DEVELOPMENT OF AN EDUCATION SIMULATOR FOR PARTICLE SWARM OPTIMIZATION IN SOLVING ECONOMIC DISPATCH PROBLEMS

Mohd Hafiz Bin Mat Hussain

Bachelor of Electrical Engineering (Honours)
Faculty of Electrical Engineering
Universiti Teknologi MARA
40450 Shah Alam, Selangor

ABSTRACT

Development of an Educational Simulator presents a windows-based educational simulator using graphical user interface (GUI) for solving Economic Dispatch (ED) problems.The developed simulator incorporated particle swarm optimizations (PSO) to solve the ED problems. The objective of ED is to obtain the minimum generation cost with optimal power generator output while satisfying several constraints. In the developed simulator, users are able to set the parameters that have influences on particle swarm optimization performance.

Keywords- - Economic Dispatch (ED), Particle Swarm Optimization (PSO), windows-based educational simulator, Graphical User Interface (GUI)

1.0 INTRODUCTION

Optimization problems are widely encountered in various fields in science and technology. Sometimes such problems can be very complex due to the actual and practical nature of the objective function or the model constraints. Most of power system optimization problems have complex and nonlinear characteristics with heavy equality and inequality constraints. Recently, as an alternative to the conventional mathematical approaches, the heuristic optimization techniques such as genetic algorithms (GAs), Tabu search, simulated annealing, and particle swarm optimization are considered as realistic and powerful solution schemes to obtain the global or quasi-global optimums [1]. The practical economic dispatch problems with valve-point and multi-fuel effects are represented as a non-smooth optimization problem with equality and inequality constraints, and this makes the problem of finding the global optimum difficult. In order to solve this problem, many salient methods have been proposed such as a hierarchical numerical method [2], dynamic programming [3], evolutionary programming [4], Tabu search [5], neural network approaches [6], differential evolution [7], particle swarm optimization [8], and genetic algorithm [9].

Recent increases in the needs of the computer modeling and simulation tools have resulted in a heighted interest in computer simulation program. The motivation of the development of the simulator is to provide the undergraduate students with a simple and useable tool for gaining an intuitive feel for particle swarm optimization and power system optimization problems. To aid the understanding of particle swarm optimization, the simulator has been developed under the user-friendly graphic user interface (GUI) environment using MATLAB. Development of an Educational Simulator presents a windows-based educational simulator with user-friendly graphical user interface (GUI) for the education and training of particle swarm optimization technique for economic dispatch applications. The developed simulator incorporated particle swarm optimizations to solve economic dispatch problems. This project involved a study on The PSO Algorithm and solving the Economic Dispatch Problem. The development of Education simulator is divided into two main parts; the first part of the simulator development includes the development of PSO Algorithm and Economic Dispatch formulation. The PSO Algorithm and Economic Dispatch Problem were studied and developed. The second part of the simulator development, include the Graphical User Interface (GUI) to create simulation that incorporated particle swarm optimizations (PSO) to solve economic dispatch (ED) problems. The objective of

ED is to obtain the minimum generation cost with optimal power generator output while satisfying several constraints. This simulation received the parameter input from the user and simulates using PSO Algorithm to solve the Economic Dispatch Problem. The developed simulator displays the results with better quality solution in terms of fast computation time, convergence characteristic and generation cost minimization in ED can be achieved successfully.

2.0 THEORY

2.1 THE ECONOMIC DISPATCH PROBLEM

The main objective of ED is to minimize total generation cost so as to meet the load demand and satisfying the constraints.

The objective function of ED problem is given as:

$$F_T = \sum_{i=1}^{n} F_i(P_i) \quad (1)$$

Where the generator cost function is:

$$F_i(P_i) = \alpha_i + \beta_i P_i + \gamma_i P_i^{\,2} \quad (2)$$

Where:
F_T = total generation cost of overall power system.
$F_i(P_i)$ = cost function of every generator.
P_i = power output of each generator.
n = number of generator.
α_i, β_i and c_i the cost coefficients of i-th generator.

