Journal of Universal Computer Science, vol. 18, no. 13 (2012), 1842-1852
submitted: 15/11/11, accepted: 30/3/12, appeared: 1/7/12 © *J.UCS*

Solving Economic Dispatch Problems with Valve-point Effects using Particle Swarm Optimization

Kusum Deep
(Department of Mathematics
Indian Institute of Technology Roorkee, Roorkee, India
kusumfma@iitr.ernet.in)

Jagdish Chand Bansal
(ABV-Indian Institute of Information Technology and Management Gwalior, Gwalior, India
jcbansal@iiitm.ac.in)

Abstract: Particle Swarm Optimization (PSO) is a swarm intelligence optimization method inspired from birds' flocking or fish schooling. Many improved versions of PSO are reported in literature, including some by the authors. Original as well as improved versions of PSO have proven their applicability to various fields like science, engineering and industries. Economic dispatch (ED) problem is one of the fundamental issues in power system operations. This problem turns out to be a non linear continuous optimization problem. In this paper, economic dispatch problem is solved using original PSO and two of its improved variants, namely, Laplace Crossover PSO (LXPSO) and Quadratic Approximation PSO (qPSO), in order to find better results than reported in the literature. Results are also compared with the earlier published results.

Keywords: Particle Swarm Optimization, Economic Dispatch, Laplace Crossover, Quadratic Approximation Particle Swarm optimization
Categories: G.1.6, I.1.2

1 Introduction

In the operation and planning of a power system, economic dispatch problem (ED) is one of the key problems that are to be dealt with. In ED problem, the optimal combination of power outputs of all generating units is to be determined, subject to meeting the required load demand at minimum operating cost while satisfying system equality and inequality constraints. The practical ED problem is a non-smooth optimization problem consisting of both equality and inequality constraints. However, there is no general traditional approach; dynamic programming method [Liang, 92] has been used to solve this problem. But the performance of dynamic programming method reduces significantly as the dimension of the problem increases. Over the past few years, many efficient non-traditional methods have been explored to solve the ED problem, such as genetic algorithm [Walters, 93], evolutionary programming [Yang, 96], [Sinha, 2003], tabu search [Lin, 2002], neural network approaches [Lee, 98], and particle swarm optimization [Park, 2005], [Victoire, 2004], [Park, 2006], [Park, 2007]. Due to wide applicability and scope of improvements, particle swarm optimization and some of its advanced variants are applied to solve the ED problem.

Two variants of PSO namely, Laplace Crossover PSO (LXPSO) [Bansal, 2009] and Quadratic Approximation PSO (qPSO) [Deep, 2009] have been proved to be efficient techniques for optimization test problems. Therefore, in order to find improved results, this paper presents the solution of economic dispatch problem with valve-point effects [Park, 2007] using standard PSO, LXPSO and qPSO. Results obtained by these three algorithms are also compared with the earlier published results.

Rest of the paper is organized as follows: Section 2 presents the mathematical formulation of economic dispatch problem with valve-point effects. In section 3, PSO and in section 4, LXPSO and qPSO are summarized. Numerical results are obtained and analyzed in section 5. Section 6, concludes the paper.

2 Mathematical Formulation of the Problem

The main objective of ED problem is to minimize the total fuel cost of power plants subject to the operating constraints of a power system. Generally, it can be formulated with an objective function and two constraints [Park, 2007]:

$$\text{Min}\, F_T = \sum_{i=1}^{n} F_i(P_i) \tag{1}$$

where $F_i(P_i) = a_i + b_i P_i + c_i P_i^2 \qquad \forall\ i = 1,2,....,n \qquad (2)$ and

F_T : Total generation cost
F_i : Cost function of generator *i*
a_i, b_i, c_i: Cost coefficients of generator *i*
P_i: Power output of generator *i*
n: Number of generators.

Subject to,
Constraint I:
Active Power Balance Equation: Power balance requires an equality constraint should be satisfied. The total generated power should be the equal to the total demand and the total line loss. For simplicity purpose, the transmission loss is not considered in this paper.

