Optimal Power Pooling for a Multiple Area Power System through PSO

Shaligram Agrawal[1], Tuli Bakshi[2], and Dipika Majumdar[3]

[1] Deptt. of EE & CSE, Calcutta Institute of Technology, Howrah, India
[2] Deptt. Of MCA, Calcutta Institute of Technology, Howrah, India
[3] Deptt. EEE, Bengal College of Engineering, Durgapur, India

[1]agrawaljsr@yahoo.co.in, [2]tuli.bakshi@gmail.com, [3]deepika.eee@gmail.com

Abstract: In a multiple area power system power can be transferred from one area to other to improve the load factor, reliability, security and economics of the power system. The generation cost of each unit in an area and the cost of power transfer from other area are given. For a given load in an area the problem is how much power to be generated internally by all the generating units in the area (unit wise) and how power has to be transferred (pooled) from other area for a given total load demand in the area (for economic operation of the power system). This problem has been solved using Lagrange multiplier method recently. The limitation of this method is that it is applicable only if the generation cost of each generating unit is quadratic. Sometimes the power generation cost of each unit is not quadratic but is other type of nonlinear function e.g. the valve-point effect problem it is addition of quadratic and sinusoidal function. The main objective of this paper is to overcome this limitation by solving this general type of nonlinear optimal problem using a Meta heuristic method, PSO (Particle Swarm Optimization) and its variants. It is to point out that for implementation of PSO it is required to select a population size. Generally the population size is selected on ad-hoc basis. This effects the computation time as well as number of iteration for solution. A method has been suggested to select population size on the basis of optimal computation time as second objective. The method is explained by an example. The results obtained by PSO and its variants are compared among themselves and with the results obtained by analytical method. MATLAB 7 software is used for computation.

Keywords: Integrated Power System, Optimization, PSO, Economic Power Dispatch.

1. Introduction

Electric power system is the largest man made system in the world. It consists of synchronous generators, transformers, transmission lines, active and reactive power controllers, relays and switches. For proper operation and control of such a large nonlinear no stationary system along with operational constraints requires solution of an optimization problem for a given objective. In a simple form it can be defined as:

For a given objective function f(x) (x as variable) find the optimum value of x under the given inequality constraints $h(x) \leq 0$ and equality constraints $g(x) = 0$.

The electric power is generated by different modes of generation such as fossil fired generation, using diesel oil, petrol, nuclear fuels, hydro generation and non-conventional energy sources such as wind power, tidal waves, solar energy and biogas etc. The generation cost of each method different depending on the fuel cost and method of generation. As for example the operating cost of coal fired (thermal) , diesel, petrol units are higher due to the cost of fuel is higher in comparison to the hydro generating units or wind power generation whereas the fixed cost of hydro unit or wind power units are higher. The constraints of the different units are also different such as the availability of water is a constraint in case of hydro generation.

Received: November 8th, 2013. Accepted: May 16th, 2014

In early stage each unit were serving a limited local area. It has got its own limitations such as reliability, quality of service, security, spinning reserve capacity. Keeping in view of these problems the concept of interconnection of different generating units forming an area (region) developed where all the generating units are working in unison. In an area there are a number of generating units, each has different cost characteristic. 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 known as Economic Dispatch (ED) problem. This is computed and controlled by a power system grid and the information is sent to different generating units. The solution of this problem is found in standard books [1-2] and reputed journals [3-5]. Various mathematical programming methods has been applied to solve this problem such as LP (Linear Programming) [6-7] where all the constraints & objective function are linear, Non Linear Programming (NLP) [8], Dynamic Programming (DP) [9-10]. The DP is suffering from the problem of "curse of dimensionality". Computational intelligence based techniques/ heuristic methods are also developed for solution of such problems e.g. Particle Swarm Optimization (PSO) [11&24], Genetic Algorithm (GA) [12], Artificial Neural Networks (ANN) [13] etc.

The load curve of different area of power system is of different shape. The peak load of different area occurs at different time in a day. The load can be transferred from one area to other, depending on the power generating capacity and load demand of each area and their difference, through tie line between the two areas resulting in a co-ordinated operation of the power system. It improves the load factor, reduces the spinning reserve of each area, and improves the power system security, reliability and power quality. Recently a paper has been published [14] considering power transfer from one area having a number of generating units to other on the basis of multiple flat rates (cost) depending on the quantity of power transfer. This results in a 2^{nd} order smooth nonlinear optimization problem. The solution method proposed was a conventional method of Lagrange multiplier.

In this paper it is proposed to solve the above problem using PSO a heuristic method and its variants and the results are compared. The advantage of the method is that it is applicable to both linear, smooth/non-smooth nonlinear/piece wise linear systems. The computations are carried out considering different population sizes and terminating the algorithm for a given number of iterations. The results of these methods and the result obtained using conventional method ref. [14] are also compared. It has been shown that the results obtained by PSO are approximately same as obtained by conventional method. On the basis of computation it has also been shown that as the population (number of particles) increases the number of iterations (generation) decreases for the solution but the CPU time of the computation increases after a particular population size. The optimum population size is suggested for minimum CPU time. If the number of particles (Population) is less than the dimension (number of variables) of the optimization problem it may give erratic result, and if the population size is too large the CPU time will be large. So it has been suggested that population should be somewhat more than the number of variables.

The method is explained by taking an example having two generating machines in an area interconnected to another area by tie line. It is assumed that the generation cost curve of each machine and the power generating limits are given. The results obtained by PSO and its variants are compared with the conventional one obtained in ref. [14]. All the computations are carried out using MATLAB- 7 software. In the next section a PSO and its variants are 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 [15-18] 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 ED 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.

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

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.

