Economic Load Dispatch of Generating Units with Multiple Fuel Options Using PSO

S. Agrawal[1], T. Bakshi[1] and D. Majumdar[2]

[1]Calcutta Institute of Technology, Uluberia, Howrah-712316
[2]Bengal College of Engineering & Technology, Durgapur

arrawaljsr@yahoo.co.in, tuli.bakshi@gmail.com, dipika.eee@gmail.com

Abstract

This paper presents a method to solve the optimal generation and dispatch of electrical power with multiple fuel options at different power levels. In general, the cost function for each generator is considered by a single quadratic function of power, for the optimal generation and dispatch problem. However, it is more realistic to represent the generation cost function for fossil fired plants by non smooth cost function i.e. the total generation cost function has non differentiable points such as segmented piece-wise quadratic function and valve point loading. Some generating units utilize multiple fuels sources (such as gas, coal, oil etc) are faced with the problem of determining which is the most economical fuel to be selected. This problem has been solved by Hota & Das [17] considering equal incremental cost. In this method the program has to be run for a number of times for fuel combinations for different units and then to select the best one out of all combinations. This is a complicated process. Also if some other type of nonlinear cost function is considered along with this, the method will not be applicable.

In view of the above mentioned problems of economic load dispatch (ELD) of multiple fuel generation, a more general heuristic method known as Particle Swarm Optimization (PSO) is considered in this paper to solve the problem in which either type of nonlinearity in cost function can be considered. For solution of this problem dynamic PSO is proposed. It is dynamic in the sense that the velocity bound or limit is updated in each iteration. This proposed method is applicable to the generating units that can use multiple fuels through valve at different generation levels as well as other problems which results in multiple intersecting continuous and discontinuous cost curves for any unit. The advantage of the method is that the method is applicable to both continuous and discontinuous cost curves of systems. Also it does not require selection of unit and type of fuel to be used after computation for a number of combination which is a cumbersome task as used in ref [17]. The complete method is explained and validated by taking an example. The simulation is carried out using MATLAB software. It has been shown that the method is simple, direct, and practical. It can be used for real time implementation and operation.

***Keywords**: Unit commitment, Optimal Load Dispatch, Piece-wise quadratic cost, Multiple Fuel Power Generation, PSO*

1. Introduction:

In early stage each unit was serving a limited local area. It has its own limitations such as reliability, spinning reserve capacity, security, quality of service. Keeping in view of these problems the concept of interconnection of different generating units forming an area where all the generating units are working in unison. In an area there are a number of generating units and each unit has a generating cost characteristic different from other unit. The problem arises that for a given load what should be the generation of each unit (operating under given constraints) so that the total cost involved is minimum. This problem is known as Economic Load Dispatch (ELD) problem. This is computed and controlled by a power system grid and the information is sent to all the generating units. The solution is found in standard text kooks [1-3] and reputed journals [4-5]. A complete survey is available in [6] to solve the problem using conventional techniques for a single area as well as multiple area power system. Various mathematical programming methods such as Linear Programming [7-8], non linear Programming [9], and Dynamic Programming [10-11] have been applied to solve this problem. Computational intelligence techniques based heuristic methods are also developed for solution of such problems e.g. Particle Swarm Optimization (PSO) [12], Genetic Algorithm(GA) [13] and Artificial Neural Networks (ANN) [14] etc.

In ELD problem generally the cost function of each generating unit is represented by a single quadratic function. It is more realistic if the cost function of each generating fossil fired unit is represented as a segmented piece-wise quadratic function [5], Such operating units are operated by multiple sources such as (i)gas or gases with different heat contents (ii) coal or different heat content of coal (iii) different types of oils. Hence the cost function of the valve operated generating units can be partitioned into different segments for multiple fuel operation. Each segment for multiple fuel operation the cost function is being associated with different types of fuels. The generating units which are supplied with multiple fuel sources (gas/oil/coal) are faced with the problem of determining which is the most economical fuel to be burnt. A method to solve ELD problem with piece-wise quadratic cost function has been suggested by Lin and Viviani [15]. The solution suggested by them is hierarchical using decentralized computation. It is a very complicated algorithm. This problem has also been solved by using ANN [16]. In this method a number of coefficients has to be chosen properly to represent ELD problem by neural network, which is a most difficult task. Hota and Das [17] suggested a method based on incremental cost basis (as in conventional methods). In this method the program has to be run for a number of times depending on the combinations of fuels for different units and then the best out of these values is selected. This is a time consuming, cumbersome and computationally complex process.

Keeping in view of above facts and problems, a heuristic optimization technique Particle Swarm Optimization (PSO) is proposed in this paper to solve this problem. The main advantage of this technique is that it gives a direct solution to this problem and is very much useful for on line operation. The solution procedure is illustrated by taking an example considered in ref. [17]. The computation is done using MATLAB software. In the next section the PSO technique is discussed in brief.