Constraint II:
Minimum and Maximum Power Limits: Generation output of each power generating unit should be bounded between its minimum and maximum limits. The corresponding inequality constraints for each generator are:

$$P_{i,\min} \le P_i \le P_{i,\max} \tag{3}$$

where $P_{i,\min}$ and $P_{i,\max}$ are the minimum and maximum output of generator *i*, respectively.

The fuel cost function is significantly modified if the generation units with multi-valve steam turbines are considered.

The generating units with multi-valve steam turbines exhibit a greater variation in the fuel cost functions. Since the valve point effects result in the ripples, a cost function contains higher order nonlinearity. Therefore, to consider the valve point effects, the cost function (2) may be written as:

$$F_i(P_i) = a_i + b_i P_i + c_i P_i^2 + \left| e_i \times \sin\left(f_i \times \left(P_{i,\min} - P_i \right) \right) \right| \qquad \forall\ i = 1,2,....,n \qquad (4)$$

where e_i and f_i are the cost coefficients of generator *i* reflecting valve-point effect.

Thus, the nonlinear optimization problem defining economic dispatch problem with valve point effects is to minimize F_T given by (1) and (4) subject to the constraints I and II, discussed above.

3 Particle Swarm Optimization

The particle swarm optimization algorithm, originally introduced in terms of social and cognitive behaviour by Kennedy and Eberhart in 1995 [Kennedy, 95], solves problems in many fields, especially engineering and computer science. Only within a few years of its introduction, PSO has gained wide popularity as a powerful global optimization tool and is competing with well-established population based evolutionary and swarm intelligence algorithms. The fundamental idea behind PSO is the mechanism by which the birds in a flock and the fishes in a school cooperate while searching for food. In PSO, a group of active, dynamic, and interactive members called swarm produces a very intelligent search behaviour using collaborative trial and error. Each member of the swarm called particle, represents a potential solution of the problem under consideration. Each particle in the swarm relies on its own experience as well as the experience of its best neighbour. Each particle has an associated fitness value. These particles move through search space with a specified velocity in search of optimal solution. Each particle maintains a memory which helps it in keeping the track of the best position, it has achieved so far. This is called the particle's personal best position (pbest) and the best position, the swarm has achieved so far is called global best position (gbest). The movement of the particles is influenced by two factors using information from iteration-to-iteration as well as particle-to-particle. As a result of iteration-to-iteration information, the particle stores in its memory the best solution visited so far, called *pbest,* and experiences an attraction towards this solution as it traverses through the solution search space. As a result of the particle-to-particle information, the particle stores in its memory the best solution visited by any particle, and experiences an attraction towards this solution, called *gbest*, as well. The first and second factors are called cognitive and social components, respectively. After each iteration, the *pbest* and *gbest* are updated for each particle if a better or more dominating solution (in terms of fitness) is found. This process continues, iteratively, until either the desired result is converged upon, or it is determined that an acceptable solution cannot be found within computational limits.

For a D-dimensional search space, the i^{th} particle of the swarm is represented by a *D-* dimensional vector, $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})^T$. The velocity of this particle is represented by another D-dimensional vector $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})^T$. The previously best visited position of the i^{th} particle is denoted as $P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})^T$. '*g*' is the index of the best particle in the swarm. The velocity of the i^{th} particle is updated using the velocity update equation given by

$$v_{id} = v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) \quad (5)$$

and the position is updated using

$$x_{id} = x_{id} + v_{id} \quad (6)$$

where $d = 1, 2\ldots D$ represents the dimension and $i = 1, 2,\ldots, S$ represents the particle index. S is the size of the swarm and c_1 and c_2 are constants, called cognitive and social scaling parameters respectively (usually, $c_1 = c_2$ and r_1, r_2 are random numbers drawn from a uniform distribution). Equations (5) and (6) define the classical version of PSO algorithm. A constant, *Vmax*, is used to arbitrarily limit the velocities of the particles and improve the resolution of the search. The maximum velocity *Vmax*, serves as a constraint to control the global exploration ability of particles in the swarm. Further, the concept of an inertia weight was developed to better control exploration and exploitation. The motivation is to be able to eliminate the need for *Vmax*. The inclusion of an inertia weight in the particle swarm optimization algorithm was first reported in the literature in 1998 [Shi, 98].