The movement of particles iteration to iteration will depend on the importance of the above two information. A random weight is applied to each part and the velocity is determined as in (2)

$$V_i^{k+1} = V_i^k + c_1.\text{rand1}.\ (p_{best\ i}^{\ k} - X_i^k) + c_2.\text{rand2}.\ (g_{best}^{\ k} - X_i^k) \quad (2)$$

Where, X_i^k (Position vector of a particle i) = $[X_{i1}^k, X_{i2}^k, \ldots\ldots X_{im}^k]$ at k^{th} iteration
V_i^k (Velocity vector of a particle i) = $[V_{i1}^K, V_{i2}^K, \ldots\ldots V_{im}^K]$ at k^{th} iteration
k = iteration count and m = dimension (number of variables).

$p_{best\ i}^{\ k}$ = i th particle has a memory of the best position in the search space at k^{th} iteration . It is computed as $p_{best\ i}^{\ k+1} = X_i^{k+1}$ if the fitness function of i^{th} particle at k+1 iteration is less then (for minimum) the fitness function at k^{th} iteration other wise $p_{bst\ i}^{\ k+1} = p_{best\ i}^{\ k}$. $g_{best}^{\ k}$ is that particle which has the minimum value of fitness function (for minimization) among all the particles in k^{th} 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 [19]

(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 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 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 implementation of the PSO method (algorithm) explained above is indicated below:

(i). Initialize the swarm by assigning a random position to each particle in the problem space as evenly as possible satisfying all equality and inequality constraints.
(ii). Evaluate the fitness function of each particle.

(iii). For each individual particle, compare the particle's fitness value with its p_{best}. If the current value is better than the past value of p_{best} , then set this value as the current $p_{best\ i}$ and the current particle's position X_i as $p_{best\ i}$.
(iv). Identify the particle that has the best fitness value among all particles and corresponding position of the particle as g_{best}.
(v). Update the velocity and positions of all the particles using equations (1) & (2).
(vi). Repeat steps i) to v) until a stopping criterion is met (e.g. maximum number of iterations or any other stopping criterion defined).

For 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 $V_{id} > V_{max}$ then $V_{id} = V_{max}$, Else if $V_{id} < -V_{max}$ then $V_{id} = -V_{max}$
Generally the value of V_{max} is selected empirically.
(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 "Explosion" of the swarm. Two methods are suggested to control this explosion (a) Inertia constant control and (b) Constriction factor control, the two variants of PSO.

[a] Inertia Constant Control: The velocity improvement represented by equation (2) is modified [21, 22, 26] and written as

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

W is known as Inertia Constant. It can be fixed or dynamically changing as given by (4) known as Dynamic PSO.

$$W = Wmax - (Wmax - Wmin)*itr/itrmax \quad (4)$$

Where Wmax = 0.9,Wmin = 0.4,itrmax= maximum iterations, itr = current iteration

[b] Constriction Factor: This is another method to control "Explosion" of the swarm. The velocity in equation (2) is redefined using constriction factor developed by Clark and Kennedy [23], is represented in equation (5) as

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

Where K is known as constriction factor and given by (6)

$$K = 2/(abs(2 - c - sqrt(c\hat{}2 - 4*c)) \quad (6)$$

Where, $c = c_1 + c_2 > 4.0$

A survey is given in reference [25] and a featured article in [20]. The present problem is discussed in next section.

3. Problem Statement

A two area power system interconnected through tie line represented in Figure A. In area 1 there are 'N' generating units interconnected through loss less power lines is assumed and it is interconnected through tie-line to other area. The power borrowed from area 2 is represented by P_t. It is also assumed that there is always power available in area 2 to be transferred as and when required and power transmission loss is negligible. It is assumed that the power generating cost (operating cost) of each unit depends upon its own generation and is represented as

$$CP_i = a_i + b_i.P_i + c_i.P_i^2 \quad (7)$$

Where a_i, b_i and c_i are constants depending on each unit, P_i is power generation of i^{th} unit and CP_i is the operating cost of i^{th} unit. However The operating cost represented in eq. (7) may contain additional smooth/non-smooth nonlinear terms e.g. valve-point effect and change of fuel.

It is assumed that power can also be borrowed from other area through tie line may be during peak load hours or otherwise when it is required for economic operation of composite power system, It is further assumed that the cost of borrowed power from other area is on the flat rate basis. Let the maximum power that can be borrowed is P_{tmax} and the cost of power transfer is on two slabs, where the cost per unit of power transfer from 0 to P_{t1} is λ_1 and P_{t1} to P_{tmax} is λ_2. The tie line power cost can be represented as in (8).

$$\begin{aligned} C_{tie} &= \lambda_1.P_t && \text{for } P_t < P_{t1} \\ C_{tie} &= \lambda_1.P_{t1} + \lambda_2.(P_t - P_{t1}) && \text{for } P_{t1} < P_t < P_{tmax} \end{aligned} \quad (8)$$

The total cost of internal generation and tie line power is given by

$$C_{total} = \sum_{i=1}^{N} CP_i + C_{tie} \quad (9)$$

Now the problem can be stated as:

For a given load Pd find the power generation of each unit (Pi) and tie line power (Pt) borrowed from other area, such that the total cost given by (9) is minimum subject to the following constraints:

(i). Equality constraints

$$\sum_{i=1}^{N} Pi + P_t = Pd \quad (10)$$

(ii). Inequality Constraints

The power generation of each unit is limited by maximum power P_{max} and minimum power P_{min} as represented in (11)

$$Pmin\ i < Pi < Pmax\ i \quad (11)$$

This problem has been solved by using LaGrange method [14]. Lagrange multiplier method is applicable only for quadratic type of smooth nonlinear optimization problem, whereas the PSO method is applicable to a general type of smooth /non-smooth nonlinear optimization.

Hence in this paper it is proposed to solve this problem using PSO method 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, nonlinear programming, gradient method, integer programming and so on, and intelligence based methods/heuristic methods such as GA, ANN, and others. In this section it is proposed 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). Sometime the method may land up with "Explosion". To prevent Explosion some variations of PSO are used as discussed in previous section given in equations (3) & (4) and (5) & (6). The detail of solution procedure is presented below.

