LETTERS
International Journal of Recent Trends in Engineering, Vol 2, No. 6, November 2009

Economic Dispatch with Valve Point Effect using various PSO Techniques

K. Sathish Kumar * R. Rajaram V. Tamilselvan V. Shanmugasundaram S.Naveen
I.G.Mohamed Nowfal Hariharan T. Jayabarathi
School of Electrical Engineering, VIT University, Vellore-632014.Tamilnadu.India
k_sathishkr@sify.com rajaram_rpg@rediffmail.com tamil2210@yahoo.com sundara1982@yahoo.co.in
nav_vit88@yahoo.co.in ignowfal@gmail.com hari.terminated@gmail.com tjayabarathi@vit.ac.in

*Corresponding author

***Abstract*--Four modified versions of particle swarm optimizer (PSO) have been applied to the economic power dispatch with valve-point effects. In order to obtain the optimal solution, traditional PSO search a new position around the current position. The proposed strategies which explore the vicinity of particle's best position found so far leads to a better result. In addition, to deal with the equality constraint of the economic dispatch problems, a simple mechanism is also devised that the difference of demanded load and total generating power is evenly shared among units except the one reaching its generating limit. To show their capability, the proposed algorithms are applied to two cases with thirteen and forty generators, respectively. Comparison among particle swarm optimization and other modified particle swarm optimization is given. The results show that the proposed algorithms indeed produce more optimal solutions in both cases. The different PSO techniques are New PSO, Self Adaptive PSO and Chaotic PSO. Among the different PSO techniques, it is found that Self-Adaptive PSO is better than other PSO techniques in terms of better solution, speed of convergence, time of execution and robustness but it has more premature convergence.**

***Index Terms*—Economic dispatch, PSO, APSO, NPSO and CPSO**

I. Introduction

Scarcity of energy resources, increasing power generation cost and ever-growing demand for electric energy necessitates optimal economic dispatch in today's power systems. Optimal system operation involves consideration of economy of operation, system security, emission at fossil fuel plants and optimal release of water at hydro generation. In a practical power system, power plants are not loaded at the same distance from the center of loads and their fuel cost is different. The generation capacity is more than the demand and losses. So there is a need to schedule the generation. In an interconnected power system, the objective is to find the real and reactive power scheduling of each power plant in such a way to minimize the operation cost. The generators real and reactive powers are allowed to vary within certain limits to meet a particular load demand with minimum fuel cost. Electrical energy cannot be stored, but is generated from natural sources and delivered as demand arises. A transmission system is used for the delivery of bulk power over considerable distance and a distribution system is used for local deliveries. An interconnected power system consists of mainly three parts :

1. The generator, which produce electrical energy
2. The transmission line which transmit it to faraway places.
3. The load which use it.

Such a configuration applies to all inter connected networks, where the number elements may vary. The economic dispatch problem is to define the production level of each plant so that the total cost of generation and transmission for a prescribed schedule of loads.

II. Problem Description

Here we are considering only the generating units, but not as the system. We are neglecting the transmission line losses, line impedance etc. for analysis, the system is having only one bus with all generations and loads are connected.

As there is no transmission loss, the total demand P_D is the sum of all generation. For each plant assume the cost function FC_i

$$\text{Min } FC_{total} = \sum_{i=1}^{N_G} FC_i \qquad (1)$$

$$= \sum_{i=1}^{N_G} \alpha_i + \beta_i P_i + \gamma_i P_i^{\,2} \qquad (2)$$

Subject to the constraint

$$\sum_{i=1}^{N_G} P_i = P_D \qquad (3)$$

The power output of any generator should not exceed the its rating nor should it be below that necessary for stable turbine operation thus, the generations are restricted to lie within given minimum and maximum limits. The problem is to find the real power generation for each plant such that the objective function as defined by (2) is minimum, subject to the constrain given by (3) and the inequality constraints given by

$$P_i^{min} \le P_i \le P_i^{max} \quad \text{where } i = 1,2,3,\ldots\ldots\ldots N_G$$

P_i^{min}, P_i^{max}-minimum and maximum generating limits

FC_{total} – total production cost, FC_i – production cost of i_{th} plant, P_i – power generation of i_{th} plant, P_D – total load demand, N_G – total number of generating units

III. Overview of Particle Swarm Optimization

Kennedy and Eberhart first introduced PSO in year 1995. The features of the method are as follows:

The method is based on researches about swarms such as fish schooling and a flock of birds. Therefore, the computation time is short and it requires less memory. It was originally developed for nonlinear optimization problems with continuous variables. However, it is easily expanded to treat problems with discrete variables. Therefore, it is applicable to both continuous and discrete variables. Particle swarm optimization (PSO) is a form of swarm intelligence. Imagine a swarm of insects or a school of fish. If one sees a desirable path to go (e.g., for food, protection, etc.) the rest of swarm will be able to follow quickly even if they are on the opposite side of the swarm. On the other hand, in order to facilitate felicitous exploration of the search space, typically one wants to have each particle to have a certain level of "craziness" or randomness in their movement. . The position of each individual is represented by XY axis position and its velocity is expressed by V_x in x direction and V_y in y direction. Modification of the individual position is realized by the velocity and position information. PSO algorithm for N-dimensional problem formulation based on the above concept can be described as follows. Let P be the 'particle' co-ordinates (position) and V its speed (velocity) in a search space. Consider i as a particle in the total population (swarm). Now the ith particle position can be represented as

$P_i = (P_{i1}, P_{i2}, P_{i3}, \ldots\ldots P_{iN})$ in the N-dimensional space.

The best previous position of the i_{th} particle is stored and represented as

$P_{besti} = (P_{besti1}, P_{besti2}, \ldots\ldots P_{bestij})$.

The modified velocity of each particle can be calculated using the information, (i) the current velocity (ii) the distance between the current position and Pbest and (iii) the distance between the current position and gbest. This can be formulated as an equation

$$v_i^{k+1} = w_i v_i^k + (c_x * rand_i * (pbest_i - s_i^k)) + (c_2 * rand_i * (gbest - s_i^k)) \quad (4)$$

The current position can be modified by the following equation:

$$P_i^{(k+1)} = P_i^k + v_i^{k+1}$$

v_i^k - current velocity of particle i at iteration k, v_i^{k+1} modified velocity of particle i, P_i^k current position of particle i at iteration k, rand random number between 0 and 1

S_i^k current position of particle i at iteration

pbest- pbest of particle i, gbest- gbest of the group

w inertia weight factor

c1, c2 acceleration constant

The use of linearly decreasing inertia weight factor w has provided improved performance in all the applications. Its value is decreased linearly from about 0.9 to 0.4 during a run. Its value is set according to the following equation

$$w = w_{max} - \frac{w_{max} - w_{min}}{iter_{max}} * iter \quad (5)$$

Where w_{max} and w_{min} are both random numbers called initial weight and final weight

$iter_{max}$ the maximum iteration number

iter the current iteration number

$w_{max} = 0.9$, $w_{min} = 0.4$ and $C_1 = C_2 = 2.0$

In Eq. (4) the first term indicates the current velocity of the particle, second term represents the cognitive part of PSO where the particle changes its velocity based on its own thinking and memory. The third term represents the social part of PSO where the particle changes its velocity based on the social-psychological adaptation of knowledge.

Implementation of PSO method in ED problems

Step (1) : In the ED problems the number of on-line generating units is the 'dimension' of this problem. The particles are randomly generated between the maximum and the minimum operating limits of the generators. For example, if there are N units, the i_{th} particle is represented as follows:

$$P_i = (P_{i1}, P_{i2}\ldots\ldots\ldots\ldots P_{iN})$$

Step (2) : The particle velocities are generated randomly in the range $[-V_i^{max}, V_i^{max}]$.

The maximum velocity limit in the ith dimension is

$$V_i^{max} = \frac{P_{i.max} - P_{i,min}}{R}$$

Where R is the chosen number of intervals in the ith dimension

Step (3) : Objective function value of the particles are evaluated using the respective objective functions given by equ. (1), (2). These values are set as the Pbest value of the particles.

Step (4) : The best value among all the Pbest values, gbest, is identified.

Step (5) : New velocities for all the dimensions in each particle are calculated using equ. (4)

Step (6) : The position of each particle is updated using equ. (5).
Step (7) : The objective function values are calculated for the updated positions of the particles If the new value is better thant the previous Pbest, the new value is set to Pbest. If the stopping criteria are met, the positions of particles represented by gbest are the optimal solutions. Otherwise, the procedure is repeated from step (4).