The resulting velocity update equation becomes:

$$v_{id} = w * v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) \quad (7)$$

Eberhart and Shi, [Eberhart, 2000] indicate that the optimal strategy is to initially set w to 0.9 and reduce it linearly to 0.4, allowing initial exploration followed by acceleration toward an improved global optimum.

Clerc [Clerc, 99] has introduced a constriction factor, χ , which improves PSO's ability to constrain and control velocities. χ is computed as:

$$\chi = \frac{2}{\left|2 - \phi - \sqrt{\phi(\phi - 4)}\right|} \quad (8)$$

where $\phi = c_1 + c_2, \phi > 4$, and the velocity update equation is then

$$v_{id} = \chi * \left(v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id})\right) \quad (9)$$

Eberhart and Shi, [Eberhart, 2000] found that χ, combined with constraints on *Vmax*, significantly improves the PSO performance.

4 Some Improved PSO Versions

4.1 Laplace Crossover Particle Swarm Optimization (LXPSO)

In classical particle swarms, the particle moves using the information from its previous best and the global best particle. In LXPSO [Bansal, 2009], a crossover based on Laplace distribution is introduced in PSO that develops an interaction model between any two randomly chosen particles. The details of LXPSO are as follows:

Laplace crossover operator was first introduced for genetic algorithms in [Deep, 2007]. This is a parent centric operator. LX has similar properties like Simulated Binary Crossover Operator (SBX) [Deb, 2001]. The probability density function for Laplace distribution is given by

$$f(x \mid a,b) = \frac{1}{2b}\exp\left(-\frac{|x-a|}{b}\right), \; -\infty < x < \infty \tag{10}$$

and the cumulative density function of Laplace distribution is given by

$$F(x \mid a,b) = \begin{cases} \frac{1}{2}\exp\left(-\frac{|x-a|}{b}\right), \; x \le a \\ 1-\frac{1}{2}\exp\left(-\frac{|x-a|}{b}\right), \; x > a \end{cases} \tag{11}$$

where, $a \in R$ is called the location parameter and $b > 0$ is termed as scale parameter.

Using LX, two off-springs $y_1 = (y_{11}, y_{12},..., y_{1D})$ and $y_2 = (y_{21}, y_{22},..., y_{2D})$ are generated from a pair of parents $x_1 = (x_{11}, x_{12},..., x_{1D})$ and $x_2 = (x_{21}, x_{22},..., x_{2D})$ in the following way:

First, a uniformly distributed random number $u_i = (0, 1)$ is generated. Then, from Laplace distribution function, the ordinate β_i is calculated so that the area under the probability curve excluding area from *a* (location parameter) to β_i is equal to chosen random number u_i. The calculation is carried out in the following way:

First consider β_i right to a, then

$$u_i = 1 - \int_a^{\beta_i} \frac{1}{2b}\exp\left(-\frac{|x-a|}{b}\right)dx$$

Since $a \le x \le \beta_i$, so $|x-a| = (x-a)$

$$u_i = 1 - \int_a^{\beta_i} \frac{1}{2b}\exp\left(-\frac{(x-a)}{b}\right)dx \; = 1 - \left[\frac{\frac{1}{2b}\exp\left(-\frac{(x-a)}{b}\right)}{-\frac{1}{b}}\right]_a^{\beta_i}$$

$$= 1 + \frac{1}{2}\left[\exp\left(-\frac{(\beta_i - a)}{b}\right) - 1\right] \Rightarrow 2u_i - 1 = \exp\left(-\frac{(\beta_i - a)}{b}\right)$$

$$\Rightarrow \beta_i = a - b\log_e(2u_i - 1)$$

Similarly, when β_i is left to a, then

$$u_i = 1 - \int_{\beta_i}^{a} \frac{1}{2b}\exp\left(-\frac{|x-a|}{b}\right)dx \Rightarrow \beta_i = a - b\log_e(1 - 2u_i)$$