Initialization: To start the iteration process initialization of individual particle position and velocity is necessary. At first a population size (the number of particles the population will have) n is selected. Next the number of variables m is chosen. Then each particle's initial position is selected randomly wide spread in the search space such that it does not violate the constraints represented by equations (10) & (11) for i^{th} particle as $X_i = (X_{i1}^0, X_{i2}^0, \ldots\ldots, X_{im}^0)$ in m dimensional space. In this case m-1 variables are the generation of generating units and the last variable is the tie line power. The initial velocity of i^{th} particle $V_i = (V_{i1}^0, V_{i2}^0, \ldots\ldots, V_{im}^0)$ is selected such that it does not violate the velocity constraints ($-V_{min} < V_{id} < V_{max}$). 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 computed after computing the fitness function of all the particles and then selecting the particle which has optimum value.

Updating 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 (10) and (11) respectively.

If the position of particles crosses its limiting value (generation is beyond its limits), it is adjusted first satisfying inequality constraint equation (11) as shown below in equation (12).

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

In each iteration, the equality constraints (10) has to be satisfied in addition to the inequality constraints (11). 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 variable of the particle.

Updating p_{best} and g_{best} : The $p_{best\,i}$ of each individual particle at each iteration k = i is updated as follows:

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

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

Where, F_i^k is the objective function / fitness function (C_{total}) 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 $p_{best\ i}$ 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 otherwise the current $p_{best\ i}$ 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. In this case stopping criterion is selected as predefined maximum number of iterations.
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. An algorithm is presented in appendix-1 representing the complete solution procedure. In the following section the complete solution procedure is explained by taking an example of a power system.

5. Example and Result

The example considered here for explaining the procedure is taken from ref, [14] as:

A two area power system connected through tie line is considered. Area 1 has 2 generating units having cost characteristics as given by equation (7).

$$CP_1 = 120 + 40.P_1 + 0.1.P_1^2$$
$$CP_2 = 100 + 32.P_2 + 0.125.P_2^2 \tag{7}$$

The tie line power cost is assumed to be in two slabs having per unit cost of Rs 60 ($\lambda 1$) for first 50 MW (P_{t1}) and Rs 65 (λ_2) per MW up to next 50 MW. So the maximum tie line power that can be borrowed is assumed to be 100 MW. The power generation limit of each generating unit is assumed as per equation (11).

$$\text{Unit 1: 5 MW } (P_{min\ 1}) < P_1 < 150 \text{ MW } (P_{max1})$$
$$\text{Unit 2: 5MW } (P_{min2}) < P_2 < 150 \text{ MW } (P_{max2}) \tag{11}$$
$$\text{Tie-Line power } P_t,\ 0 < P_t < 100 \text{ MW}$$

The objective function/ fitness function is given by the total cost

$$C_{total} = CP_1 + CP_2 + C_{tie} \quad (C_{tie} \text{ is the cost of tie-line power}) \tag{9}$$

$$C_{tie} = \lambda_1.P_t \quad \text{for } P_t < 50 \text{ MW}$$
$$= \lambda_1.P_{t1} + \lambda_2.(P_t - P_{t1}) \quad \text{for } 50 < P_t < 100 \text{ MW} \tag{8}$$

The problem could be stated, for a given load P_d find the generation of each unit and tie - line flow such that the cost-function given by equation (9) is minimum and the inequality constraints given by equation (11)and equality constraint given by (10) are satisfied.

Assumed the values of c_1 & c_2 as $c_1 = 2$ & $c_2 = 2$for simple PSO and Dynamic PSO and for Constriction PSO the values of c_1 & c_2 as $c_1 = 2.05$ & $c_2 = 2.05$and K comes out to be 0,729. For dynamic PSO the maximum & minimum weights considered are 0.9 & 0.4 respectively. The computation has been carried out for two values of maximum number of iterations 50 and

100. The initial value of p_{best} is taken as the initial value of position. The total load demand is assumed to be 232 MW. It is also assumed that the particle velocity is bounded between $-P_d/2$ to $P_d/2$. For meeting the equality constraints in each iteration after updating the position, the difference between the given P_d and the total computed generation and tie-line power is equally divided among each dimension i.e. in this case the difference is divided by 3 and this value is added to the three dimensions (P_1, P_2 and P_t). Other criteria may also be assumed.

For implementation of PSO & its variants 5 sets of population is considered having 1st set having 2 particles, 2nd set consisting of 3 particles, 3rd set consisting of 4 particles, 4th set consisting of 6 particles and 5th set consisting of 8 particles. The initial values of position and the velocity vectors are assumed as per guide lines proposed earlier Tables 1 to 5 as indicated below:

Table 1. Set-1 (2 Particles)

Prt	P_1	P_2	P_t	V_1	V_2	V_t
1	50	120	62	-50	60	20
2	130	80	22	70	-70	40

Table 2. Set-2 (3 Particles)

Prt	P_1	P_2	P_t	V_1	V_2	V_t
1	50	120	62	-50	60	20
2	130	80	22	70	-70	40
3	80	100	52	10	50	-20

Table 3. Set-3 (4 Particles)

Prt	P_1	P_2	P_t	V_1	V_2	V_t
1	50	120	62	-50	60	20
2	130	80	22	70	-70	40
3	80	100	52	10	50	-20
4	110	60	62	30	40	50

Table 4. Set-4 (6 Particles)

Prt	P_1	P_2	P_t	V_1	V_2	V_t
1	50	120	62	-50	60	20
2	130	80	22	70	-70	40
3	80	100	52	10	50	-20
4	110	60	62	30	40	50
5	90	110	32	-10	20	40
6	70	90	72	-30	40	-30

Table 5. Set-5 (8 Particles)

Prt	P_1	P_2	P_t	V_1	V_2	V_t
1	50	120	62	-50	60	20
2	130	80	22	70	-70	40
3	80	100	52	10	50	-20
4	110	60	62	30	40	50
5	90	110	32	-10	20	40
6	70	90	72	-30	40	-30
7	40	105	87	20	10	60
8	112	100	20	40	-20	-10

A program has been developed in METLAB-7 for the solution following the steps indicated in previous section "Solution Procedure ". The solution is obtained for the considered 5-sets and the results are tabulated in tabular form as indicated below.