Advantages of PSO

(1) PSO is easy to implement, and there are few parameters to adjust.
(2) In GAs, chromosomes share information so that the whole population moves like one group, but in PSO, only global best particle (Gbest) gives out information to the others. It is more robust.
(3) PSO can be more efficient than GAs; that is, PSO often finds the solutions with fewer objective function evaluations than are required by GAs.
(4) PSO uses payoff (performance index or objective function) information to guide the search in the problem space.
(5) Unlike GAs and other heuristic algorithms, PSO has the flexibility to control the balance between global and local exploration of the search space. This unique feature of PSO overcomes the premature convergence problem and enhances the searches the search capability.

IV.VARIOUS PSO TECHNIQUES

There are different types of PSO techniques where the few techniques used here are APSO, CPSO and NPSO

A. Adaptive Particle Swarm Optimization

In the simple PSO method, the inertia weight is made constant for all the particles in a single generation, but the most important parameter that moves the current position towards the optimum position is the inertia weight (x). In our adaptive PSO, the particle position is adjusted such that the highly fitted particle (best particle) moves slowly when compared to the lowly fitted particle. This can be achieved by selecting different x values for each particle according to their rank, between x_{min} and x_{max} as in the following form

$$w_i = w_{\min} + \frac{(w_{\max} - w_{\min}) * Rank_i}{Totalpopul\ ation} \quad (6)$$

So, from Eq. (6), it can be seen that the best particle takes the first rank, and the inertia weight for that particle is set to the minimum value while that for the lowest fitted particle takes the maximum inertia weight, which makes that particle move with a high velocity. The velocity of each particle is updated using Eq. (7), and if any updated velocity goes beyond V_{max}, it is limited to V_{max} using Eq. (8),

$$v_{ij}(t+1) = w_i v_{ij}(t) + c_1 r_1 (p_{ij}(t) - x_{ij}(t)) + c_2 r_2 (p_{gj}(t) - x_{ij}(t)) \quad (7)$$

$$v_{ij}(t+1) = sign(v_{ij}(t+1)) * \min(|v_{ij}(t+1)|, v_{j\max}) \quad (8)$$

$j = 1, 2 \ldots d$; $\quad i = 1, 2 \ldots n$

The new particle position is obtained by using Eq. (9), and if any particle position goes beyond the range specified, it is adjusted to its boundary using Eq. (10)

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1), \quad (9)$$

$j = 1, 2 \ldots d$; $\quad i = 1, 2 \ldots n$

$$x_{ij}(t+1) = \min(x_{ij}(t+1), range_{j\max}),$$

$$x_{ij}(t+1) = \max(x_{ij}(t+1), range_{j\min}) \quad (10)$$

The concept of re-initialization is introduced to the proposed APSO algorithm after a specific number of generations if there is no improvement in the convergence of the algorithm. This effect of population re-initialization is, in a sense, similar to the mutation operator in a GA. This effect is favorable when the algorithm prematurely converges to a local optimum and further improvement is not noticeable. This re-initialization of population is performed after checking the changes in the 'F_{best}' value in each and every specific number of generations.

B. The procedure of Adaptive Mechanism

Step 1: Get the input parameters like range [min max] for each of the variables, c_1, c_2,
Iteration counter = 0, V_{max}, w_{min} and w_{max} .
Step 2: Initialize n number of population of particles of dimension di with random Positions and velocities.
Step 3: Increment iteration counter by one.
Step 4: Evaluate the fitness function of all particles in the population, find particles best Position Pbest of each particle and update its objective value. Similarly, find the global best position (Gbest) among all the particles and update its objective value.
Step 5: If stopping criterion is met go to step (11). Otherwise continue.
Step 6: Evaluate the inertia factor according to Eq. (6), so that each particles movement is directly controlled by its fitness value.
Step 7: Update the velocity using Eq. (7) and correct it using Eq. (8).
Step 8: Update the position of each particle according to Eq (9), and if the new position goes out of range, set it to the boundary value using Eq. (10).

Step 9: The elites are inserted in the first position of the new population in order to maintain the best particle found so far.

Step 10: For every generations, this $F_{Best,new}$ value (at the end of these 5 generations) is compared with the $F_{Best,old}$ value (at the beginning of these 5 generations), if there is no noticeable change, then re-initialize k% of the population. Go to step (3).

Step 11: Output the Gbest particle and its objective value.