2. Particle Swarm Optimization (PSO)

It is a computational intelligence based optimization technique such as genetic algorithm (GA). It is a population based stochastic optimization technique developed by Kennedy and Eberhart in 1995 [18-21] and inspired by the social behavior of bird flocking in a group looking for food and fish schooling.

Some terms related to PSO:

The term PARTICLE refers to a member of population which is mass less and volume less m dimensional quantity. It can fly from one position to other in m dimensional search space with a velocity. For example for ELD problem of 3 machine systems each particle will have 3 dimensions representing generation of each machine (i.e. the dimension is same as the number of variables). POPULATION constitutes a number of such particles. The number of iteration for the solution of the problem is same as the number of generations in GA. The fitness function in PSO is same as the objective function for an optimization problem.

In real number space, each individual possible solution can be represented as a particle that moves through the problem space. The position of each particle is determined by the vector X_i and its movement by the velocity of the particle V_i represented in (1) and (2) respectively.

$$X_i^{k+1} = X_i^k + V_i^{k+1} \qquad (1)$$

The information available for each individual is based on

I) its own experience (the decisions it has made so far ,stored in memory)

II) the knowledge of performance of other individuals in its neighborhood.

Since the relative importance of these two information can vary from one decision to other, a random weight is applied to each part and the velocity is determined as in (2)

$$V_i^{k+1} = V_i^k + c1.rand1.\,(p_{best\,i}{}^k - X_i^k) + c_2.rand2.\,(g_{best}{}^k - X_i^k) \qquad (2)$$

Where, X_i^k = Position vector of a particle i = $[X_{i1}{}^k, X_{i2}{}^k, \ldots\ldots X_{im}{}^k]$ at kth iteration

V_i^K = Velocity vector of a particle i = $[V_{i1}{}^K, V_{i2}{}^K, \ldots\ldots V_{im}{}^K]$ at kth iteration

k = iteration count

$p_{best\,i}{}^k$ = i th particle has a memory of the best position in the search space at kth iteration . It is computed as $p_{best}\,i^{k+1} = X_i^{k+1}$ if the fitness function of ith particle at k+1 is less then (for minimum) the fitness function at kth iteration otherwise $p_{bst\,i}{}^{k+1} = p_{best\,i}{}^k$.

$g_{best}{}^k$ = It is that particle which has the minimum value of fitness function (for minimization) among all the particles in kth iteration.

c_1 & c_2 -= positive acceleration coefficients more then 1.0. Normally its value is taken $c_1 + c_2 = 4$ or $c_1 = c_2 = 2$.

rand1 & rand2 are random numbers between 0.0 & 1.0.

Both the velocity and positions have same units in this case MW.

The velocity update equation (2) has three components [22]

i) The first component is referred to “Inertia” or “Momentum”. It represents the tendency of the particle to continue in the same direction it has been traveling. This component can be scaled by a constant or dynamically changing in the case of modified PSO.

ii) The second component represents local attraction towards the best position of a given particle (whose corresponding fitness value is called the particles best (p_{best}) scaled by a random weight factor c_1.rand1. This component is referred as “Memory” or “Self knowledge”.

iii) The third component represents attraction towards the best position of any particle whose corresponding fitness value is called global best (g_{best}).scaled by another random weight c_2.rand2. This component is referred to “cooperation”,”social knowledge”, ”group knowledge” or “shared information”.

The PSO method is explained as above. The implementation of the algorithm is indicated below [23]:

i) Initialize the swarm by assigning a random position to each particle in the problem space as evenly as possible.

ii) Evaluate the fitness function of each particle.

i) For each individual particle, compare the particle's fitness value with its p_{best}. If the current value is better than the p_{best} value , then set this value as the p_{best} and the current particle's position X_i as $p_{best\ i}$.

ii) Identify the particle that has the best fitness value and corresponding position of the particle as g_{best}.

iii) Update the velocity and positions of all the particles using equations (1) & (2).

iv) Repeat steps i) to v) until a stopping criterion is met (e.g. maximum number of iterations or a sufficient good fitness value).

On implementation of PSO following considerations must be taken into account to facilitate the convergence and prevent an "explosion" (failure) of the swarm resulting in the variants of PSO.

i) Selection of Maximum velocity:

At each iteration step, the algorithm proceeds by adjusting the distance (velocity) that each particle moves in every dimension of problem space. The velocity of a particle is a stochastic variable and it may create an uncontrolled trajectory leading to "explosion". In order to damp these oscillations upper and lower limits of the velocity V_i is defined as

If $V_{id} > V_{i\ max}$ then $V_{id} = V_{i\ max}$

Else if $V_{id} < -V_{i\ max}$ then $V_{id} = -V_{i\ max}$

Most of the time, the value $V_{i\ max}$ is selected empirically depending on the characteristic of the problem. It is important it note that if the value of this parameter is too high, then the particle may move erratically, going beyond a good solution, on the other hand, if $V_{i\ max}$ is too small, then the particle movement is limited and it may not reach to optimal solution. The dynamically changing $V_{i\ max}$ can improve the performance given by