The results obtained in this method are compared with the results of Ref. [14] as indicated here. For a load demand of 232 MW the generations of the two units are as P_1 = 100 MW, P_2 = 112 MW, tie line power is P_{tie} = 20 MW and the minimum cost = Rs. 11572. The computation time i.e. CPU time is 5.9 seconds.

The solution using the present method has been obtained for 5- Sets , Set -1 consisting 2 particles, Set – 2 consisting 3 particles, Set – 3 consisting 4 particles, Set – 4 consisting 6 particles and Set – 5 consisting 8 particles, for maximum number of 50 iterations and 100 iterations. For 50 iterations, it does not give very encouraging results. Only constriction PSO method stabilizes for population having four and above number of particles or members. Other two methods Simple PSO and Dynamic PSO do not give any fruitful results. Hence, the results obtained for 100 iterations by the three methods (Simple PSO, Dynamic PSO and constriction PSO) are presented here in tabular form and some plots are drawn as indicated below:

Table 6. Result of Set-1, Population-2

Method	P_1	P_2	P_t	P_{total}	*CPU*	*Cost*	*iteration*	*Comment*
K	92	119.6	20.3	232	6.64	11586	70	Not stable
D	102.9	113.7	15.3	232	6.63	11573	70	Not stable
P	100.5	107.7	23.8	232	6.64	11574	60	Not stable

Table 7. Result of Set-2, Population-3

Method	P_1	P_2	P_t	P_{total}	*CPU*	*Cost*	*iteration*	*Comment*
K	100	112	20	232	6.75	11572	30	Stable
D	102.4	115.9	13.7	232	6.7	11572	75	Stabilizing
P	99.9	111.7	20.3	232	6.67	11572	100	Stabilizing

Table 8. Result of Set-3, Population-4

Method	P_1	P_2	P_t	P_{total}	*CPU*	*Cost*	*iteration*	*Comment*
K	100	112	20	232	6.61	11572	21	Stable
D	100	112	20	232	6.64	11572	36	Stable
P	100	111.8	20.2	232	6.69	11572	30	Not Stable

Table 9. Result of Set-4, Population-6

Method	P_1	P_2	P_t	P_{total}	*CPU*	*Cost*	*iteration*	*Comment*
K	100	112	20	232	6.62	11572	18	Stable
D	100	112	20	232	6.73	11572	65	Stable
P	111	121	0	232	6.66	11594	10	Not Stable

Table 10. Result of Set-5, Population-8

Method	P_1	P_2	P_t	P_{total}	*CPU*	*Cost*	*iteration*	*Comment*
K	112	112	20	232	6.72	11572	22	Stable
D	112	112	20	232	6.76	11572	46	Stable
P	112.4	112.4	19.6	232	6.73	11572	25	Not Stable

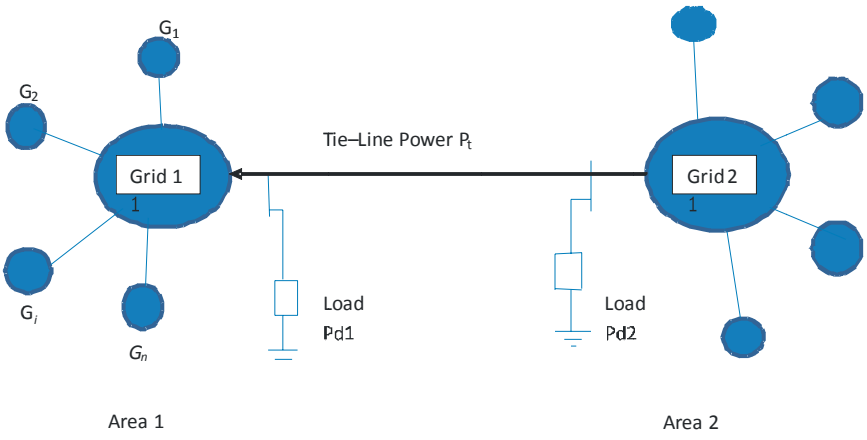

Figure A. Two area Power System

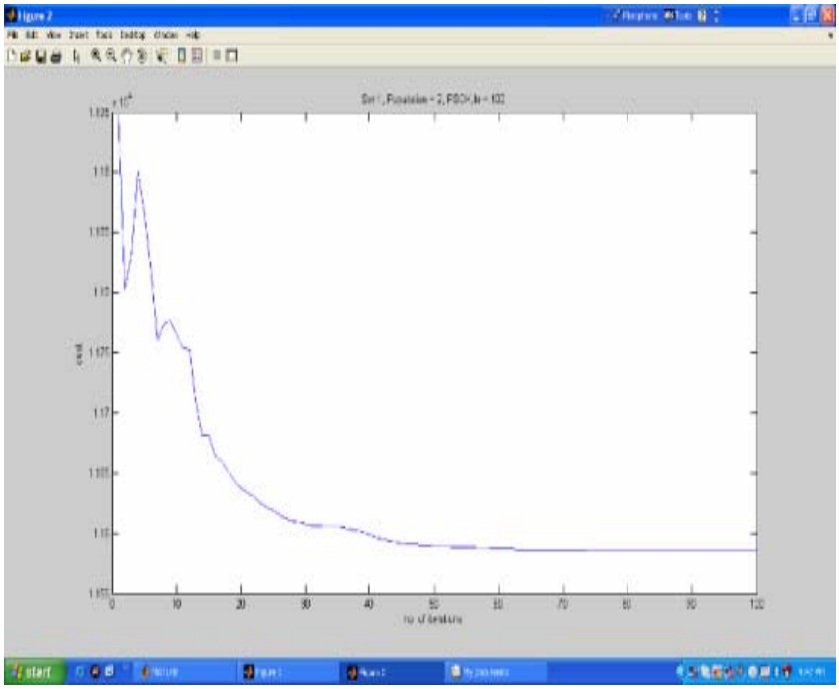

Figure 1. Cost Vs No. of Iterations population 2(K-PSO)

