effect which leads to non-smooth, non-convex input-output characteristics, to be solved using the heuristic techniques.[3] [6]

When a steam admission valve starts to open, a sudden increase in losses occurs, which results in ripples in the unit's cost function (see Fig.2.3) valve-points are those output levels at which a new admission valve is open. When valve point effects are considered, the ED problem becomes extremely difficult to solve via conventional gradient based techniques, due to the abrupt changes and discontinuities present in the corresponding incremental cost functions. Valve point effects are usually modeled by adding a recurring rectified sinusoidal term to the basic quadratic cost function, as it is shown in equ (2.9) below.

$$F_i\left(P_{G_i}\right)=a_i+b_i\,P_{G_i}^{\ 2}+\left|\,d_i\sin\left(e_i\left(P_{G_i}^{\ \min}-P_{G_i}\right)\right)\right| \qquad (2.9)$$

where

ai , bi, ci, di and ei are the cost coefficients of unit i.

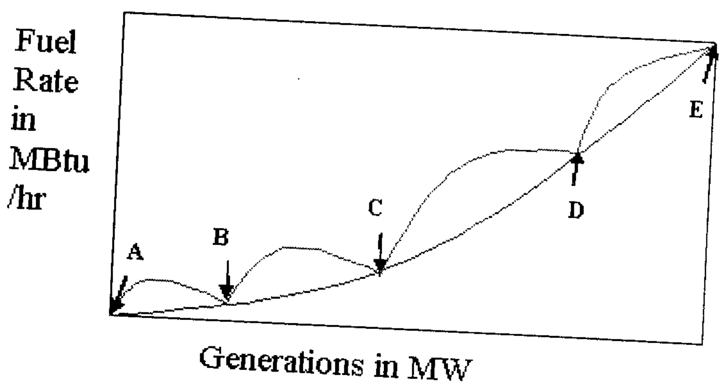

Figure 2.3. Valve Point Curve. (A – Primary Valve, B – Secondary Valve, C – Tertiary Valve, D – Quaternary Valve and E – Quandary Valve).

2.3.2 Multiple fuels

Traditionally, the cost function of each generator has been approximately represented by a single quadratic cost function. In practice, generating units are capable of operating using different types of fuels. The use of multiple fuel types may result in multiple cost curves that are not necessarily parallel or continuous. The lower region of the resulting cost curve determines which fuel type is most economical to bum.[7][4]

This cost function can be represented by a piecewise curve (see Fig. 2.4), and the segments are defined by the range in which each fuel is used (2.10). The ED problem with piecewise quadratic cost curves is very difficult to solve by standard techniques. Piecewise quadratic cost functions have as many segments as fuel types.

$$Fi(Pi)=\begin{cases} a_{i1}+b_{i2}P_i+C_{i1}P_i^2 & if P_{i\min}\le P_i\le P_{i1} \\ a_{i2}+b_{i2}P_i+C_{i2}P_i^2 & if P_{i1}\le P_i\le P_{i2} \\ .. & \\ .. & \\ a_{in}+b_{in}P_i+C_{in}P_i^2 & if P_{in-1}\le P_i\le P_{i\max} \end{cases} \qquad (2.10)$$

Where a_i, b_i, c_i are the cost coefficients of generator I for the p-th power level

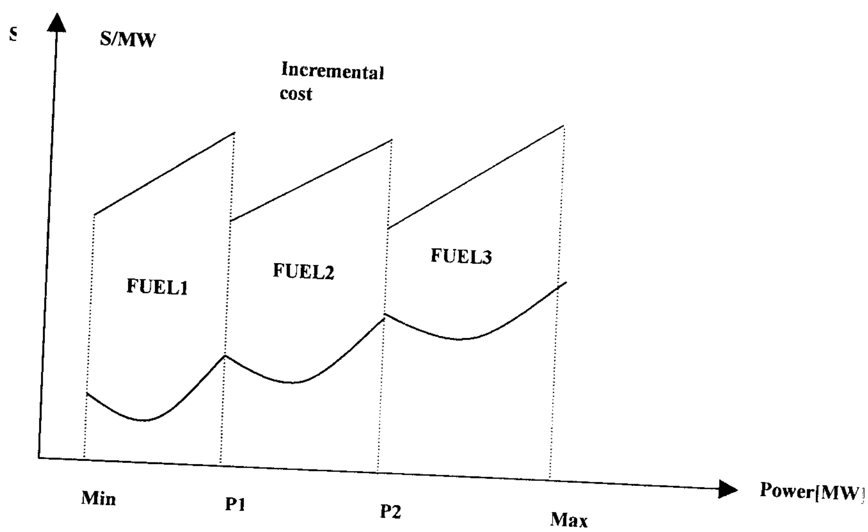

Figure 2.4 Piecewise quadratic and incremental cost functions of a generator

P-1869

CHAPTER.3
PARTICLE SWARM OPTIMIZATION

3.1 EVOLUTIONARY PROGRAMMING (EP) TECHNIQUES:

Roughly 45 years ago, evolutionary programming (EP) was proposed for the evolution of the finite state machines in order to solve prediction tasks. Since then a number of modifications, enhancements and implementations have been proposed and investigated. EP was extended to work on real-valued object variables based on normally distributed mutations. In EP, mutation is the only operator used to generate new offspring. In recent years, a lot of developments have taken place in EP in terms of efficiency and quality. The mutation is implemented often by adding to the parent; a random number from a certain distribution. An important parameter of the Gaussian mutation is its standard deviation which is also called strategy parameter. Larger jumps are beneficial when the current solution is far away from the global optimum, but such large jumps near the optimum point deteriorate the solution process .Chellapilla et al have proposed another variation of mutation with mean of Gaussian and Cauchy mutations (called MFEP) with an objective of having a step size greater than Gaussian and less than Cauchy types so that advantages of both Gaussian as well as Cauchy mutations are exploited. Again, Xin Yao et al have proposed an improved fast EP (IFEP) using both Gaussian and Cauchy mutations for creation of offspring from the same parent and better ones are chosen for next generation. They have shown that this method performs better than all other EP methods in almost all the benchmark functions.[9]

PSO technique, first introduced by Kennedy and Eberhart, is a recent addition to modern heuristic algorithms. It was developed following the simulation of a simplified social system and has been found to be robust in solving continuous non-linear optimization problems. PSO is an also population based search procedure in which individuals, called particles, change their positions (states) with time. The PSO technique can generate high quality solution within shorter calculation time and more stable convergence characteristic than other stochastic methods. More recent trend is to develop more robust heuristic techniques by integrating PSO and EP techniques so that positive features of both the techniques are exploited.