$V_{max} = (X_{max} – X_{min})/N$

Where X_{max} and X_{min} are maximum and minimum values of the found so far and N is the number of intervals.

ii) Selection of Acceleration Constants:

c_1 & c_2 are the acceleration constants; they control the movement of each particle towards its individual and global best positions. Small values limit the movement of the particles, while larger values may cause the particle to diverge. Normally the constants $c_1 + c_2$ limited to 4. If it is taken more than 4 the trajectory may diverge leading to "Explosion". In general a good start is when $c_1 = c_2 = 2$.

iii) Selection of Constriction Factor or Inertia Constant

Experimental study performed on PSO shows that even the maximum velocity and acceleration constants are correctly chosen, the particles trajectory may diverge leading to infinity, a phenomenon known as "Explosion" of the swarm. Two methods are to control this explosion (a) Inertia control and (b) Constriction factor control, the two variants of PSO.

(a) Inertia Constant

The velocity improvement represented by equation (2) is modified [24, 25, 26] and written as

$V_i^{k+1} = W.V_i^k + c_1.\text{rand1}. (p_{\text{best i}}{}^k – X_i^k) + c_2.\text{rand2}.(g_{\text{best}}{}^k – Xi^k)$ (3)

The first right hand side part (velocity of previous iteration) of equation (3) multiplied by a factor W is known as "Inertia Constant". It can be fixed or dynamically changing. It controls the "Explosion" of search space. Initially it is taken as high value (0.9) which finds the global neighborhood fast. Once it is found that it is decreasing gradually to o.4 in order to find narrow search as shown in equation (4)

W = wmax – (wmax – wmin)*itr/itrmax (4)

Where wmax = 0.9,wmin = 0.4,itrmax= maximum iterations, itr = current iteration

Since the weighting factor W is changing iteration wise it may be called as Dynamic PSO.

(b) Constriction Factor [27-28]

This is another method of control of "Explosion" of the swarm. The velocity in equation (2) is redefined using constriction factor developed by Clark and Kennedy [27], is represented in equation (5) as

$V_i^{k+1} = K*(V_i^k + c_1.\text{rand1}.(p_{\text{best i}}{}^k – X_i^K) + c_2.\text{rand2}.(g_{\text{best}}{}^k – X_i^K))$ (5)

Where K is known as constriction factor

K = 2/(abs(2 – c – sqrt(c^2 – 4*c)) (6)

Where, $c = c_1 + c_2 > 4.0$

Typically when this method is used, c is set to 4.1 and $c_1 = c_2 = 2.05$ the value of K comes out to be 0.729, In general, the constriction factor improves the convergence of the particle by damping oscillations. The main disadvantage is that the particles may follow wider cycles when $p_{\text{best i}}$ is far away from g_{best} .A survey is given in reference [29].The present problem is discussed in next section.

3. Problem Formulation

In some cases the modern power generating units are valve operated and multiple fuels can be used in these units depending on power levels required. The generating cost of such fossil fired units are represented by piece-wise quadratic cost function. Such types of cost function are called hybrid cost function as shown in Figure 1.

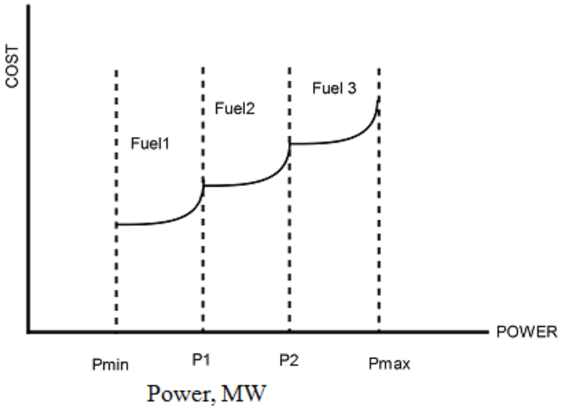


Figure 1. Multiple Cost Curve

The notations used for representing the cost of ith unit for kth fuel used is as indicated below (It is assumed that m is the number of generating units and f is the number of fuels which can be used):

i) P_{min} is the minimum power generation limit of ith unit with fuel 1.

ii) P_1 is the maximum power generation limit of ith unit with fuel 1.

iii) P_{f-1} is the minimum power generation limit of ith unit with fuel f..

iv) P_{max} is the maximum power generation limit of ith unit with fuel f.

iv) Subscript i indicates generating unit number and subscript k indicates type of fuel.

vi) a_{ik} , b_{ik} and c_{ik} are cost curve coefficients of ith generating unit with fuel k.

v) cost (P_i) is the hybrid cost function of ith unit.