C. Chaotic Particle Swarm Optimization

PSO has gained much attention and widespread applications in different fields. However, the performance of the simple PSO greatly depends on its parameters, and it often suffers the problem of being trapped in local optima so as to prematurely converge . In order to avoid these disadvantages, Liu et al. proposed a chaotic particle swarm optimization (CPSO) method that combines PSO with AIWF (adaptive inertia weight factor) and chaotic local search (CLS) based on the logistic equation.

$$w = \begin{cases} w_{\min} + \dfrac{(w_{\max} - w_{\min})(f - f_{\min})}{f_{avg} - f_{\min}} & f \le f_{avg} \\ w_{\max} & f \ge f_{avg} \end{cases}$$

where w_{max} and w_{min} maximum and minimum of w
f is the current objective value of the particle f_{avg} and f_{min} are the average and minimum objective values of all particles. The procedures of CLS based on the logistic equation can be illustrated as follows:

Step 1: Setting k=0 and mapping the decision variables $x_i^{(k)}$, I = 1, 2,………………n among the intervels (xmin,I, xmax,i), I = 1,2,…………….n to the chaotic variables $cx_i^{(k)}$ located in the interval (0,1) using the following equation

$$cx_i^{(k)} = \frac{x_i^k - x_{\min,i}}{x_{\max,i} - x_{\min,i}}, \text{ I=1,2,………………n}$$

Step 2: Determine the chaotic variables $cx_i^{(k+1)}$ for the next iteration using the logistic equation according to $cx_i^{(k)}$

Step 3: Convert the chaotic variables $cx_i^{(k+1)}$ to the decision variables a $x_i^{(k+1)}$ using the following equation

$$x_i^{(k+1)} = x_{\min,i} + cx_i^{(k+1)}(x_{\max,i} - x_{\min,i}),$$

where i = 1, 2,……………..n

Step 4: Evalute the new solution with decision variables $x_i^{(k+1)}$, I = 1, 2,…………..n

Step 5: If the new solution is better than $X_{(0)} = [[x_1^{(0)}, \ldots\ldots, x_n^{(n)}]$ or the predefined maximum iteration is reached, output the new solution as the result of the CLS; otherwise, let k = k+1 and go to back to step 2.

D. New Particle Swarm Optimization

PSO is a population-based, self-adaptive, stochastic optimization technique. The velocity of a particle is influenced by three components, namely, inertial, cognitive, and social. The inertial component simulates the inertial behavior of the bird to fly in the previous direction. The particles move around the multidimensional search space until they find the food (optimal solution). Based on the above discussion, the mathematical model for PSO is as follows.
Velocity update equation is given by

$$v_i^{k+1} = w_i v_i^k + (c_{1g} * r1 * (pbest_i - s_i^k)) + (c_{1w} * r2 * (s_i^k - pworst_i)) + (c_2 * r3 * (gbest_i - s_i^k))$$

r_1, r_2, r_3 are the random no. between [0,1]
$pworst_i$ --- pworst of particle i $C_{1g} = 1.4, C_{1w} = 0.6$

4.3.1 Initialization of the Best and Worst Positions:

In the strategy of PSO, the particle's best position (P_{best}) and global best position(G_{best}) are the key factors. The best position of a particle is the position, which gives the minimum PF_T, and the best position out of all the P_{best} is taken as. In this paper, the particle's worst position (P_{worst}) is introduced. At the beginning of the iteration process, the P_{best} and P_{worst} for all the particles are taken as the same as the initial positions. The PF_T at G_{best}, is taken as $F_{Gbest}{}^0$. The search procedures of the proposed NPSO methods for ED are described as follows:

Step 1: Input data's are to be given.
Step 2: Initialization of position, velocity, Pbest, Pworst, Gbest and iteration count.
Step 3: Incrasing the iteration count.
Step 4: Updating the position, velocity, Gbest,Pbest and Pworst.
Step 5: Invoke LRS (local random search)
Step 6: Better Gbest optained by LRS.
Step 7: Bettter Gbest value is not obtained by LRS
Step 8: Repalce Gbest of NPSO with optimum of LRS
Step 9: Maximum iteration for NPSO
Step 10: Result is obtained.