3.2 OVERVIEW OF THE PSO

Kennedy and Eberhart developed a PSO algorithm in 1995, which is a population-based stochastic optimization technique, based on the behavior of individuals (i.e., particles or agents) of a swarm. Its roots are in zoologist's modeling of the movement of individuals (e.g., fishes, birds, and insects) within a group. It has been noticed that members of the group seem to share information among them, a fact that leads to increased efficiency of the group. The PSO algorithm searches in parallel using a group of individuals similar to other AI based heuristic optimization techniques. Each individual corresponds to a candidate solution to the problem. Individuals in a swarm approach to the optimum through its present velocity, previous experience, and the experience of its neighbors.[8.] [11]

In a physical *n*-dimensional search space, the position and velocity of individual *i* are represented as the vectors X_i = (x_{i1} ,........, x_{in}) and V_i = (v_{i1} ,......., v_{in}), in the PSO algorithm. Let $Pbest_i$ = (x_{i1}^{Pbest} ,........., x_{in}^{Pbest}) , and $Gbest_i$ = (x_{i1}^{Gbest} ,........., x_{in}^{Gbest}), be the best position of individual *i* and its neighbors' best position so far, respectively. Using the information, the updated velocity of individual *i* is modified under the following equation in the PSO algorithm.[10][12]

$$V_i^{k+1} = \omega V_i^k + c_1 rand_1 \times \left(Pbest_i^k - X_i^k\right) + c_2\, rand_2 \times \left(Gbest^k - X_i^k\right) \tag{3.1}$$

where,

V_i^k = velocity of individual *i* at iteration *k*,

ω = weight parameter,

c_1, c_2 = weight factors,

$rand_1$, $rand_2$ = random numbers between 0 and 1,

X_i^k = Position of individual *i* at iteration *k*,

$Pbest_i^k$ = best position of individual *i* until iteration *k*,

$Gbest^k$ = best position of the group until iteration *k*.

In this velocity updating process, the values of parameters such as ω , c_1 , and c_2 should be determined in advance. In general, the weight ω is set according to the following equation.

$$\omega = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{Iter_{max}} \times Iter \tag{3.2}$$

where,

ω_{max} , ω_{min} initial, final weights,

$Iter_{max}$ = maximum iteration number,

$Iter$ = current iteration number.

Each individual moves from the current position to the next one by the modified velocity using the following equation:

$$X_i^{k+1} = X_i^k + V_i^{k+1} \tag{3.3}$$

Fig. 3.1 shows the concept of the searching mechanism of PSO using the modified velocity and position of individual *I* based on (3.1) and (3.3) if the values of ω, c_1 , c_2 , $rand_1$, $rand_2$ are 1.[2]

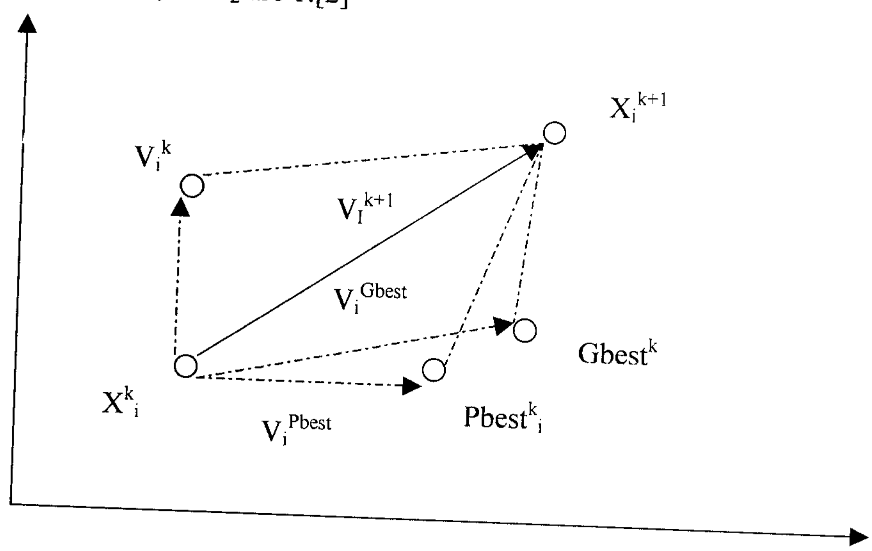

Figure 3.1 Concept of modification of a searching point by PSO.

3.2.1 PSO algorithm

PSO algorithm can be simply described as follows.

Step 1: Randomly generate the position and velocity for each particle.

Step 2: Compute the fitness of each particle, and get the *pbest* and *gbest* values at the current iteration.

Step 3: Use (3.1) and (3.3) to update the position and velocity for each particle.

Step 4: If there are infeasible particles due to the violated constraints, introduce the penalty terms to their fitness.

Step 5: Set the velocity at the limit value for any velocity-violated particle J

$$v^{j}_{i,t+1} = V^{\max}, \quad if\ v^{j}_{i,t+1} > V^{\max} \quad i = 1,2,....n \tag{3.4}$$

$$v^{j}_{i,t+1} = V^{\min}, \quad if\ v^{j}_{i,t+1} < V^{\min} \quad i = 1,2,....n \tag{3.5}$$

Step 6: If the maximum iterations are reached, stop; else, go to step 2.

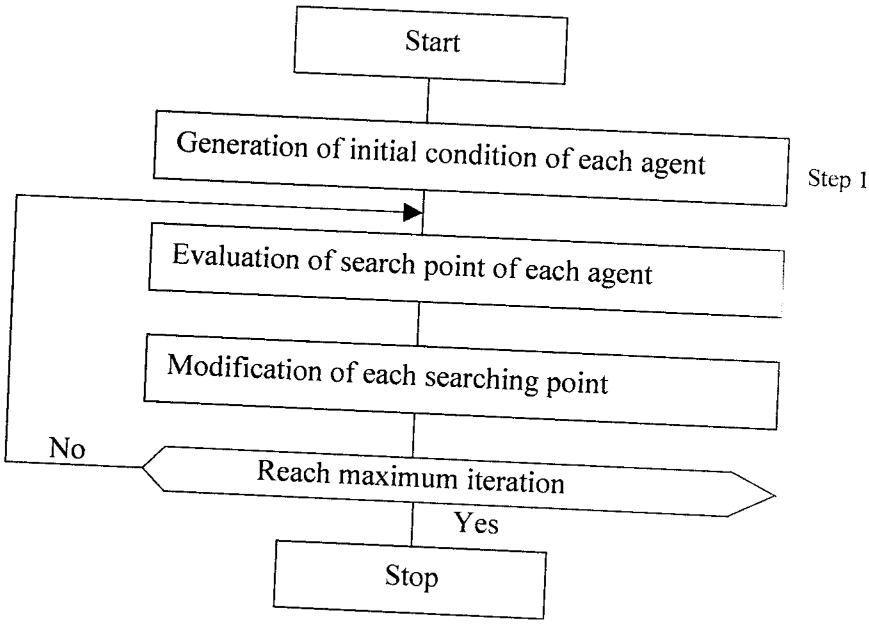

Figure 3.2 General flow graph of PSO

3.3 MODIFIED PSO

This method integrates the particle swarm optimization with the simulated annealing algorithm. It can solve the problem of local minimum of the particle swarm optimization, and narrow the field of search continually, so it has higher efficiency of search. The equality and inequality constraints of the ED problem are dealt using the modified PSO algorithm. Additionally, to accelerate the convergence speed, the dynamic search-space reduction strategy is devised. The process of the modified PSO algorithm can be summarized as follows[13],

Step 1: Initialization of a group at random while satisfying constraints.

Step 2: Velocity and position updates while satisfying constraints.

Step 3: Update of *Pbest* and *Gbest*.

Step 4: Activation of space reduction strategy.

Step 5: Go to Step 2 until satisfying stopping criteria.