The constraints to be considered are:

a) *Maximum and minimum power generation limit.*

$$P_{i,min} \leq P_i \leq P_{i,max} \quad (3)$$

Where:
$P_{i,min}$ = minimum generator power output.
$P_{i,max}$ = maximum generator power output.

b) Power balance constraints.
The equality constraint to be considered is the power balance equation:

$$\sum (P_i) = P_D + P_L \quad (4)$$

Where,
$\sum (P_i)$ = total generation power output.
P_D = total load demand.
P_L = total power losses.

In this paper, total power losses are determined by using *loss coefficient* or *B – coefficient* method [10]. The equation is defined as:

$$P_L = \sum_{i=1}^{n}\sum_{j=1}^{n} P_i\, \beta_{ij} P_j + \sum_{i=1}^{n} \beta_{0i} P_i + \beta_{00} \quad (5)$$

Coefficients B_{ij} are loss coefficients and assumed constant.

2.2 THE PSO ALGORITHM

PSO iteratively searches for solutions in the problem space by taking advantage of the cooperative and competitive behavior of simple agents called particles. The PSO algorithm search is directed by its velocity equation [11]:

$$V_{id}^{(t+1)} = V_{id}^{(t)} + C_1(pBest - X_{id}^{(t)}) \times rand_1 + C_2(gBest - X_{id}^{(t)}) \times rand_2 \quad (6)$$

Which modifies the particle's position, X_{id}:

$$X_{id}^{(t+1)} = X_{id}^{(t)} + V_{id}^{(t+1)} \quad (7)$$

Where:
$V_{id}^{(t+1)}$ = updated particle velocity.
$V_{id}^{(t)}$ = velocity of ith particle at iteration t.
$X_{id}^{(t+1)}$ = updated particle position.
$X_{id}^{(t)}$ = current position of ith particle at iteration t.
t = pointer of iterations (generations) so far.
pBest = particle's best fitness so far.
gBest = best solution achieved by the swarm so far.
C_1, C_2 = acceleration constant.
$rand_1$, $rand_2$ = random numbers between 0 and 1.

There are two main variants that can be applied to the PSO velocity equation to enhance the optimization process. The two variants are called Inertia Weight (IW) and Constriction Factor (CF) [12]. When the inertia weight factor is applied in PSO, the velocity updating equation becomes:

$$V_{id}^{(t+1)} = w \times V_{id}^{(t)} + C_1(pBest - X_{id}^{(t)}) \times rand_1 + C_2(gBest - X_{id}^{(t)}) \times rand_2 \quad (8)$$

Where:
$V_{id}^{(t+1)}$ = updated particle velocity.
$V_{id}^{(t)}$ = velocity of ith particle at iteration t.
$X_{id}^{(t+1)}$ = updated particle position.
$X_{id}^{(t)}$ = current position of ith particle at iteration t.
t = pointer of iterations (generations) so far.
w = inertia weight factor.

$pBest$ = particle's best fitness so far.
$gBest$ = best solution achieved by the swarm so far.
C_1, C_2 = acceleration constant.
$rand_1$, $rand_2$ = random numbers between 0 and 1.

In general, the Inertia Weight factor, w is set according to the follow equation:

$$w = w_{max} - \frac{w_{max} - w_{min}}{iter_{max}} \times iter \qquad (9)$$

Where:
w_{max} = initial inertia weight factor.
w_{min} = ending inertia weight factor.
$iter_{max}$ = maximum number of iterations.
$iter$ = current number of iterations.
CF modifies the original PSO algorithm by gradually decreasing particle velocities as the iteration progresses, with the intention that particle movements near the optimum are localized. Using CF, the velocity equation becomes:

$$V_{id}^{(t+1)} = \chi[V_{id}^{(t)} + C_1(pBest - X_{id}^{(t)}) \times rand_1 + C_2(gBest - X_{id}^{(t)}) \times rand_2] \qquad (10)$$

The Constriction Factor (χ) is calculated by using:

$$\chi = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|} \qquad (11)$$

and φ must conform to:

$$\varphi = C_1 + C_2, \varphi > 4 \qquad (12)$$

Subsequent to new calculated velocities, X_{id} is updated similar to Equation (7).