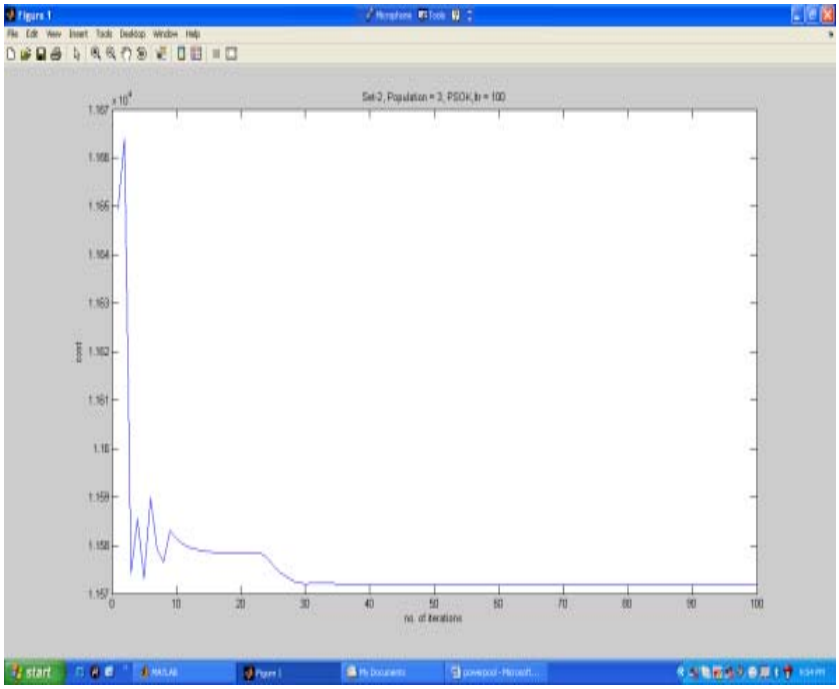

Figure 2. Cost Vs No. of Iterations population 3(K-PSO)

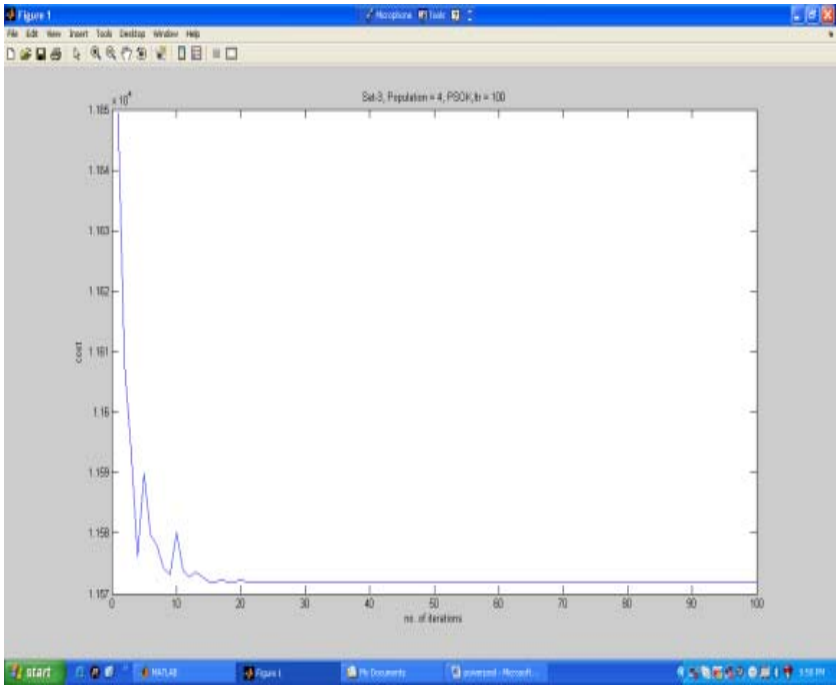

Figure 3. Cost Vs No. of Iterations population 4(K-PSO)

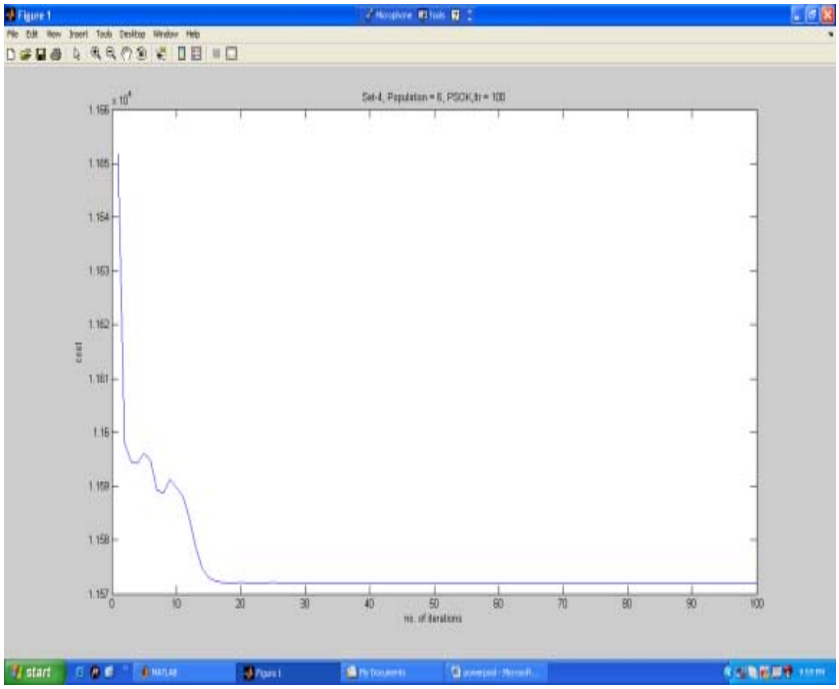

Figure 4. Cost Vs No. of Iterations population 6(K- PSO)