Thus,

$$\beta_i = \begin{cases} a - b\log_e(1-2u_i), & u_i \le \frac{1}{2} \\ a - b\log_e(2u_i - 1), & u_i > \frac{1}{2} \end{cases} \tag{12}$$

The offsprings are given by the equations

$$y_{1i} = x_{1i} + \beta_i |x_{1i} - x_{2i}| \tag{13}$$

$$y_{2i} = x_{2i} + \beta_i |x_{1i} - x_{2i}| \tag{14}$$

$$i = 1, 2, ..., n.$$

From the above two equations it is clear that both the offsprings are placed symmetrically with respect to the position of the parents. For smaller values of b, offsprings are likely to be close to the parents in search space and for larger values of b offsprings are expected to be far from the parents. For a fixed value of a and b, LX dispenses off-springs proportional to the spread of parents i.e. if the parents are near to each other, the off-springs are expected to be near to each other and if the parents are far from each other then the off-springs are likely to be far from each other. A realization of the above idea can be had from equation (15) which is derived from equations (13) and (14),

$$y_{1i} - y_{2i} = x_{1i} - x_{2i} \tag{15}$$

In this way the proposed crossover operator exhibits self-adaptive behaviour. Note that the spiky nature of the Laplacian distribution controls the spread of the offsprings.

Based on the Laplacian operator described as above, two new particles are formed. The best particle (in terms of fitness) is selected. This new particle, called Laplacian particle, can replace one of the particles from which it is formed or replace the worst performing particle in the swarm. LXPSO analyze swarms behaviour if the worst particle (in terms of fitness) is replaced by this Laplacian particle. PSO with Laplace crossover is called as Laplace Crossover PSO (LXPSO).

4.2 Quadratic Approximation Particle Swarm Optimization (qPSO)

4.2.1 Motivation

Deep and Das [Deep, 2008], hybridized a binary GA by incorporating Quadratic Approximation (QA) operator as an additional operator for local search which showed a substantial improvement in the performance of GA. PSO has the efficiency to solve a wide variety of problems with a larger percentage of success. Mohan and Shankar [Mohan, 94] proved that Random Search technique (RST) which uses QA operator provides fast convergence rate but once stuck in a local optima, it is generally difficult to come out of it. Perhaps social knowledge concept of PSO could help RST in coming out of the local optima. As compared to GAs, the PSO has much more profound intelligent background and could be performed more easily. These two facts motivated to hybridize PSO and QA with the expectation of faster convergence (from QA) and improved results (from PSO) [Deep, 2009].

4.2.2 Quadratic Approximation Operator

QA is an operator which determines the point of minima of the quadratic hyper surface passing through three points in a D- dimensional space. It works as follows:
1. Select the particle R_1, with the best objective function value. Choose two random particles R_2 and R_3 such that out of R_1, R_2 and R_3, at least two are distinct.
2. Find the point of minima R* of the quadratic surface passing through R_1, R_2 and R_3, where

$$R^* = 0.5\left(\frac{\left(R_2^2 - R_3^2\right)f\left(R_1\right) + \left(R_3^2 - R_1^2\right)f\left(R_2\right) + \left(R_1^2 - R_2^2\right)f\left(R_3\right)}{\left(R_2 - R_3\right)f\left(R_1\right) + \left(R_3 - R_1\right)f\left(R_2\right) + \left(R_1 - R_2\right)f\left(R_3\right)}\right) \qquad (16)$$

where $f\left(R_1\right)$, $f\left(R_2\right)$ and $f\left(R_3\right)$ are the objective function values at R_1, R_2 and R_3 respectively. The calculations are to be done component wise using (16) to obtain R*.