3.0 METHODOLOGY

3.1 GRAPHICAL USER INTERFACE (GUI) PROGRAM DESIGN

The Graphical User Interface (GUI) program is designed in order to allow user to access the simulation. In this GUI program development, the first thing need to be done is to design the GUI window. The first GUI window will allow the user to choose the type of problem that the user need to solve. In this project, only Economic Dispatch problem are considered. Other option for the future development is unit commitment problem. Figure 1 shows the option for the user to choose.

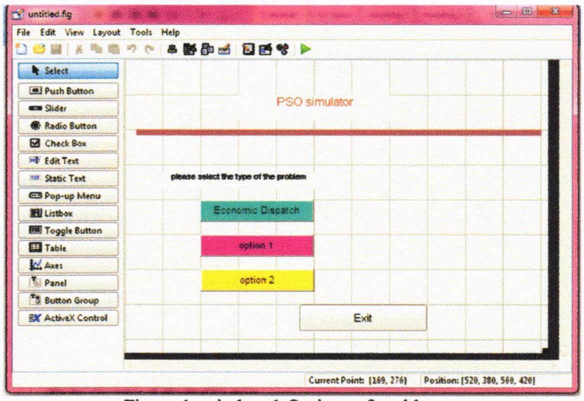

Figure 1: window 1 Options of problems

The second GUI window will allow the user to choose the type of method that the user want to use to solve the problem as stated in window 1. The options that can be choosing by the user are PSO, Option 1 and option 2. In this project, only PSO method is considered. Other option for future development includes Genetic Algorithm (GA) and Evolutionary Programming (EP). Figure 2 shows the option that the user can select.

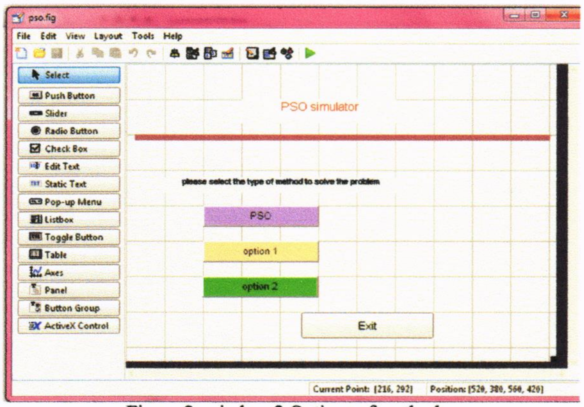

Figure 2: window 2 Options of methods

After user click the PSO button, the window for parameter setting will be appeared. Figure 3 shows PSO parameter setting menu.

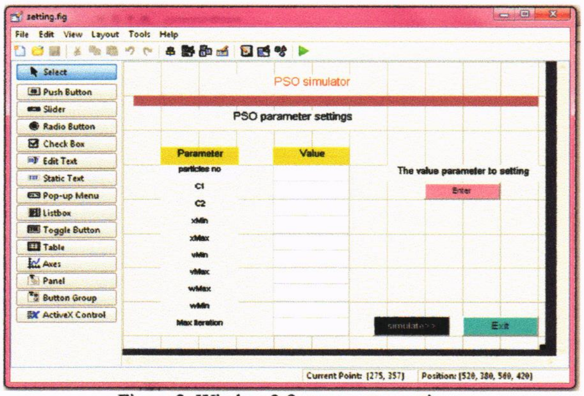

Figure 3: Window 3 for parameter setting.

As the user clicks the simulate button, the window for the result will be shown. If the user click the Exit button, the conformation box will be appear before the window close.

Figure 4 shows the flowchart in developing the GUI program.