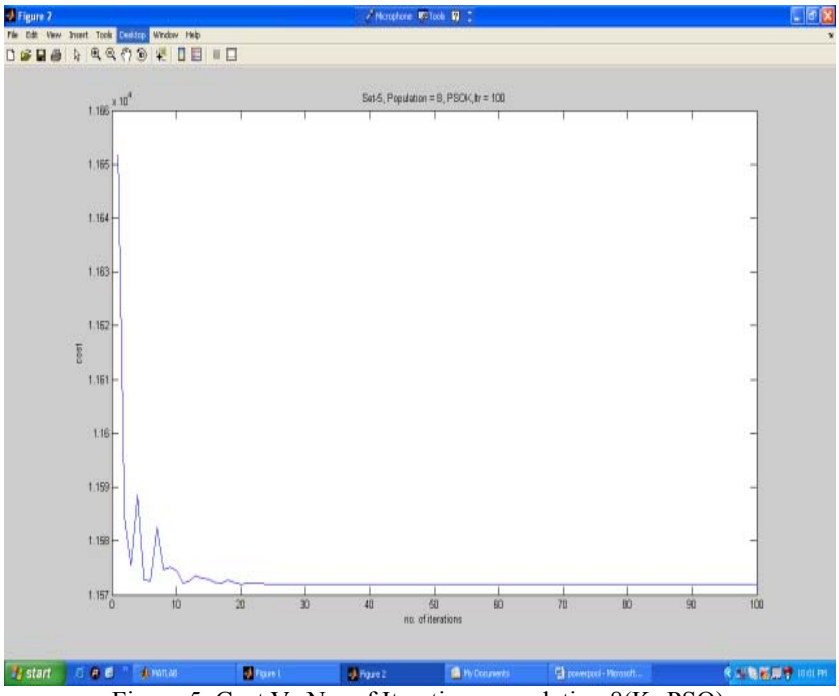

Figure 5. Cost Vs No. of Iterations population 8(K- PSO)

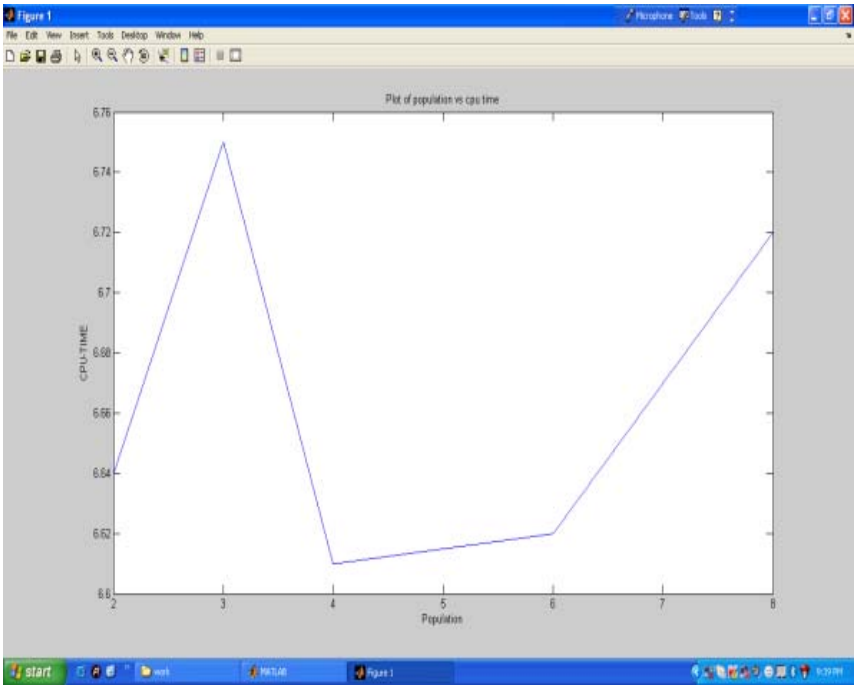

Figure 6. CPU time Vs No of population (K-PSO)

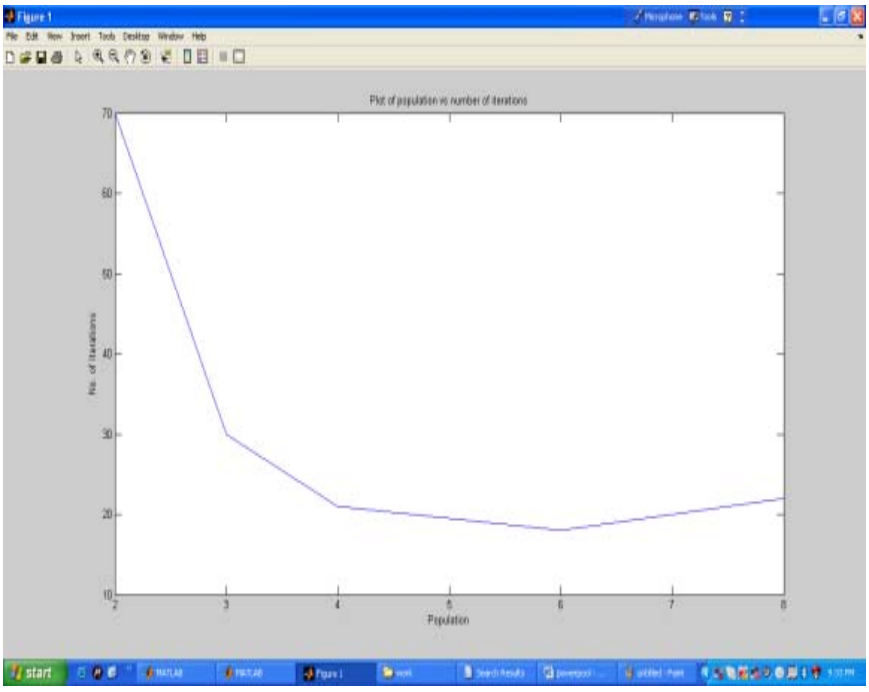

Figure 7. No. of iterations (stab) Vs No.of popu (K- PSO)

The results presented above are compared and the observations drawn are as indicated Below;

[a] It is observed from tables 6 to 10 and other computations (not shown) that Simple PSO is not suitable for this problem as it does not give required result.