In the subsequent sections, the detailed implementation strategies of the MPSO are described.

3.3.1 INITIALIZATION AND STRUCTURE OF INDIVIDUALS:

In the initialization process, a set of individuals is created at random. In this project, the structure of an individual for ED problem is composed of a set of elements (i.e., generation outputs). Therefore, individual's position at iteration 0 can be represented as the vector of $X_i^0 = (P_{i1}^0, \ldots\ldots\ldots\ldots, P_{in}^0)$ where n is the number of generators. The velocity of individual i (i.e: $V_i^0 = (v_{i1}^0, \ldots\ldots v_{in}^0)$)correspond to the generation update quantity covering all generators. The elements of position and velocity have the same dimension, i.e., MW in this case. Note that it is very important to create a group of individuals satisfying the equality constraint and inequality constraints. That is, summation of all elements of individual i should be equal to the total system demand D and the created element j of individual at random should be located within its boundary. Although we can create element j of individual i at random satisfying the inequality constraint by mapping [0, 1] into $[P_{j\,min}, P_{j\,max}]$, it is necessary to develop a new strategy to handle the equality constraint. To do this, the following procedure is suggested for any individual in a group.

Step 1: Set j=1;

Step 2: Select an element (i.e., generator) of an individual at random.

Step 3: Create the value of the element (i.e., generation output) at random satisfying its inequality constraint.

Step 4: If j=n-1 then go to Step 5; otherwise j=j+1 and go to step 2 ;

Step 5: The value of the last element of an individual is determined by subtracting $\sum_{j=1}^{n-1} P_{ij}^0$ from the total system demand. If the value is in the range of its operating region then go to Step 6 otherwise go to Step 1.

Step 6: Stop the initialization process.

After creating the initial position of each individual, the velocity of each individual is also created at random. The following strategy is used in creating the initial velocity:

$$(P_{j\,min} - \varepsilon) - P_{ij}^0 \leq v_{ij}^0 \leq (P_{j\,max} + \varepsilon) - P_{ij}^0 \quad (3.6)$$

where ε is a small positive real number. The velocity of element j of individual i is generated at random within the boundary. The developed initialization scheme always guarantees to produce individuals satisfying the constraints while maintaining the concept of the PSO algorithm. The initial $Pbest_i$ of individual i is set as the initial position of individual i and the initial *Gbest* is determined as the position of an individual with minimum payoff of (2.1).

3.3.2 VELOCITY UPDATE:

To modify the position of each individual, it is necessary to calculate the velocity of each individual in the next stage, which is obtained from (3.1). In this velocity updating process, the values of parameters such as ω, c_1 and c_2 should be determined in advance. In this paper, the weighting function is defined as follows

$$\omega = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{Iter_{max}} \times Iter \qquad (3.7)$$

where

$\omega_{max}, \omega_{min}$ Initial, final weights;

$Iter_{max}$ maximum iteration number;

Iter current iteration number

3.3.3 POSITION MODIFICATION CONSIDERING CONSTRAINTS:

The position of each individual is modified by (3.3). The resulting position of an individual is not always guaranteed to satisfy the inequality constraints due to over/under velocity. If any element of an individual violates its inequality constraint due to over/under speed then the position of the individual is fixed to its maximum/ minimum operating point. Therefore, this can be formulated as follows:

$$P_{ij}^{k+1} = \begin{cases} P_{ij}^{k} + V_{ij}^{k+1} & \text{if } P_{ij\min} \le P_{ij}^{k} + V_{ij}^{k+1} \le P_{ij\max} \\ P_{ij\min} & \text{if } P_{ij}^{k} + V_{ij}^{k+1} < P_{ij\min} \\ P_{ij\max} & \text{if } P_{ij}^{k} + V_{ij}^{k+1} > P_{ij\max} \end{cases} \qquad (3.8)$$

Fig. 4 illustrates how the position of element j of individual i is adjusted to its maximum when the over-velocity situation occurs. Although the aforementioned method always produces the position of each individual satisfying the inequality constraints, the problem of equality constraint still remains to be Resolved. Therefore, it is necessary to develop a new strategy such that the summation of all elements in an individual is equal to the total system demand. To resolve the equality constraint problem without intervening the dynamic process inherent in the PSO algorithm, we propose the following heuristic procedures:

Step 1: Set j=1. Let the present iteration be k.

Step 2: Select an element (i.e., generator) of individual i at random and store in an index array A (n),

Step 3: Modify the value of element j using (7), (8), and (11).

Step 4: If j=n-1 then go to Step 5, otherwise j=j+1and go to Step 2.

Step 5: The value of the last element of individual i is determined by subtracting $\sum_{j=1}^{n-1} P_{ij}^{k}$ from D. If the value is not within its boundary then adjust the value using (11) and go to Step 6, otherwise go to Step 8.

Step 6: Set i=1,

Step 7: Readjust the value of element I in the index array A(n) to the value satisfying equality condition $D - \sum_{i=1, j\neq 1}^{n} P_{ij}^{k}$ If the value is within its boundary then go to Step 8; otherwise, change the value of element I using(11).Set I=I+1 , and go to Step 7. If I = n+1 , go to step 1.

Step 8: Stop the modification procedure.

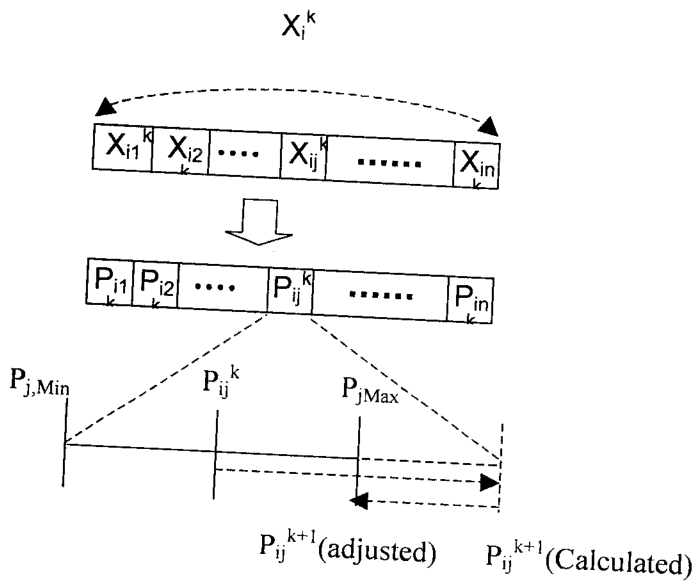

Figure. 3.3. Adjustment strategy for an individual's position within boundary.