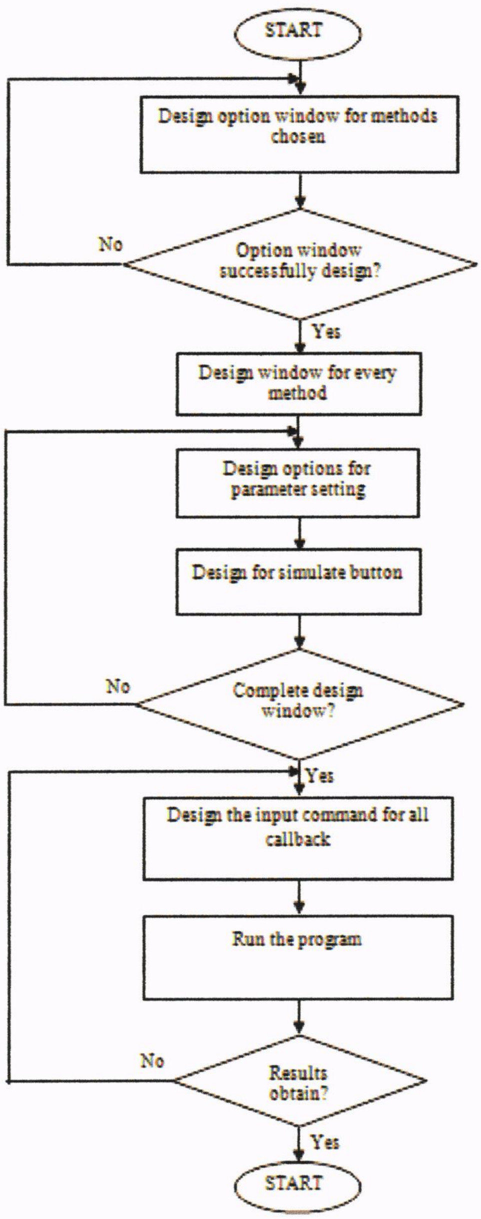

Figure 4: GUI Program Design flowchart

3.2 PSO ALGORITHM DESIGN

The PSO algorithm first needs to be initialized with random values for the generator output. The random values are normally distributed to ensure that the particles are well-distributed across the problem space. The initial particle velocities are also generated randomly as well. After that, the random value of each generator output power will be optimized by Particle Swarm Optimization (PSO) algorithm by updating each of particles velocities and positions by applying PSO algorithm formula in Equation (10) and (7) respectively.

The next process is to evaluate the fitness value of each particle in process to meet the best optimal value of power generator output. The power limit of each generator output should be considered by satisfying constraint in Equation (3). The total power loss is calculated by applying the *B-coefficient* method in Equation (5). Consider the power balance constraint of the system by using Equation (4). Check the power balance value which must be not lesser than total power demand of the system. If the demand is not satisfied, the solution is deemed to be unfit, and punished by returning an extremely bad fitness value. If the power demand is met, then determine total generation cost of the system.

Applying Equation (2), the total generation cost value of the system is calculated for each particle. The fitness value is then returned for further optimization for the PSO algorithm. The optimization will continue to iteratively optimize the variables in the particles until the maximum number of iterations is reached, or until the objective is met. The flowchart shown in Figure 5 shows the solution process for economic dispatch problem using PSO.