$$\text{Cost}(P_i) = \begin{array}{l} a_{i1} + b_{i1}P_{i1} + c_{i1}P_{i1}^2 \\ P_{min} \le P_{i1} \le P_1 \quad \text{for fuel (1)} \\ a_{i2} + b_{i2}P_{i2} + c_{i2}P_{i2}^2 \\ P_1 \le P_{i2} \le P_2 \quad \text{for fuel (2)} \\ . \\ a_{ik} + b_{ik}P_{ik} + c_{ik}P_{ik}^2 \\ P_{k-1} \le P_{ik} \le P_k \quad \text{for fuel (k)} \\ . \\ . \\ a_{if} + b_{if}P_{if} + c_{if}P_{if}^2 \\ P_{f-1} \le P_{if} \le P_{max} \quad \text{for fuel (f)} \end{array} \qquad (7)$$

Let the total power demand of the area is PD. Then the power balance equation assuming lossless transmission line written as

$$\sum_{i=1}^{m} P_{ik} - PD = 0 \tag{8}$$

Total power generation cost to meet the load demand is given by

$$J = \sum cost(P_i) \tag{9}$$

Now the ELD problem for multiple fuel options with given hybrid piece-wise quadratic cost curve can be stated as:

"Given the hybrid cost characteristics and inequality constraints for each fuel used in generation for each unit as represented by equation (7) and the power balance (equality constraints) represented by equation (8) , it is required to find power generation of each unit and type of fuel for each unit to meet the total power demand such that the total cost of generation given by equation (9) is minimum."

It is attempted to solve this problem using PSO technique as discussed in next section.

4. Solution Procedure

The optimization problem for an interconnected multiple area power system is stated in the previous section. There are a number of methods exists for solution of such problems using analytical methods such as linear programming, non linear programming, gradient method, integer programming and so on, and heuristic methods such as GA,ANN, and others. In this section we will try to solve this problem using PSO and its variants. It is a heuristic iterative method revolving around two equations velocity (distance) and position updating given by the two equations (1) & (2). Some time the method may land up with "Explosion". To prevent Explosion some variations of PSO are there as discussed in previous section as given in equations (3) & (4) and (5) & (6). The details of solution procedure are given below.

Initialization

To start the iteration process initialization of individual particle position and velocity is necessary. At first a population is selected i.e. the number of particles n the population will have. Then each particle's position is selected randomly wide spread in the search space such that it does not violate the constraints represented by equations (7) & (8) for ith particle as $X_i = (X_{i1}^0, X_{i2}^0, \ldots\ldots.., X_{im}^0)$ in m dimensional space where m is the number of variables. In this case m is the number of generating units and the generation cost depends on f number of fuels that can be used for each unit. The initial velocity of ith particle $V_i = (V_{i1}^0 ,V_{i2}^0 ,\ldots\ldots.,V_{im}^0)$ is created. Keeping in view that the velocity components chosen must satisfy the inequality constraints i.e. they should lie within the maximum and minimum velocity limit ($-V_{i\,min} < V_{id} < V_{i\,max}$) where

$$V_{i\,max} = (X_{id\,max} – X_{id\,min})/ N \tag{10}$$
$$V_{i\,min} = - V_{i\,max} \tag{11}$$

where N is iteration number and suffix d is the dimension.

The initial value of p_{best} of individual i is set as the initial position of individual i i.e. initial value of p_{best} is same as the initial value of position. The initial value of g_{best} is obtained by finding the fitness function of all the particles and select the particle which has optimum value.

Updating the Position and Velocity of Particles

In order to modify the position of each particle, it is necessary to calculate the velocity of each particle in the next stage (iteration / generation).The velocity is modified first using equation (2) or its variants equation (3) or (5). Then the position is modified using (1). The modified position of each individual particle may not satisfy the equality and inequality constraints equations (8) and (7) respectively.

If the position of particles crosses its limiting value (generation is beyond its limits i.e. less then minimum value or more then maximum value), it is adjusted first satisfying inequality constraint equation (7) as shown below in equation (12).

$$X_{id}^{k+1} = \begin{cases} X_{id}^{k} + V_{id}^{k} & \text{if } X_{id\,min} < X_{id}^{k} + V_{id}^{k} < X_{id\,max} \\ X_{id\,min} & \text{if } X_{id}^{k} + V_{id}^{k+1} < X_{id\,min} \\ X_{id\,max} & \text{if } X_{id}^{k} + V_{id}^{k+1} > X_{id\,max} \end{cases} \tag{12}$$

In addition to the inequality constraints as in (7),the equality constraint given by (8) has to be satisfied for implementing the PSO algorithm in each iteration. To satisfy the equality constraints a heuristic method is proposed as given below.

i) Find the sum of the variables of a particle (sum of generations of each machine in this case).

ii) Compare it with the equality constraint and find the difference.

iii) The difference is divided by the number of variables and then adds this value to each element of the particle.

iv) Repeat (i) to (iii) till it is satisfied or remains within certain limit.