3.3.4 UPDATE OF P BEST AND G BEST:

The *Pbest* of each individual at iteration k+1 is updated as follows

$$Pbest_i^{k+1} = X_i^{k+1} if TC_i^{k+1} < TC_i^k$$
$$Pbest_i^{k+1} = Pbest_i^k if TC_i^{k+1} \geq TC_i^k \quad (3.9)$$

Where

TC_i the object function evaluated at the position of individual i. Additionally, *Gbest* at iteration k+1 is set as the best evaluated position among $Pbest_i^{k+1}$

3.3.5 SPACE REDUCTION STRATEGY:

To accelerate the convergence speed to the solutions, the MPSO has introduced the search space reduction strategy. This strategy is activated in the case when the performance is not increased during a prespecified iteration period. In this case, the search space is dynamically adjusted (i.e., reduced) based on the "distance" between the *Gbest* and the minimum and maximum output of generator j. To determine the adjusted minimum/maximum output of generator j at iteration k, the distance is multiplied by the predetermined step-size Δ and subtracted (added) from the maximum (minimum) output at iteration k as described in (3.10)

$$P_{j\max}^{k+1} = P_{j\max}^{k} - (P_{j\max}^{k} - Gbest_j^k) \times \Delta$$
$$P_{j\min}^{k} = P_{j\min}^{k} + (P_{j\min}^{k} - Gbest_j^k) \times \Delta \quad (3.10)$$

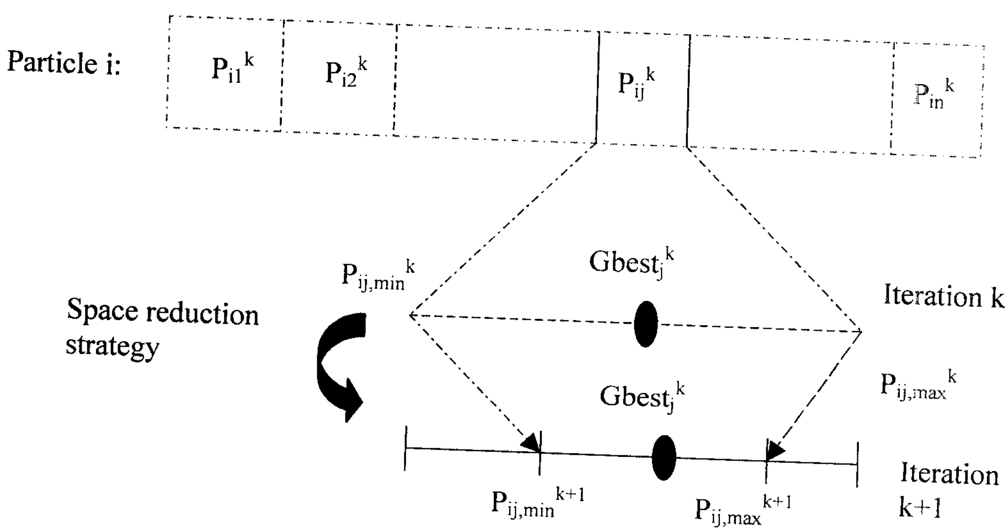

Figure. 3.4. Schematic of the dynamic space reduction strategy.

3.3.6 STOPPING CRITERIA:

The MPSO is terminated if the iteration approaches to the predefined maximum iteration or when the specified stopping condition is satisfied.

CHAPTER.4

PROGRAM:

4.1 MPSO METHOD FOR ELD:

```
clc
clear

% The old system variables are stored in a file
"ecogenvariables". User may use the same values or enter
new values for a new system.

loadnew = '';
loadnew = input(['Do you want to load NEW values (Y)/(N) : '],'s');
if (loadnew == 'Y' || loadnew == 'y')
    noofgen = input(['Give the number of generators : ']);

    % The system constants are entered one by one. like
A + Bx + Cx^2. first enter A - constant term, enter B -
x term coefficient, enter C - x squared term coefficient.
Then enter the number of valves, then minimum generation
capacity and maximum generation capacity.

for gen = 1:noofgen
      costfunc(gen, 1) = input(['\n\nEnter cost function of generator ', ...
          num2str(gen), ' (constant term) : ']);
      costfunc(gen, 2) = input(['Enter cost function of generator ', ...
          num2str(gen), ' (X term)         : ']);
      costfunc(gen, 3) = input(['Enter cost function of generator ', ...
          num2str(gen), ' (X^2 term)       : ']);

    valves(gen) = input(['\nEnter number of valves for generator ', ...
        num2str(gen), ' : ']);

    minmaxgen(gen, 1) = input(['MINIMUM generation capacity of generator ', ...
        num2str(gen), ' : ']);
```

```
minmaxgen(gen, 2) = input(['MAXIMUM generation capacity of generator ', ...
    num2str(gen), ' :  ']);
end
save ecogenvariables;
else
    load ecogenvariables;
end


clc
looprun = 1;
exclude = zeros(1,noofgen);
while looprun ==1
    demand = 0;
    minmaxgentot = sum(minmaxgen);
    while demand == 0

            %Enter total demand from all generators.This
value must be greater than sum of the minimum generation
capacities of all generators.

            demand = input(['\n\nEnter total demand from all generators  :  ']);
if (demand < minmaxgentot(1))
disp('Demand is less than minimum of all generators');
        demand = 0;
    end
end


demandratio = demand/minmaxgentot(2);
%ratio = actual demand / total of maximum of all gens.
    initialpower = minmaxgen(:,2).*demandratio;
%initial power = Max gen capacity * ratio
exclude(1) = 0;
demand1 = demand;
minmaxgentot1 = minmaxgentot(2);
```

```
% sum of max capacity of all gens.

inequalityok = 0;
while inequalityok == 0
inequalityok = 1;
for gen = 1:noofgen
if initialpower(gen) < minmaxgen(gen,1)
% initial power < minimum capacity
inequalityok = 0;
difference = initialpower(gen) - minmaxgen(gen,1);
%note the difference
initialpower(gen) = minmaxgen(gen,1);
%and assign minimum capacity
exclude(gen) = 1;
%and exclude this generator from any adjustment in this
iteration
demand1 = demand1 - initialpower(gen);
%balance demand after assigining above gen
minmaxgentot1 = minmaxgentot1 - minmaxgen(gen,2);
%find new ratio after removing this gens capacity
demandratio1 = demand1/minmaxgentot1;

for gen1 = 1:noofgen
%assign new initial values after adjusting the
if (gen1 = gen && exclude(gen1)~=1)
%demand for above gen to min capacity
initialpower(gen1) = minmaxgen(gen1,2) * demandratio1;
            end
          end
        end
    end
end

%power = [300 300 250];
```

```
%minmaxgen = [200 600; 150 400; 100 200];
%the above specified values are taken from
reference[1]

maxiter = 500;
iter = 0;
%initial velocity = (max capacity - minimum
capacity)*0.01
initialvelocity = (minmaxgen(:,2) - minmaxgen(:,1)) ./ 100;
weightparametermax = 1;
weightparametermin = 0;
initialvelocity1 = initialvelocity;
runpower1 = initialpower;
%systems operation power = inital power
costmean1 = 0;
convergence = 1;
%execute following numerical method analysis for
finding new powers for all generators. Inside the loop
the above technique of power distribution is used again.
while (convergence > 0.001)
iter = iter + 1;
weightparameter = weightparametermax -
(weightparametermax-weightparametermin)...
*(iter/maxiter);

%velocity update
newvelocity = initialvelocity1 .* weightparameter;

for gen = 1:noofgen
cost1(gen) = costfunc(gen,1) +       costfunc(gen,2)*runpower1(gen)+...
costfunc(gen,3)*(runpower1(gen)^2);
% cost = A + Bx + Cx^2 (x = power generated)
end
```

```
runpower2 = runpower1 + newvelocity;
%new values for power generation => dp
for gen = 1:noofgen
cost2(gen) = costfunc(gen,1) + costfunc(gen,2)*runpower2(gen)+...
costfunc(gen,3)(runpower2(gen)^2);
%new cost values
end

incrementalcost = cost2-cost1;
%differential cost => dc
costrate = incrementalcost ./ runpower1'; % dp/dc
costmean2 = mean(costrate);
costrate1 = costmean2 - costrate;
%deviation of each gen from average
initialvelocity1 = 10*(initialvelocity1 .* costrate1');
%update velocity
newvelocity = initialvelocity1 .* weightparameter;
%include weight parameter

runpower1 = runpower1 + newvelocity;
convergence = abs(costmean1 - costmean2);
inequalityok = 0;

%power distribution routine

while inequalityok == 0
inequalityok = 1;
for gen = 1:noofgen
if runpower1(gen) < minmaxgen(gen,1)
inequalityok = 0;
difference = runpower1(gen) - minmaxgen(gen,1);
runpower1(gen) = minmaxgen(gen,1);
exclude(gen) = 1;
```

```
demand1 = demand1 - runpower1(gen);
minmaxgentot1 = minmaxgentot1 - minmaxgen(gen,2);
demandratio1 = demand1/minmaxgentot1;

for gen1 = 1:noofgen
if (gen1 ~= gen && exclude(gen1)~=1)
runpower1(gen1) = minmaxgen(gen1,2) * demandratio1;
                end
              end
            end
          end
        end
        if (iter == maxiter)
          convergence = 0;
% to come out of loop at max iteration
              end
        disp([' ']);
        for gen = 1:noofgen
            disp (['Run power for generator ', num2str(gen), ' after iteration ',
num2str(iter), ' is ', num2str(runpower1(gen))]);
        end
    end
    disp([' ']);
    disp([' ']);
% after completing one calculation you may run the same
program for new
% demand for the same system of generators.
        demandnew = ' ';
    demandnew = input(['Do you want to load NEW values (Y-Yes)/(N-No)/(E-Exit)
:  '],'s');
    if (demandnew == 'Y' || demandnew == 'y')
        looprun = 1;
    else
        looprun = 0;
```

```
    end
end
```