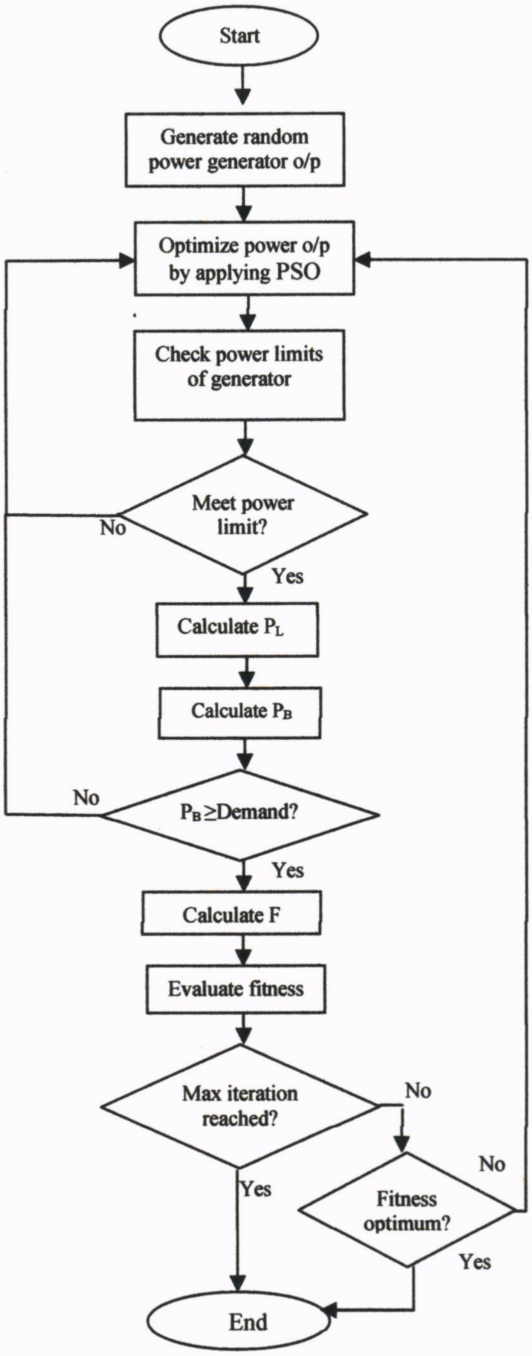

Figure 5: Economic Dispatch solution using PSO flowchart

3.3 PSO PSEUDO CODE

The pseudo-code for PSO is given:

1) Randomly generate particles initial velocities and positions within their limits.
2) Then, *pBest* is set to each initial searching point. The best evaluated values among pBest is set as *gBest*.
3) Update new velocities, $V_{id}^{(t+1)}$ of each particles by applying Equation (6).
4) If $V_{id}^{(t+1)} < V_{dmin}$, then declare $V_{id}^{(t+1)} = V_{dmin}$ if $V_{id}^{(t+1)} > V_{dmax}$, then declare $V_{id}^{(t+1)} = V_{dmax}$.
5) Update new position, $X_{id}^{(t+1)}$ of each particles by applying Equation (7).
6) Then check the capacity limit constraints. If $X_{id}^{(t+1)} > X_{dmax}$, then declare $X_{id}^{(t+1)} = X_{dmax}$, if $X_{id}^{(t+1)} < X_{dmin}$, then declare $X_{id}^{(t+1)} = X_{dmin}$.
7) Then, evaluate fitness values for new particle position. If the evaluation value of each individual is better than the previous *pBest*, then set the current value as new *pBest*. Then, if the best *pBest* is better than *gBest*, the current value is set to be new *gBest*.
8) If the number of iterations reaches the maximum, then go to the next optimization step. Otherwise, repeat the optimization process at Step 3.
9) Stop of the optimization process (The individual that generates the latest *gBest* is the optimal generation power of each unit with the minimum total generation cost.

4.0 RESULTS AND DISCUSSION

4.1 PSO PARAMETER SETTINGS

For the purposes of this project, the parameter for PSO is set as in Table 1.

Table 1: The value of PSO parameter settings

Parameter	Value
Particles no	10 - 100
C_1	2.05
C_2	2.05
xMin	0
xMax	1
vMin	-1
vMax	1
wMax	0.9
wMin	0.4
Max Iteration	1000

The particle values were changed to evaluate the convergence at different swarm sizes. Theoretically, as the swarm size increases, the solution should be obtained quicker since more particles are searching the problem space. Values C_1 and C_2 are set to 2.05 to balance between individual and swarm influences during optimization. The values have to consider the constraint set by Equation (8), (10) respectively.