4.2.3 The Process of Hybridization

In each iteration, the whole swarm S is divided into two subswarms (say S_1 and S_2). From one generation to the next generation, S_1 is evolved using PSO, whereas S_2 is evolved using QA. Figure 1 shows the idea that stands behind qPSO and the way to integrate the two techniques. qPSO consists of a strong co-operation of QA and PSO, since it maintains the integration of the two techniques for the entire run. It should be noted that R_1 used in QA and gbest used in PSO both are the global best position of the entire swarm (let us call it GBEST) i.e R_1 = GBEST and gbest = GBEST. The strength of the qPSO lies in the facts that both PSO and QA use the GBEST simultaneously or in other words, subswarm S_1 and S_2 share their best positions with each other and for transition from one iteration to the next, both updating schemes use the entire swarm's information. However, in updating a particle's position by QA, no information about its current position is applied as in PSO but the presence of memory of the corresponding subswarm preserves the best performed particles. So in *(i+1)*th iteration QA will not produce worse solution than that in *i*th iteration. For more details of qPSO process refer [Deep, 2009].

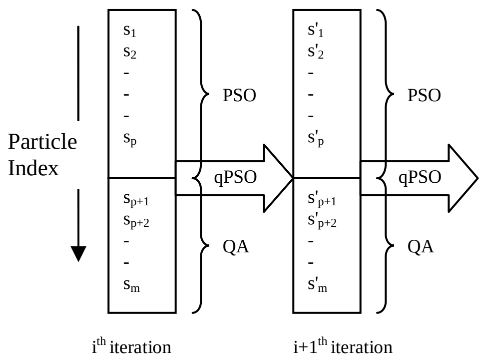

Figure 1: Transition from ith iteration to (i+1)th iteration

Percentage of swarm to be updated by PSO or QA is an important parameter of qPSO known as *coefficient of hybridization (CH). CH* is the percentage of swarm which is evolved using QA in each iteration. Thus, if CH = 0, then the algorithm is pure PSO (the whole swarm is updated by PSO operators), and if CH = 100 then the algorithm is pure QA (the whole swarm is updated by QA operator) while for 0<CH<100 the corresponding percentage of swarm is evolved by QA and the rest with PSO. The optimal value of CH is 30% [Deep, 2009].

5 Solution of the Problem

In this paper, the power system of 40 generating units with valve-point effects is considered [Park, 2007]. Refer [Park, 2007], for the input data of the test system with 40 generating units and the total demand is considered as 10,500 MW.

5.1 Selection of Parameters

In the literature, different values of parameters are used. In this paper, the selection of parameters is based on [Bansal, 2009] and [Deep, 2009]. Swarm size S is set to be 100. Constriction coefficient version of PSO, LXPSO and qPSO are applied to solve the considered problem. Constriction coefficient is calculated from equation (8). The cognitive and social scaling parameters c_1 and c_2 are set to 2.8 and 1.3 respectively [Bansal, 2009]. Maximum velocity, Vmax is set equal to 0.5*(Xmax-Xmin), where Xmax and Xmin are the upper and lower bounds of corresponding decision variable. The location and scale parameters a and b for Laplace crossover are 1 and 0.9, respectively [Bansal, 2009]. The total simulations considered are 100. The criterion to terminate a simulation of the algorithms is reaching maximum number of iterations which is set 3000.

5.2 Computational Results

In Table 1, the minimum objective function value (Min OBJ), mean objective function value (Mean OBJ), and the standard deviation (SD) obtained by PSO, LXPSO, and qPSO are tabulated. Table 1 also compares the results obtained in this paper to the earlier published results by Hybrid PSO with crossover (HPSO) [Park, 2007] and Improved Particle Swarm Optimization (IPSO) [Park, 2006]. Table 2 summarizes the generation output of each generator and the corresponding cost in 40-unit system obtained by PSO, LXPSO and qPSO. It is observed that the generation output obtained by PSO, LXPSO and qPSO satisfy both the constraints and the minimum cost obtained by qPSO is the best over PSO, LXPSO and other methods applied earlier for this problem. Therefore, qPSO with the proposed parameter setting is strongly recommended for the solution of economic dispatch problems with valve-point effects.