4.2 CONVENTIONAL ELD METHOD:

```
clc
clear
noofgen = 3;
costfunction = [510.0 7.2 0.00142; 310 7.85 0.00194; 78.0 7.97 0.00482];
fuelcost = [1.1 ;1.0 ;1.0];
fuelcost = [fuelcost fuelcost fuelcost];

% power = [375 300 175];
% minmax = [200 600; 150 400; 100 200];

for n = 1:noofgen
    power(n) = input(['Initial power generation condition of generator ',...
        num2str(n), ' is  : ']);
    minmax(1,n) = input(['Input minimum generation value of generator ',...
        num2str(n), ' :  ']);
    minmax(2,n) = input(['Input maximum generation value of generator ',...
        num2str(n), ' :  ']);
    disp([power(n), minmax(:,n)']);
end
ptotal = sum(power);
costfn = (costfunction .* fuelcost);
lamda = [0 0 0];
lamdaavg = 1;

while (sum(abs(lamda - lamdaavg)) > 0.001)
powermat = [1 1 1; power; power.^2];
costfnpow = costfn * powermat;
costfnpow1 = diag(costfnpow)
sum(costfnpow1)
lamdamat = [1 2];
lamdamat = [lamdamat;lamdamat;lamdamat];
lamdamat = costfn(:, 2:3) .* lamdamat;
lamda = lamdamat * [[1 1 1 ; power]];
lamda = diag(lamda);
```

```
lamdaavgold = lamdaavg;
lamdaavg = mean(lamda);

pbest = power - (10*(lamda-lamdaavg)')
power = pbest;
ptotal1 = sum(pbest)
end
```

CHAPTER.5
RESULT

5.1. Results of ELD using PSO method

The results for the ELD problem with MPSO method is given below, figure 5.1 shows the input values given for generator's 1 and 2 and figure 5.2 represents the input of generator 3. The output of the ELD problem for a demand of 850 and 750 MW using MPSO method is shown in the figure 5.3 and 5.4 respectively. And from the results, it is shown that it requires only two iterations to solve the ELD problem using MPSO method.