The values of xMin and xMax prevent the particles from searching in unfeasible areas. If the particles violate the constraints set by xMin and xMax, they will be pulled back to xMin or xMax, and their velocity set to zero to prevent any more search in that direction. Consequently, the values of vMin and vMax are set to reflect the maximum movement range of xMin and xMax.The values of wMax and wMin were set according to the recommendation by [12]. The maximum iterations were set to 1000, since preliminary tests indicated that iterations beyond this number would yield insignificantly better results at the cost and optimization time.

4.2 UNIT GENERATION PARAMETER

The power system considered in this project consists of six unit generator. Data for the generating unit are tabulated in Table2 [13].

Table 2: Generating coefficients and unit capacity.

Unit	P_i^{max} (MW)	P_i^{min} (MW)	α ($)	β ($/MW)	γ ($/MW^2)
1	500	100	240	7.0	0.0070
2	200	50	200	10.0	0.0095
3	300	80	220	8.5	0.0090
4	150	50	200	11.0	0.0090
5	200	50	220	10.5	0.0080
6	120	50	190	12.0	0.0075

The *B - loss coefficient* matrixes of the loss formula for this system are given as [13]:

$$B_{ij} = \begin{bmatrix} 0.0017 & 0.0012 & 0.0007 & -0.0001 & -0.0005 & -0.0002 \\ 0.0012 & 0.0014 & 0.0009 & 0.0001 & -0.0006 & -0.0001 \\ 0.0007 & 0.0009 & 0.0031 & 0.0000 & -0.0010 & -0.0006 \\ -0.0001 & 0.0001 & 0.0000 & 0.0024 & -0.0006 & -0.0008 \\ -0.0005 & -0.0006 & -0.0010 & -0.0006 & 0.0129 & -0.0002 \\ -0.0002 & -0.0001 & -0.0006 & -0.0008 & -0.0002 & 0.0150 \end{bmatrix}$$

$$B_{0i} = 1.0e^{-03} * [-0.3908\ -0.1297\ 0.7047\ 0.0591\ 0.2161\ -0.6635\ ]$$

$$B_{00} = 0.0056$$

4.3 PROGRAM RUNNING

This program designed to run together with PSO Algorithm design. The user needs to set all the parameter. Figure 6 show the window of the parameters for PSO.

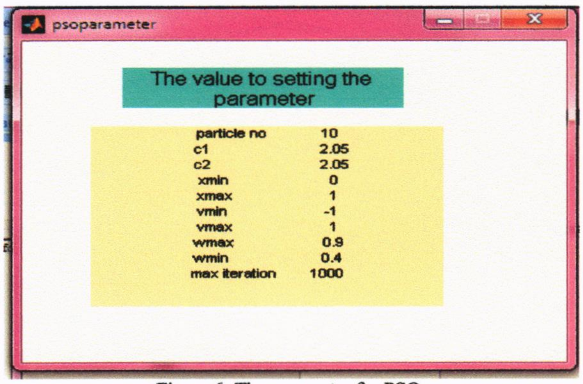

Figure 6: The parameter for PSO

After the user clicks the Simulate button, the result window will appear. Figure 7 shows the result window.