Updating p_{best} and g_{best}

The $p_{best\,i}$ of each individual particle at each iteration k+1 is updated as follows:

$$P_{best\,id}^{k+1} = X_{id}^{k+1} \quad \text{if} \quad J_i^{k+1} < J_i^{k} \tag{13}$$

$$P_{best\,id}^{k+1} = p_{best\,id}^{k} \quad \text{if} \quad J_i^{k+1} > J_i^{k} \tag{14}$$

Where, J_i^k is the objective function / fitness function evaluated at the position of individual particle at iteration k , X_{id}^{k+1} is the position of the particle i at iteration k+1 $p_{best\,id}^{k+1}$ is the best position of the individual particle i until iteration k+1.

Equation(13) & (14) compares the p_{best} of every individual particle with its current fitness value. If the new position value of an individual particle has better performance then the current $p_{best\,i}$, the $p_{best\,i}$ is replaced by new position. If the new position has lower performance then the current p_{best} value remains unchanged. The g_{best} the global best position at iteration k+1 is set as the best evaluated position among all $p_{best\,i}$ (among all particles).

The Stopping Criterion

The proposed iterative method is terminated if the iteration approaches a predefined criterion, usually a sufficient good fitness or if it reaches the maximum number of iterations as defined. At the end when the optimum value of the problem is reached, all the particles will have the same value of P_{best} as the value of g_{best} .

The problem stated in previous section can be solved using PSO method as explained in this section. Normally this method gives solutions to all types of problems it may be linear, continuous, discontinuous or nonlinear problems. In the next section the method is explained by taking an example and simulation is done by using MATLAB software.

The complete solution procedure step-wise is present as below:

Steps for solution:

(i) Find out the dimensions of the problem which is equal to the number of generating units.

(ii) Choose the population i.e. the number of particles.

(iii) Initialize the position of the swarm keeping in view of the constraints (equality and inequality)

(iv) Select the minimum and maximum velocity limit of the particles.

(v) Initialize the velocity of the swarm keeping in view of the velocity limit.

(vi) Select the initial value of P_{best} as the initial value of the position of the swarm.

(vii) Find the fitness function of each particle and then find G_{best} , which is the minimum of all particles.

(viii) Update the velocity using equation (2).

(ix) Test for velocity limit.

(x) Update the position using equation (1).

(xi) Test for equality and inequality constraints for position.

(xii) Find the fitness function.

(xiii) Find P_{best} using equation (13 & 14).

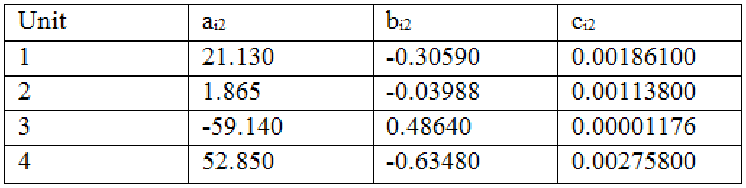
(xiv) Find G_{best} .

(xv) Test for the termination of iteration process if satisfied stop the iterations, otherwise go to step (viii).

5. Simulation and Result

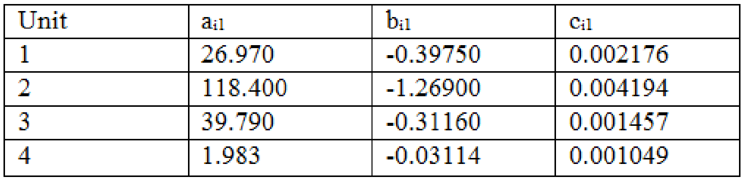
The complete solution procedure for the problem stated has been explained by taking an example from Ref. [17]. In this problem there are four generating units. *It is assumed that the* first generating unit uses fuel type 1 and type 2 where as the other three units can use all the three types of fuels. The data’s are given in tabular form. Table 1 represents cost coefficients of different units for different types of fuels a_{ik}, b_{ik} and c_{ik}. The suffix i stand for unit number and suffix k stands for fuel number. The power generation limit of each unit depending on the type of fuel is given in Table 2. To meet the total load demand PD, each generating unit (out of four units) has to generate power such that the total generating cost is minimum. Hence there are four variables, one variable corresponding to each generating unit. For implementation of PSO for the said ELD problem (with multiple fuel options) a certain number of particles each having the same dimension as the number of variables (known as Population) has to be considered. The number of particles (Population) assumed for this problem is five as per the guide lines that the population assumed should be a bit higher then or equal to the number of variables so that the problem solution time and computational complexity is reduced as discussed in ref. [30]. So the swarm consists of five particles and each particle has four dimension in this problem. As discussed in solution procedure the position of a particle X_{id} is the power generation of ith particle by dth unit (P_{id}). The problem is to find the generation and type of fuel used of each unit for a total load demand (PD) of 915 MW such that the total cost of generation is minimum.