IEEE13 generator system contains the following input datas (for EL problems)

GENERATORS DATA FOR CASE 1 (13 UNITS)

G	P_{min}(MW)	P_{max}(MW)	a	b	c	e	f
1	0	680	0.00028	8.10	550	300	0.035
2	0	360	0.00056	8.10	309	200	0.042
3	0	360	0.00056	8.10	307	200	0.042
4	60	180	0.00324	7.74	240	150	0.063
5	60	180	0.00324	7.74	240	150	0.063
6	60	180	0.00324	7.74	240	150	0.063
7	60	180	0.00324	7.74	240	150	0.063
8	60	180	0.00324	7.74	240	150	0.063
9	60	180	0.00324	7.74	240	150	0.063
10	40	120	0.00284	8.6	126	100	0.084
11	40	120	0.00284	8.6	126	100	0.084
12	55	120	0.00284	8.6	126	100	0.084
13	55	120	0.00284	8.6	126	100	0.084

V. ANALYSIS OF THE RESULT

A. Results obtained by using the PSO

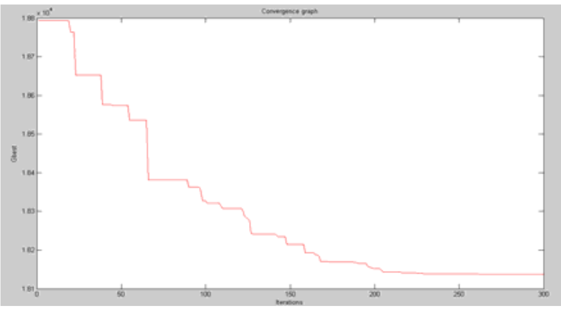
Parameters

1. Total population size =100
2. Total load demand=1800MW
3. Maximum iteration=300
4. Acceleration coefficient c1, c2=2
5. Weight inertia factor,
   W_{max}=0.9
   W_{min}=0.4

Output:

Method	Total Gen. cost($)
PSO(ref)	1.8104e+004
PSO	1.8105e+004

Graph of G_{best} versus iterations for best solutions of PSO:



B. Results obtained by using the APSO

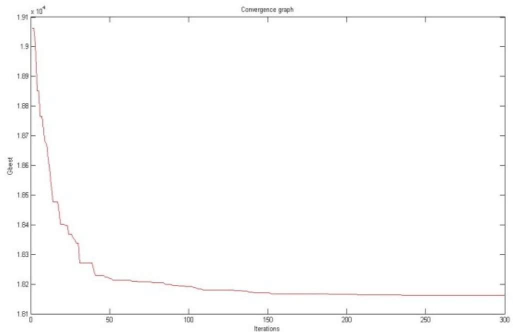
Parameters

1. Total population size =100
2. Total load demand=1800 MW
3. Maximum iteration=300
4. Acceleration coefficient c1=0.5, c2=2.5
4. Weight inertia factor,
   W_{max}=0.9
   W_{min}=0.45

Output:

Method	Total Gen.cost($)
PSO(ref)	1.8104e+004
PSO	1.8105e+004
SAPSO	1.8076e+004

Graph of G_{best} versus iterations for best solutions of APSO:



C. Results obtained by using the CPSO

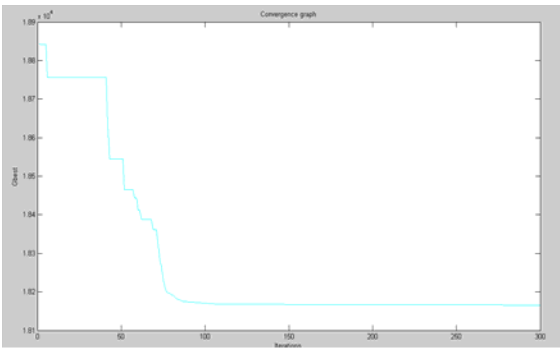
Parameters

1. Total population size =100, 2. Total load demand=1800 MW, 3. Maximum iteration=300,
4. Acceleration coefficient c1, c2=2, 5. Weight inertia factor, W_{max}=0.9, W_{min}=0.4

Output:

Method	Total Gen.cost($)
PSO(ref)	1.8104e+004
PSO	1.8105e+004
SAPSO	1.8076e+004
CPSO	1.8090e+004

Graph of G_{best} versus iterations for best solutions of CPSO:



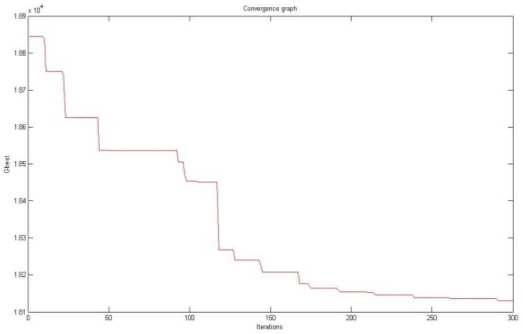
D. Results obtained by using the NPSO

Parameters
1. Total population size =100, 2. Total load demand=1800 MW, 3. Maximum iteration=300
4. Acceleration coefficient c1=1.4, c2=0.6,
5. Weight inertia factor, W_{max}=0.9, W_{min}=0.4

Output:

Method	Total Gen.cost($)
PSO(ref)	1.8104e+004
PSO	1.8105e+004
SAPSO	1.8076e+004
CPSO	1.8090e+004
NPSO	1.8124e+004

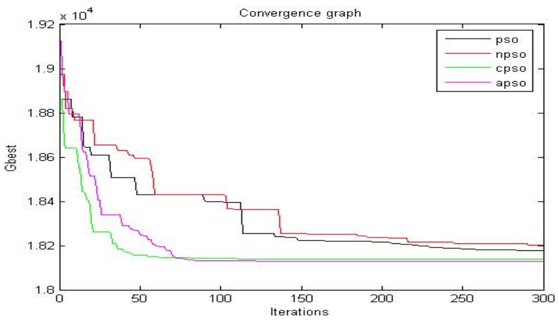
Graph of G_{best} versus iterations for best solutions of NPSO:



E. Analysis of four PSO techniques

Algorithm	Gen.cost($)			Time of execution	No. of iterations
	Min.	Max.	Avg.		
PSO(ref)	18014	18207	18014	---------	300
PSO	18029	18178	18015	4.813000	300
CPSO	18074	18101	18090	4.797000	300
APSO	18063	18085	18076	4.954000	300
NPSO	18038	18244	18124	6.125000	300

F. Comparison of graphs



VI. DISCUSSION AND CONCLUSION

In this paper, we have successfully employed the PSO method to solve the ED problem. The PSO algorithm has been demonstrated to have superior features, including high-quality solution, stable convergence characteristic, and good computation efficiency From the results it is inferred that Self Adaptive PSO gives the better solution for IEEE 13 generator system when compared to other PSO techniques. Speed of convergence is fast in SAPSO and CPSO when compared to other PSO techniques.
SAPSO and CPSO are robust when compared to other PSO techniques. Premature convergence is more in CAPSO when compared to other PSO techniques. Computational time is more for New PSO when compared to other PSO techniques. Self-Adaptive Particle Swarm Optimization is the most suitable optimization technique for Economic Load Dispatch Problem by comparing various PSO algorithms.

ACKNOWLEDGEMENT

We thank our guide and the management of Vellore Institute of Technology, Vellore, India, for providing the necessary facilities and encouragement to carry out this work.

REFERENCES

1.C.H.Chen, S.N. Yeh senior member IEEE, “Particle Swarm Optimization for Economic Dispatch with Valve-Point Effect”.
2.M.Sudhakaran,P.Ajay-D-Vimal Raj and T.G.Palanivelu, “Application of Particle Swarm Optimization for Economic Load Dispatch Problem”.
3.Zhengjia Wu, Jianzhong Zhou “A Self-Adaptive Particle Swarm Optimization Algorithm with Individual Coefficients Adjustment”.
4.D.N.Jeyakumar, T.Jayabarathi, T.Raghunathan “Particle Swarm Optimization for various types of Economic Dispatch Problem”.
5.Nidul sinha, R.Chakrabarti, and P.K.Chattopadhyaya, “Evolutionary Programming Techniques for Economic Load Dispatch”.
6.Cai Jiejin, Ma Xiaoquian, Li Lixiang, Peng Haipeng, “Chaotic Particle Swarm Optimization for Economic Dispatch”.
7.A.Immanuel Selvakumar, K.Thanushkodi “A New Particle Swarm Optimization Solution to Nonconvex Economic Dispatch Problems”.
8.Haadi Sadat “Power Systems” , presents and excellent introduction to the Economic Dispatch .
9.Zwe-Lee Gaing “Particle Swarm Optimization to solving the Economic Dispatch Considering the Generator Constraints”.
10.A.J.Wood “Power generation,operation and Control”.