[b] The computation is said to be stabilized when each particle of the population will have same value in each dimension i.e. P_1,P_2 & P_t will be same for each particle and also the value of cost function will be same for each particle.

[c] Experience has shown that the number of iterations/ CPU time may differ from run to run for same initial value of the variables assumed, as the solution procedure is stochastic in nature. But the final stabilized values of the variables will be same after stabilization of the process

[d] It is clear from the table 6 that if the population size is less then the number of variables the process may not stabilize or even if it is stabilized it may fall in local minima not a global one (shown in Figure1) which has also been tested by carrying out computation for 150 iterations.

[e] The tables 8 to 10 show that constriction PSO method gives better results then other two methods. However, Dynamic PSO also gives similar results but CPU time is a bit higher then Constriction method.

[f] Figure 2 to 5 shows that initially as the population size increases the number of iterations for stabilization decreases. It becomes almost flat after some population size as shown in Figure 7.

[g] The plot CPU time vs. population size Figure 6 shows that at first the CPU time will be decreasing as the population increases and reaches minimum for a particular population size and then increases as population size increases. In this case it is 4 or may be 5. As the number of variables is 3, the population size may be chosen a bit higher than the number of variables of the optimization problem.

Result: From the above observations it can be said that the population size should be selected somewhat higher than the number of variables for optimum implementation of PSO as shown in fig 6. The Constriction PSO gives better result (minimum CPU time). The results are compared with the results of ref. [14]. The optimum cost and the variables i.e. the generations of the two generating units and the tie-line power pooling are same in both the cases. The CPU time in λ iteration method is 5.9 sec where as in PSO it is 6.61 sec a bit higher than conventional method. However, the added advantage of PSO method is that it is applicable to all types of nonlinear systems whereas the λ iteration method is applicable to only quadratic type of nonlinear system. Since the computation time is low, the method can be implemented on-line. The overall comments and conclusions are presented in next

6. Conclusions and Comments

Conclusions: The complete problem is solved by PSO and its variants, and the results obtained are shown in tabular form tables 6 to 10 and graphical form plots fig 1 to Figure7. The results are compared with the results obtained by analytical method Ref. [14]. It has been shown that the two results i.e. the power sharing among the machines, borrowed tie line power from other areas for a given total load in the area and optimum cost , are same obtained by both the methods analytical & PSO. The computation is carried out by PSO and its variants and has been shown (Tables 7 to 10) that the Constriction PSO gives better results in this problem. The PSO method is superior to Lagrange multiplier method as it is applicable to general type of nonlinear optimization problems.

Normally nothing is available in the literature regarding the selection of population size to be chosen. In this project 5-sets of population size is considered having i) 2- particles, ii) 3-particles, iii) 4-particles, iv) 6-particles,v) 8-particles with 3-dimensions (generations of two machines and 3rd tie line power). It has been shown (Table 6) that if the population selected is less than the dimension of the problem, the method normally does not give acceptable results.

As the population size increases, the number of iterations decreases (as shown in Figure7) for the solution. On increasing the population size up to a particular value the computation time decreases but if the population is increased further the computation time also increases as shown in Figure 6. As such it has been suggested that the population size selected should be somewhat higher than the dimension of the problem (as shown in Figure6).

Comments: It has been observed that at the optimum the cost of each particle is same and at the same time the value of each variable will also be same. It is to point out that since the method of solution is stochastic in nature, the iterations/ CPU time for stabilization may vary from run to run for the same assumed initial value of variables of the problem. But the final result will be same. Regarding the selection of the limits of velocity not much is available in the literature, in this example it is assumed as $P_d/2$ to $-P_d/2$. However a dynamic method can be applied for selecting velocity as suggested earlier. For meeting the equality constraints the difference between the specified value and the computed value is divided by the number of dimensions of the problem and this value is added to each dimension. Other method may be suggested or tried. The method can be implemented on-line.

7. Acknowledgement

The authors are thankful to the management of "Calcutta Institute of Technology, Uluberia, Howrah, India" for providing all required facilities to complete this work. Also thanks to Mr. Ramesh Samanta for helping in preparation of the menu script.

Appendix – 1

Algorithm for implementation of PSO

[a] Find out the dimension of the problem i.e. number of variables
[b] Choose the population i.e. the number of particles.
[c] Initialize the position of the swarm keeping in view of the constraints (equality and inequality).
[d] Select the minimum and maximum velocity limit of the particle.
[e] Initialize the velocity of the swarm keeping in view of the velocity limit.
[f] Select the iteration stopping criterion.
[g] Select the initial value of P_{best} as the initial value of the position of the swarm.
[h] Find the fitness function of each particle and then find g_{best} which is the minimum of all particles.
[i] Update the velocity using equation (2)/(3)/(5).
[j] Test for velocity limit.
[k] Update the position using equation (1).
[l] Test for equality and inequality constraints for position.
[m] Find the fitness function (C_{total}).
[n] Find P_{best} as follow:
if $P_{best\ id}^{k+1} = X_{id}^{k+1}$ if $C_{total\ i}^{k+1} < C_{total\ i}^{k}$
$P_{best\ id}^{k+1} = P_{best\ id}^{k}$ $C_{total\ i}^{k+1} > C_{Ttotali}^{k}$
[o] Find g_{best} the global best position at iteration k+1 is set as the best evaluated position among all Pbest i (among all particles).
[p] Repeat from step (ix) to (xv) till the stopping criterion is reached.