Table 1. (a) Fuel 1



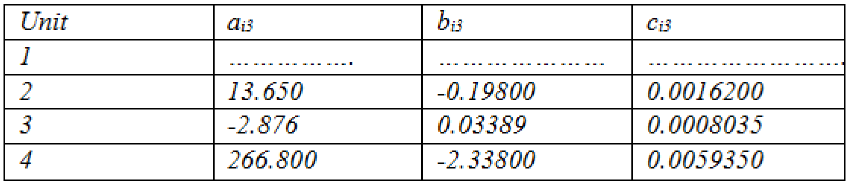
Unit	a_{i2}	b_{i2}	c_{i2}
1	21.130	-0.30590	0.00186100
2	1.865	-0.03988	0.00113800
3	-59.140	0.48640	0.00001176
4	52.850	-0.63480	0.00275800

Table 1. (b) Fuel 2



Unit	a_{i1}	b_{i1}	c_{i1}
1	26.970	-0.39750	0.002176
2	118.400	-1.26900	0.004194
3	39.790	-0.31160	0.001457
4	1.983	-0.03114	0.001049

Table 1. (c) Fuel 3



Unit	a_{i3}	b_{i3}	c_{i3}
1	…………….	…………………	………………………
2	*13.650*	*-0.19800*	*0.0016200*
3	*-2.876*	*0.03389*	*0.0008035*
4	*266.800*	*-2.33800*	*0.0059350*

Table 2. Generation Limits (MW) of Units Corresponding to Different Fuel Operations

Unit	Fuel (1), MW		Fuel (2), MW		Fuel (3), MW	
Number	P_{min}	P_1	P_1	P_2	P_2	P_{max}
1	100	196	196	250	…….	……..
2	157	230	50	114	114	157
3	200	332	388	500	332	388
4	99	138	138	200	200	265

The maximum ($P_{\max i}$) and minimum ($P_{\min i}$) the power generating limits of each unit is found out from Table-2 as indicated below:

$P_{\max 1} = 250$ MW, $P_{\min 1} = 100$ MW
$P_{\max 2} = 230$ MW, $P_{\min 2} = 50$ MW
$P_{\max 3} = 500$ MW, $P_{\min 3} = 200$ MW
$P_{\max 4} = 265$ MW, $P_{\min 4} = 99$ MW

The initial value of position i.e. power generation of each unit has been selected such that it does not violet the equality and inequality constraints given by equations (8,7) respectively. For clarity again represented as

$$\sum_{d=1}^{4} P_{id} = PD \quad (8) \quad \text{and} \quad P_{\min d} \le P_{id} \le P_{\max d} \quad (7)$$

Where suffix i stands for particle number and d stands for dimension. Based on the above constraints the initial position of the swarm is selected as:

$$X^0 = P^0 = \begin{matrix} 110 & 140 & 405 & 260 \\ 150 & 100 & 465 & 200 \\ 245 & 200 & 350 & 120 \\ 105 & 130 & 450 & 230 \\ 165 & 80 & 475 & 195 \end{matrix}$$

The velocity constraints/limits at starting is assumed (at random) to be maximum limit as $V_{\max} = 100$ and minimum limit as $V_{\min} = -100$ uniform velocity throughout all dimensions. The initial velocity at the starting of iteration is assumed to be (within the above mentioned limits) :

$$V^0 = \begin{matrix} 10 & 20 & -20 & -30 \\ -50 & -60 & 20 & 40 \\ 100 & 90 & -50 & -10 \\ 40 & 50 & 60 & 70 \\ 40 & 10 & 50 & 50 \end{matrix}$$

As mentioned in the solution procedure the initial value of P_{best} is selected as the initial value of position X^0. A function program is developed using MATLAB software to compute the generation cost of each unit depending on the the fuel used (cost coefficients given in Table-1) which also depends on power level of the unit as given in Table-2. Then the fitness function or the cost function is obtained by summing up the generating cost of each unit of a particle. A function program is developed for finding p_{best} using equations (13) & (14) and also a function program is developed for g_{best} . The function program is written for updating the velocity using equation (2) .Then it is tested if the velocity is within the limits by a function program. After testing for velocity limits, the position is updated by using equation (1) using a function program developed. Lastly the updated position is tested for equality and inequality constraints using equations (7), (8) and (12) , through a function program developed.

The iterative procedure is adopted for solution. The solution is tried for three cases of velocity update as represented in equations (2), (3) and (5) for fixed value of velocity limit i.e. $V_{\max} = 100$ and $V_{\min} = -100$. The solution was tried for iterations ranging up to 500. The generations of each unit was always changing and the generation cost was oscillating between 180 to 195. It was not stabilized.