Method	Min OBJ	Mean OBJ	SD
PSO	121425.7989	121590.3289	92.8702
LXPSO	121416.8352	121503.0048	57.6192
qPSO	121411.7764	121503.0048	57.6192
HPSO[1] [Park, 2007]	121452.6741	121537.1906	-
IPSO[1] [Park, 2006]	121432.177	121801.909	287.452

Table 1: Comparison of results obtained by PSO, LXPSO, qPSO, HPSO, and IPSO.

Unit	PSO	LXPSO	qPSO
1	112.2629	111.4317	111.776
2	111.9654	111.0495	111.7013
3	97.53544	97.44937	97.41573
4	179.7479	179.8268	179.7756
5	90.14323	88.02182	88.315
6	139.9949	139.9918	139.989
7	259.7157	259.7505	259.7313
8	284.8357	284.9749	284.7436
9	284.7714	284.5966	284.6194
10	130.017	130.0184	130.0528
11	168.8275	168.7991	168.8038
12	94.06347	168.803	168.801
13	214.7621	214.7737	214.7706
14	394.2736	394.2471	304.5479
15	304.5572	304.4901	394.283
16	394.2549	394.2577	394.2789
17	489.3522	489.4279	489.2917
18	489.3013	489.344	489.3259
19	511.2805	511.323	511.2811
20	511.3106	511.2519	511.2862
21	523.3612	523.4927	523.3405
22	523.337	523.2936	523.2933
23	523.3437	523.2838	523.3627
24	523.2956	523.4253	523.3602

[1] Results reproduced from corresponding literature

25	523.3138	523.3875	523.3191
26	523.4079	523.3427	523.287
27	10.0301	10.03665	10.02257
28	10.02232	10.03829	10.02202
29	10.01248	10.0495	10.0232
30	89.51979	88.88378	88.62734
31	189.9937	189.978	189.9887
32	189.9976	189.9673	189.9908
33	189.998	189.9456	189.9911
34	166.081	164.8954	164.9038
35	199.9889	165.2111	165.0771
36	199.9752	165.7175	165.3116
37	109.981	109.9597	109.9916
38	109.9851	109.9762	109.9878
39	109.9835	109.9999	109.9741
40	511.3926	511.2881	511.3359
Total Power (MW)	10,500	10,500	10,500
Total Generation Cost	121425.79	121416.83	121411.77

Table 2 Generation output of each generator in case of minimum total cost and the corresponding total cost in 40-unit system for PSO, qPSO and LXPSO.

References

[Bansal, 2009] Bansal, J.C., Deep, K., Veeramachaneni, K., Osadciw, L.: Information Sharing Strategy among Particles in Particle Swarm Optimization using Laplacian Operator, *Swarm Intelligence Symposium, SIS '09. IEEE*, pp.30-36 2009.

[Clerc, 99] Clerc, M.: The Swarm and the Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization, in *Proc. 1999 ICEC*, Washington, DC, pp. 1951-1957 1999.

[Deb, 2001] Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms, Wiley, ch. 4 2001.

[Deep, 2007] Deep, K. and Thakur, M.: A New Crossover Operator for Real Coded Genetic Algorithms, Applied Mathematics and Computation 188, pp. 895–911 2007.

[Deep, 2008] Deep, K., and Das, K. N.: Quadratic Approximation based Hybrid Genetic Algorithm for Function Optimization, Applied Mathematics and Computation, 203(1) 86–98 2008

[Deep, 2009] Deep, K. and Bansal, J. C.: Hybridization of Particle Swarm Optimization with Quadratic Approximation, OPSEARCH, Vol. 46, Issue 1, pp. 3-24 2009.