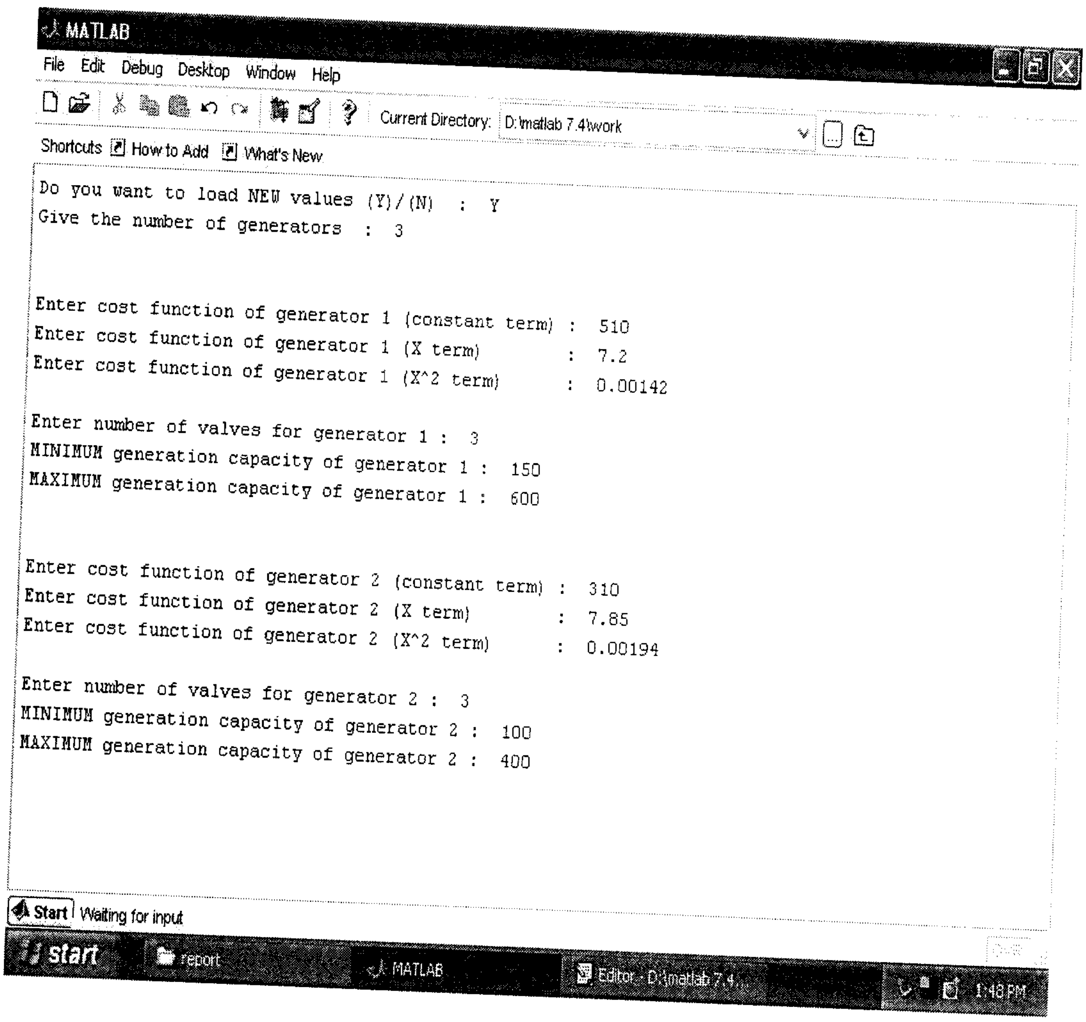

Figure 5.1 Inputs For Generate 1 and 2

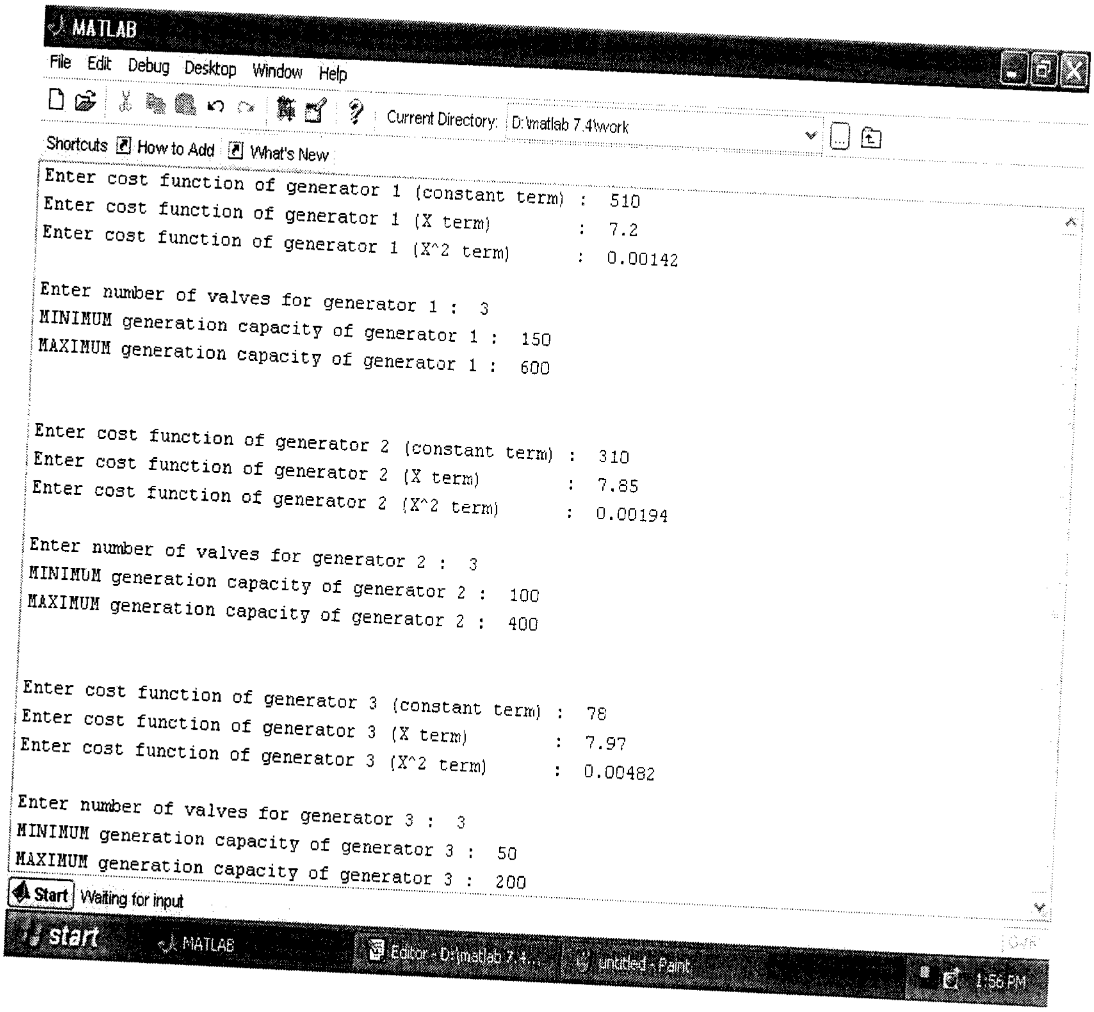

Figure 5.2 Input For Generate 3

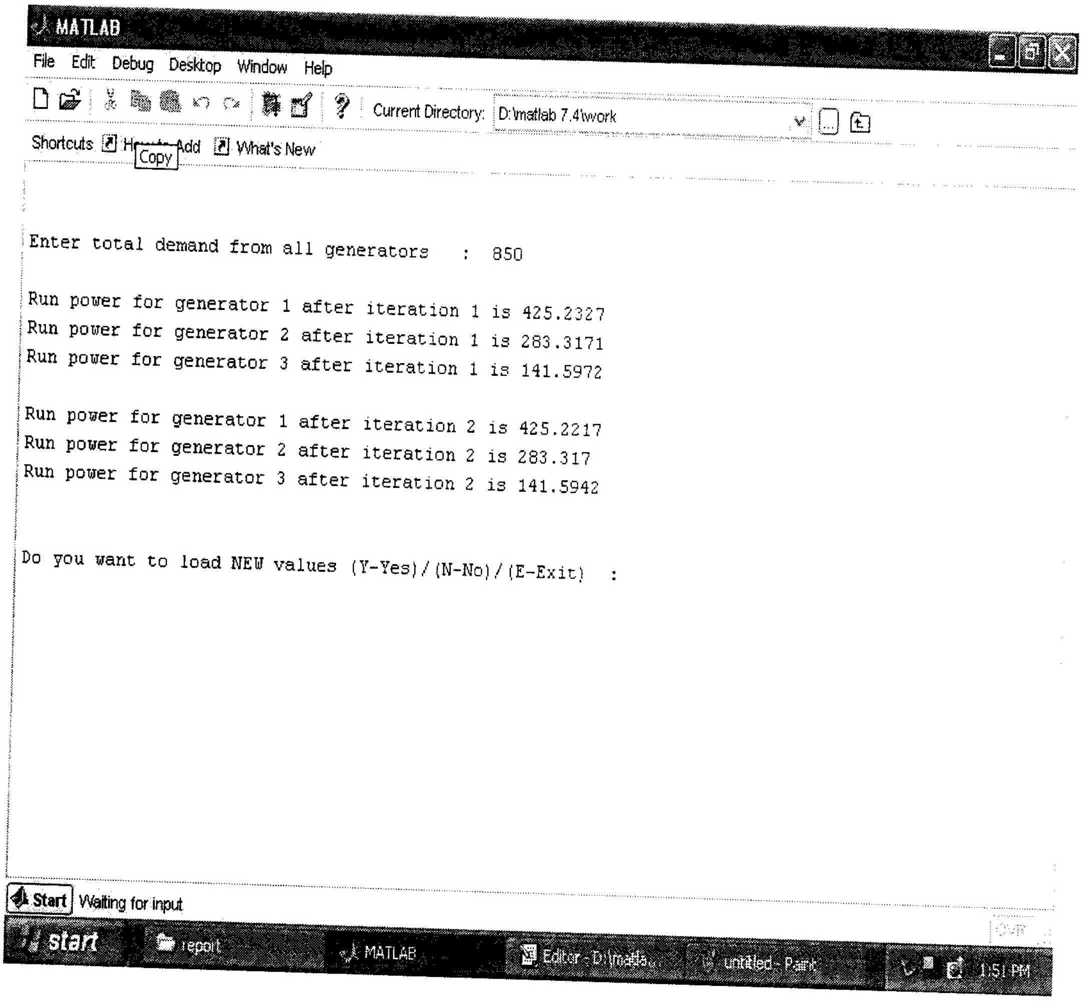

Figure 5.3 Output For The Demand Of 850 MW Using MPSO Method

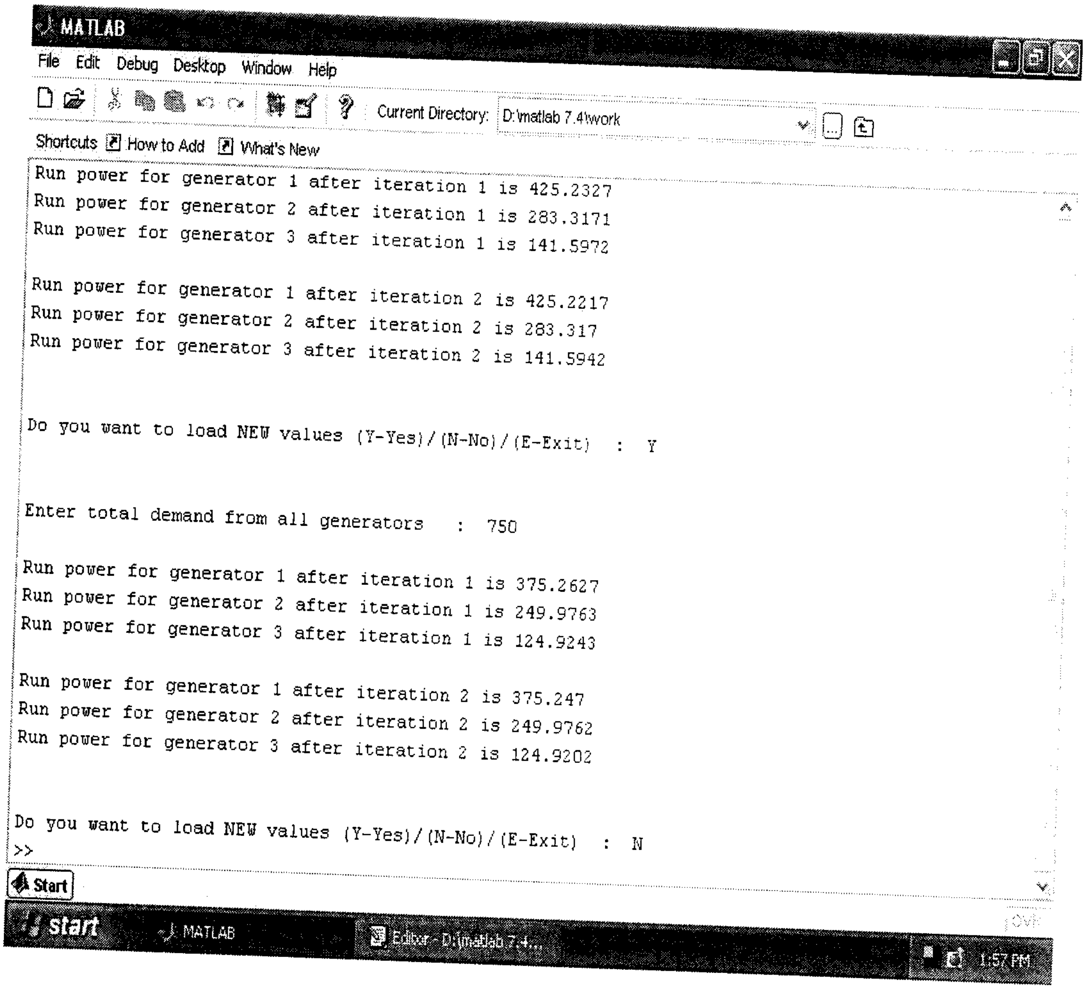

Figure 5.4 Output For The Demand Of 750 MW Using MPSO Method

5.2. Results of ELD using conventional method:

The result of the ELD problem using conventional method is shown below. Figure 5.5 shows the input values and the corresponding output it shown in the figure 5.6 and it is shown that it takes 108 iterations to solve the ELD problem.