Then the solution was tried for dynamically changing the limiting value of the velocity uniformly in all dimensions using equations (10) & (11) reproducing here for ready reference as :

$$V_{max} = (X_{id\ max} – X_{id\ min})/N \tag{10}$$

and $$V_{min} = -V_{max} \tag{11}$$

On implementing this dynamically changing velocity limit iteration-wise, the generation of each unit and the total cost stabilizes in about 100 iterations. Further it was tried up to 200 iterations in steps of 10 iterations, the values of generation of each unit and the total cost does not change. It shows that the optimum value is reached which is indicated below

Table 3

	Present Results		Results of ref. [17]	
Unit No.	Generation	Fuel	Generation	Fuel
1	206.45	2	206.63	2
2	206.64	1	206.51	1
3	265.91	1	265.88	1
4	236.00	3	235.99	3
	Total Cost 178.0957		Total Cost 178.0954	

The results of the stated problem using PSO obtained in this computation and the results by using conventional technique explained in ref. [17] are tabulated in table-3. It shows that the two results are approximately similar. Only there is very small variation of 0.0001% in cost of the two results which is negligible. The computation time taken for 100 iterations is approximately 2.5 seconds. Also it can be further reduced by improving the function program for testing equality constraints. As the computation time small this method is also suitable for on line implementation. This method has wider application as it can accommodate any other type of non-linearity other than quadratic , where as ref [17] procedure is valid only for quadratic cost function. The next section discusses comments and conclusions.

6. Comments and Conclusions

An Economic Load Dispatch problem with multiple fuel options has been considered in this paper. PSO an iterative stochastic technique has been applied to solve this problem. The complete solution procedure has been discussed in section 5 and it has been explained by taking an example from ref. [17] in section 6. Initially the solution was tried with fixed velocity limit in each direction but it was found that the solution was not converging to an optimal point. It was tried up to 500 iterations but the solution (cost of generation) was oscillating between 180 to 190. It appears that due to the large velocity limit the particles were flying under and above the optimal point. In view of above problem , the solution was tried with dynamically varying the velocity limit (converging) equally in each dimension for each particle as suggested in equations (10,11). Under this condition solution found converged to an optimal point in about 100 iterations. The computation time is about 2.5 seconds. The computation time can further be reduced by improving the function program for meeting the equality constraints. With such a low computation time ,the real time implementation is not a problem. The results of the present method is compared with the results of ref. [17]. It was found out that the variation is about 0.0001%i.e. so to say negligible. The main advantage of the present method is that in this method other type of non linearity can be accommodated where as in the method of ref. [17] the solution is not valid for any other type of non linearity other then the quadratic cost function. Hence this method is more general (superior) then the method of ref. [17].

Acknowledgements

The authors are thankful to the management of "Calcutta Institute of Technology, Uluberia, Howrah, India" for providing all required facilities to complete this work.

References

[1] O. I. Elgared, "Electric Power System Theory, An Introduction", Second Edition TMH, **(1983)**.

[2] D. P. Kothari and I. J. Nagrath, "Modern Power System Analysis", Third Edition, TMH, New Delhi, **(2003)**.

[3] A. J. Wood and B. F. Woolenberg, "Power Generation, Operation and Control", John Wiley & Sons, New York, **(1984)**.

[4] A. K. Ayub and A. D. Patton, "Optimal Thermal generating Unit Commitment", IEEE Trans. **(1971)**, PAS 90.

[5] IEEE Committee Report, "Present Practices in the Economic Operation of Power System", IEEE Trans, **(1971)**, PAS 90.

[6] H. H. Happ, "Optimal Power Dispatch – a Comprehensive Survey", IEEE Trans., **(1977)**, PAS 96.

[7] A, Farag, S, Al-Baiyat and T, Cheng, "Economic Load Dispatch Multi Objective Optimization Procedures Using Linear Programming Technique", IEEE Trans. Power Syst., vol. 10, no. 2, **(1995)** May, pp. 731-738.

[8] R. Jabr, A. Coonick and B. Cory, "A Homogeneous Linear Programming Algorithm For the Security Constrained Economic Dispatch Problem", IEEE Trans. Power Syst., vol. 15, no. 3, **(2000)** August, pp. 930-936.

[9] S. Virmani and E. Adrian, "Implementation of a Lagrangian Relaxation based Unit Commitment Problem", IEEE Trans., vol. 4, **(1999)** November, pp. 1373-1380.

[10] Z. Oyang and S. Shahidehpour, "An Intelligent Dynamic Programming for Unit Commitment Application", IEEE Trans. Power Syst., vol. 6, no. 3, **(1991)** August, pp. 1203-1209.

[11] W. Hobbs and G. Hermon, "An Enhanced Dynamic Programming Approach for Unit Commitment", IEEE Trans. Power Syst., vol. 3, no. 3, **(1988)** August, pp. 1201-1205.