[Eberhart, 2000] Eberhart, R. C. and Shi, Y.: Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization, *2000 Congress on Evolutionary Computing*, vol. 1, pp. 84-88 2000.

[Kennedy, 95] Kennedy, J., and Eberhart, R. C.: Particle Swarm Optimization, *Proceedings IEEE International Conference Neural Network.* Vol. 4, pp. 1942 – 1948 1995.

[Lee, 98] Lee, K. Y., Sode-Yome, A., and Park, J. H.: Adaptive Hopfield Neural Network for Economic Load Dispatch," *IEEE Trans. on Power Systems*, Vol. 13, No. 2, pp. 519-526 May 1998.

[Liang, 92] Liang, Z. X. and Glover, J. D.: A Zoom Feature for a Dynamic Programming Solution to Economic Dispatch Including Transmission Losses, IEEE Trans. on Power Systems, Vol. 7. No. 2, pp. 544-550 May 1992.

[Lin, 2002] Lin, W. M., Cheng, F. S., and Tsay, M. T.: An Improved Tabu Search for Economic Dispatch with Multiple Minima, *IEEE Trans. on Power Systems*, Vol. 17, No. 1, pp. 108-112 Feb. 2002.

[Mohan, 94] Mohan, C. and Shanker, K. (now Deep, K.): A Random Search Technique for Global Optimization based on Quadratic Approximation, Asia Pacifi c Journal of Operations Research, 11 93–101 1994.

[Park, 2005] Park, J. B., Lee, K. S., Shin, J. R., and Lee, K. Y.,: A Particle Swarm Optimization for Economic Dispatch with Nonsmooth Cost Functions, *IEEE Trans. on Power Systems*, Vol. 20, No. 1, pp. 34-42 Feb. 2005.

[Park, 2006] Park, Jong-Bae, Jeong, Yun-Won, Kim, Hyun-Houng and Shin, Joong-Rin: An Improved Particle Swarm Optimization for Economic Dispatch With Valve Point Effect, *International Journal of Innovations in Energy Systems and Power*, Vol. 1, no. 1 November 2006.

[Park, 2007] Park, J. B., Jeong, Y. W., Shin, J. R., Lee, K.Y. and Kim, J. H.: A Hybrid Particle Swarm Optimization Employing Crossover Operation for Economic Dispatch Problems with Valve-point Effects, *Intelligent Systems Applications to Power Systems, 2007 ISAP 2007. International Conference*, pp.281-286 2007.

[Shi, 98] Shi, Y. and Eberhart, R. C.: A Modified Particle Swarm Optimizer, in *Proc. IEEE International Conference on Evolutionary Computation*, Piscataway, NJ, IEEE Press, pp. 69–73 1998.

[Sinha, 2003] Sinha, N., Chakrabarti, R., and Chattopadhyay, P. K.: Evolutionary Programming Techniques for Economic Load Dispatch, *IEEE Trans. On Evolutionary Computations*, Vol. 7, No. 1, pp. 83-94 Feb. 2003.

[Victoire, 2004] Victoire, T. A. A. and Jeyakumar, A. E.: Hybrid Pso-Sqp for Economic Dispatch with Valve-Point Effect, *Electric Power Systems Research*, Vol. 71, No. 1, pp. 51-59 2004.

[Walters, 93] Walters, D. C. and Sheble, G. B.: Genetic Algorithm Solution Of Economic Dispatch With The Valve Point Loading, *IEEE Trans. on Power Systems*, Vol. 8, No. 3, pp. 1325-1332 Aug. 1993.

[Yang, 96] Yang, H. T., Yang, P. C., and Huang, C. L.: Evolutionary Programming Based Economic Dispatch For Units With Non-Smooth Fuel Cost Functions, *IEEE Trans. on Power Systems*, Vol. 11, No. 1, pp. 112-118 Feb. 1996.