8. Reference

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

[4] H. H. Happ, "Optimal Power Dispatch – a Comprehensive Survey", *IEEE Trans. PAS* 96, 1977.
[5] IEEE Committee Report, " Present Practices in the Economic Operation of Power System", *IEEE Trans*. PAS90, 1971.
[6] 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, pp 731 – 738, May 1995.
[7] 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, pp. 930-936, Aug 2000.
[8] S. Virmani, E. Adrian et.al. "Implementation of a Lagrangian Relaxation based Unit Commitment Problem", *IEEE Trans*., Vol. 4, pp1373-1380, Nov.-1999.
[9] Z. Oyang and S. Shahidehpour, "An Intelligent Dynamic Programming for Unit Commitment Application", *IEEE Trans. Power Syst*., Vol-6, No. 3, pp 1203-1209, Aug. 1991.
[10] W. Hobbs, G. Hermon et.al. "An Enhanced Dynamic Programming Approach for Unit Commitment ", *IEEE Trans. Power Syst.* ,Vol. 3, No. 3, pp 1201-1205, Aug. 1988.
[11] M. Sudhakaran et.al. "Particle Swarm Optimization for Economic and Emission Dispatch Problem", Institute of Engineers, India, Vol. 88, pp. 39-45, June 2007.
[12] Barkirtzis, V. Petridis and S. Kazarlis, "Genetic algorithm Solution to the Econo -mic Dispatch Problem", *IEE Proceedings of Generation Transmission, Distribution* Vol. 141, No. 4, July 1994, pp. 377-382.
[13] S. Senthil Kumar and V. Palanisamy, "A Fast computation Hopfield Neural Net-Work Method of Unit Commitment', Institute of Engineers India, Vol. 88, pp. 3-9 June 2007.
[14] S. Agrawal, S. K. Das and Dipika Mazumdar, "A Novel Approach to Unit commit-Ment and Tie Line Power Flow for multiple area Power System", *MIT International Journal of Electrical and Instrumentation Engineering*, Vol. 1, No. 2, Aug. 2011, pp. 70-75.
[15] Y. Shi and R.C. Eberhart," Particle Swarm Optimization: Developments, Applications And Resources", *Proceedings of the 2001 Congress on Evolutionary Computation*, Vol. 1, pp. 81-86, 2001.
[16] 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, pp. 84-88, 2000.
[17] J. Kennedy and R. Eberhart, "Particle Swarm Optimization ", Proc. Int. Conf. Neu-Ral Networks (ICNN), Nov. 1995, Vol. 4, pp. 1942-1948.
[18] R, Eberhart and J. Kennedy, "A New Optimizer Using Particle Swarm Theory", *Proc. 6th Int. Symp. Micro Machine and Human Science (MHS),* Oct. 1995, pp. 39-43.
[19] D. Boiringer and D. Werner, "Particle Swarm Optimization versus Genetic Algo-Rithms for Phase Array Synthesis", *IEEE Trans. Antennas Propagat*. Vol. 52, No. 3, pp. 771-779, Mar. 2004.
[20] Y. Shi, "Feature Article on Particle Swarm Optimization ", *IEEE Neural Network Society, Feature Article*, pp. 8-13, Feb. 2004.
[21] Y. Shi and R. Eberhart, "A Modified Particle Swarm Optimization", Proc. IEEE World Cong. Comput. Intell., May 1998, pp. 69-73.
[22] Y. Shi and R. Eberhart, "Empirical Study of Particle Swarm Optimization", *Proc. IEEE Cong. Evol. Comput*. July 1999, Vol. 3, pp. 1945-1950.
[23] M. Clerc and J. Kennedy, "The Particle Swarm Explosion, Stability and Conver-Gence in a multidimensional Complex Space", *IEEE Trans. Evol. Comput*. Vol. 6, No. 1, pp. 58-73, Feb. 2002.

[24] Shi Yao Lim, M. Montakhab and Hassan 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, Oct. 2009.
[25] Yamille del Valle et.al. "Particle Swarm Optimization : Basic Concepts, Variants and Applications in Power Systems", *IEEE Trans. on Evolutionary Computation*, Vol. 12, No. 2, April 2008.
[26] Anish Ahmad, Nitin Singh and Tarun Varshney, "A New Approach for Solving Economic Load Dispatch Problem", *MIT International Journal of Electrical and Instrumentation Engineering* Vol. 1, No. 2, Aug. 2011.

Shaligram Agrawal was born in 1943. He received B. Tech. and M. Tech. in Electrical Engineering from NIT Jamshedpur under Ranchi University, India in 1966 and 1978 respectively. He completed Ph. D. from I.I.T. Delhi, India in 1982 in the field of Power System Operation and Control. He has guided a number of M. Tech. dissertation projects and published a number of papers in national/international seminars/ conferences/journals. Currently He is working in the field of Power System Operation and Control using heuristic methods such as Particle Swarm Optimization (PSO). He is guiding Ph. D. research scholars.

Tuli Bakshi is born in 1979. She completed M. Sc. in Applied Mathematics and MCA. She has done M.Tech in Information Technology from Jadavpur University, Howrah, India in 2009. Her field of interest is Optimization Techniques, Artificial Intelligence and Meta heuristic Approach. Currently she is pursuing her Ph. D. work.

Dipika Majumdar was born in1986. She has done B. Tech. from Electrical and Electronics Engineering Department, Bengal College of Engineering, Durgapur, under West Bengal Technical University (WBUT), India in 2008. She has completed M. Tech. from Dr. B. C. Roy Engineering College, Durgapur, India under West Bengal Technical University (WBUT). Currently she is pursuing her Ph. D. work in the field of Power System Operation and Control using heuristic methods such as Particle Swarm Optimization (PSO).