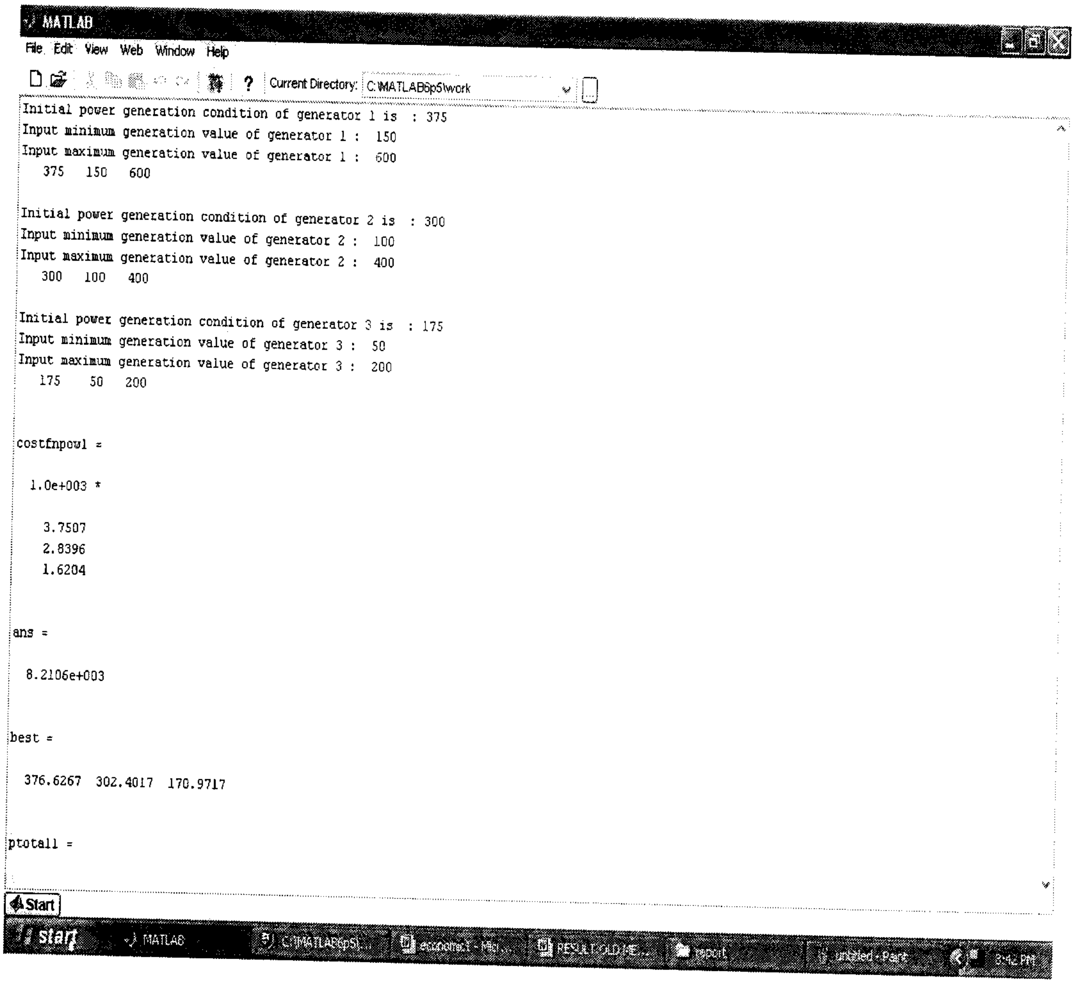

Figure 5.5 Input Data For The Conventional Method

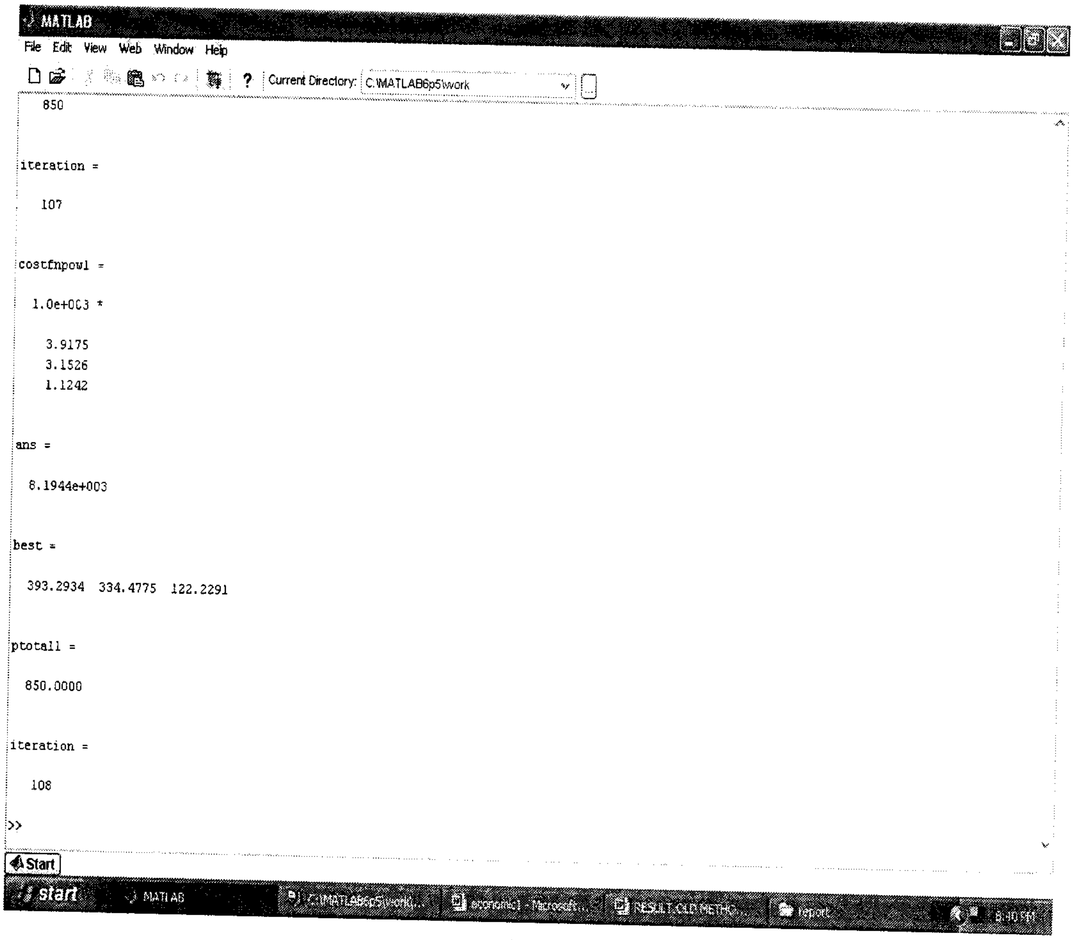

Figure 5.6 Output For The Demand Of 850 MW Using Conventional Method

Table 5.1 Comparative Results of ELD with Conventional and MPSO methods

			Conventional Method	**MPSO**
Demand (MW)			850	850
Output Power of Gen 1			393.2934	425.2217
Output Power of Gen 2			334.4775	283.317
Output Power of Gen 3			122.2291	141.5942
No of Iterations			108	2
Fuel Cost				
1	1	1	7838.2135	7821.24
0.9	0.9	0.9	7054.3863	7039.117
0.9	1.1	0.9	7684.9223	7577.067

CHAPTER.6
CONCLUSION

The issue of Economic Load Dispatch (ELD) for three generating stations with non-smooth cost functions, serving a specified load is solved using the PSO algorithm and is compared with the conventional method. From the simulation results, it is confirmed that the number of iterations required in this approach is less compared to the conventional methods. As the number of iterations is less, the system will attain its stability soon and also there is no overshoot and undershoot effects while using the PSO algorithm, hence if the algorithm is suitably applied, it will increase the efficiency of the generating systems.

REFERENCES

[1] A.J.Wood, B.J.Wollenberg, Power generation, operation and control. New York: Wiley, 1996.

[2] A Particle Swarm Optimization for EconomicDispatch With Nonsmooth Cost Functions Jong-Bae Park, *Member, IEEE,* Ki-Song Lee, Joong-Rin Shin, and Kwang Y. Lee, *Fellow, IEEE, IEEE transactions on power systems, VOL. 20, NO. 1, February 2005*

[3] Taguchi Method for Solving the Economic Dispatch Problem With Nonsmooth Cost Functions Derong Liu, *Fellow, IEEE,* and Ying Cai, *Student Member, IEEE,IEEE* Transactions on power systems, VOL. 20, NO. 4, November 2005, pp 2006-2014

[4] Economic Load Dispatch for Piecewise Quadratic Cost Function using Hybrid self-adaptive Differential Evolution with Augmented Lagrange Multiplier MethoThitithamrongchai, and B. Eua-Arporn, Member, IEEE,2006 International Conference on Power System Technology, pp 1-8

[5] Economic Power Dispatch with Non-Smooth Cost Functions Using Differential Evolution Raul E. Perez-Guerrero and Jose R. Cedenio-Maldonado, Member, IEEE *IEEE transactions on power systems-2005 pp 183-190.*

[6] H. T. Yang, P. C. Yang, and C. L. Huang, "Evolutionary programming based economic dispatch for units with nonsmooth fuel cost functions," *IEEE Trans. Power Syst.*, vol. 11, no. 1, Feb. 1996, pp. 112–118.

[7] C. E. Lin and G. L. Viviani, "Hierarchical economic dispatch for piecewise quadratic cost functions," IEEE Trans. Power App. Syst., vol. PAS-103, no. 6, June 1984, pp. 170–175.

[8] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc.IEEE Int. Conf. Neural Networks (ICNN'95)*, vol. IV, Perth, Australia, 1995, pp. 1942–1948.