[12] M. Sudhakaran, "Particle Swarm Optimization for Economic and Emission Dispatch Problem", Institute of Engineers, India, vol. 88, **(2007)** June, pp. 39-45.

[13] Barkirtzis, V. Petridis and S. Kazarlis, "Genetic algorithm Solution to the Economic Dispatch Problem", IEE Proceedings of Generation Transmission, Distribution, vol. 141, no. 4, **(1994)** July, pp. 377-382.

[14] S. S. Kumar and V. Palanisamy, "A Fast computation Hopfield Neural NetWork Method of Unit Commitment", Institute of Engineers India, vol. 88, **(2007)** June, pp. 3-9

[15] C. E. Lin and Viviani, "Hirarchical Economic Dispatch for Piece-wise Quadratic Cost Function", IEEE Trans. On PAS, vol. 103, no. 6, **(1984)** June, pp. 1170-1175.

[16] J. H. Park, "Economic Load Dispatch for Piece-wise Quadratic Cost Function using Hopfield Neural Networks", IEEE Trans. On PAS, vol. 8, no. 3, **(1993)** August, pp. 1030-1038.

[17] P. K. Hota and S. K. Das, "Economic Load Dispatching of Generating Units with Multiple Fuel Options", Institution of Engineers India, vol. 88, **(2007)** June, pp. 22-28.

[18] Y. Shi and R. C. Eberhart, "Particle Swarm Optimization: Developments, Applications And Resources", Proceedings of the 2001 Congress on Evolutionary Computation, vol. 1, **(2001)**, pp. 81-86.

[19] R. C. Eberhart and Y. Shi, "Comparing Inertia Weights and Constriction Factor in Particle Swarm Optimization", Proceedings of the 2000 Congress on Evolutionary Computation, vol. 1, **(2000)**, pp. 84-88.

[20] J. Kennedy and R. Eberhart, "Particle Swarm Optimization", Proc. Int. Conf. NeuRal Networks (ICNN), vol. 4, **(1995)**, pp. 1942-1948.

[21] R. Eberhart and J. Kennedy, "A New Optimizer Using Particle Swarm Theory", Proc. 6th Int. Symp. Micro Machine and Human Science (MHS), **(1995)** October, pp. 39-43.

[22] D. Boiringer and D. Werner, "Particle Swarm Optimization versus Genetic Algorithms for Phase Array Synthesis", IEEE Trans. Antennas Propagat., vol. 52, no. 3, **(2004)** March, pp. 771-779.

[23] Y. Shi, "Feature Article on Particle Swarm Optimization", IEEE Neural Network Society, Feature Article, **(2004)** February, pp. 8-13.

[24] Y. Shi and R. Eberhart, “A Modified Particle Swarm Optimization”, Proc. IEEE World Cong. Comput. Intell., **(1998)** May, pp. 69-73.
[25] Y. Shi and R. Eberhart, “Empirical Study of Particle Swarm Optimization”, Proc. IEEE Cong. Evol. Comput., vol. 3, **(1999)** July, pp. 1945-1950.
[26] A. Ahmad, N. Singh and T. Varshney, “A New Approach for Solving Economic Load Dispatch Problem”, MIT International Journal of Electrical and Instrumentation Engineering, vol. 1, no. 2, **(2011)** August, pp. 93-98.
[27] M. Clerc and J. Kennedy, “The Particle Swarm Explosion, Stability and Convergence in a multidimensional Complex Space”, IEEE Trans. Evol. Comput., vol. 6, no. 1, **(2002)** February, pp. 58-73.
[28] S. Y. Lim, M. Montakhab and H. Nouri, “Economic Dispatch of Power System Using Particle Swarm Optimization With Constriction Factor”, International Journal of Innovations in Energy Systems and Power, vol. 4, no. 2, (2009) October.
[29] Y. del Valle, “Particle Swarm Optimization: Basic Concepts, Variants And Applications in Power Systems”, IEEE Trans. on Evolutionary Computation, vol. 12, no. 2, **(2008)** April.
[30] S. Agrawal, “Economic Load Dispatch & Tie-Line Power Flow for a Multiple Area Power System through PSO”, Communicated.

Authors

S. Agrawal. B.Tech in 1966 and M.Tech in 1979 from N.I.T. Jamshedpur,India in the field of Electrical Engineering. P.hD.from I.I.T. Delhi, India in 1982 in the field of Power System Operation and Control. I have guided M.Tech discertation of about 14 students. A number of papers published in the field. Currently I am working in the field of Power System Operation and Control using heuristic methods such as Particle Swarm Optimization (PSO**).**

T. Bakshi. M. Sc. in Applied Mathematics, MCA and M.Tech in Information Technology. Currently pursuing P.hD. research work.

D. Majumdar. B.Tech in 2009 and M. Tech in 2012 from WBUT, India. I am working in the field of Power System Operation and Control using heuristic methods such as Particle Swarm Optimization (PSO**).**