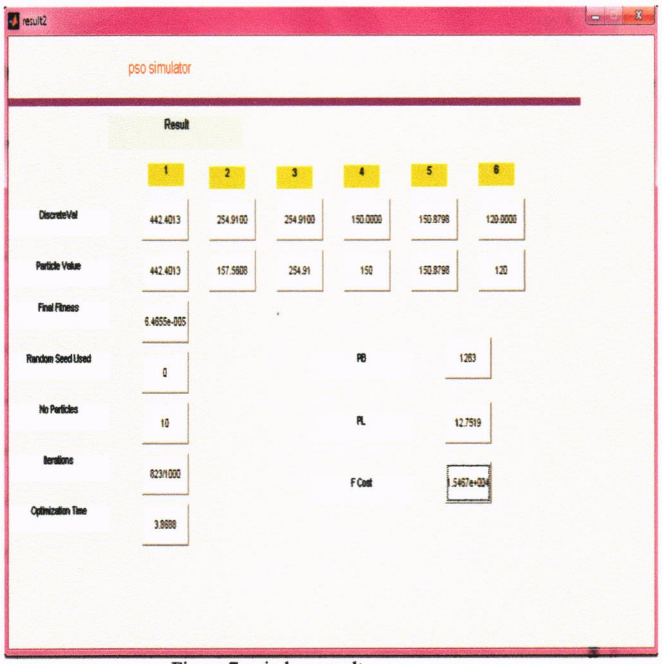

Figure 7: window result

5.0 CONCLUSION

As a conclusion, this paper presents a development of an educational simulator for particle swarm optimization (PSO) and application for solving ED problems. In the simulator, users set the parameters that have an influence on the PSO performance. The developed simulator is expected to be a useful tool for students who study electrical engineering and optimization techniques.

6.0 FUTURE DEVELOPMENT

This project can be further developed to allow the user to use the simulator to solve ED problem using other method include Ant Colony Optimization (ACO), Genetic Algorithm (GA) and Evolutionary Programming (EP). This simulator also can be further developed to incorporated unit commitment problems.

7.0 REFERENCES

[1] K. Y. Lee and M. A. El-Sharkawi, Eds. *Modern Heuristic Optimization Techniques with Applications to Power Systems*: IEEE Power Engineering Society (02TP160), 2002.

[2] C. E. Lin and G. L. Viviani, "Hierarchical economic dispatch for piecewise quadratic cost functions," *IEEE Trans. Power App. Syst.,* vol. PAS-103, no. 6, pp. 1170-1175, June 1984.

[3] A. J. Wood and B. F. Wollenbergy, *Power Genration, Operation, and Control.* New York: Wiley, 1984.

[4] Y. M. Park, J. R. Won, and J. B. Park, "A new approach to economic load dispatch based on improved evolutionary programming," *Eng. Intell. Syst. Elect. Eng. Commun.,* vol. 6, no. 2, pp. 103-110, June 1998.

[5] W. M. Lin, F. S. Cheng, and M. T. Tasy, "An improved Tabu search for economic dispatch with multiple minima," *IEEE Trans. Power Syst.,* vol. 17, pp. 108-112, Feb. 2002.

[6] J. H. Park, Y. S. Kim, I. K. Eom, and K. Y. Lee, "Economic load dispatch for piecewise quadratic cost function using Hopfield neural network," *IEEE Trans. Power Syst.,* vol. 8, pp. 1030-1038, Aug. 1993.

[7] L. S. Coelho and V. C. Mariani, "Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect," *IEEE Trans. Power Syst.,* vol. 21, No. 2, May 2006.

[8] J. B. Park, K. S. Lee, J. R. Shin, and K. Y. Lee, "A particle swarm optimization for economic dispatch with nonsmooth cost functions," *IEEE Trans. Power Syst.,* vol. 20, no. 1, pp. 34-42, Feb. 2005.

[9] D.C. Walters and G. B. Sheble, "Genetic algorithm solution of economic dispatch with the valve point loading," *IEEE Trans. Power Syst.,* vol. 8, pp. 1325-1332, Aug. 1993.

[10] Hadi Saadat. Power System Analysis, Second Edition. 2004.

[11] Y. Shi and R.Eberhart,"A modified particle swarm optimizer," Proc.IEEE Int. Conf. Evol. Comput. ,pp. 69-73, May 1998.

[12] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in Proc. Congr. Evol. Compt.vol. 1, pp. 84 - 88, Jul. 2000.

[13] Zwe-Lee Gaing, " Particle swarm optimization to solving the Economic dispatch considering the generator constraints," *IEEE Trans. Power Syst.,* vol. 18, pp. 1187-1195, Nov. 2003.