Self-Organizing Hierarchical Particle Swarm Optimization for Large-Scale Economic Dispatch with Multiple Fuels Considering Valve-Point Effects

Le D. Luong[1], P. Vasant[2], Vo N. Dieu[3], Truong H. Khoa[2], and Doan V. K. Khanh[2]

[1]*Faculty of Mechanical - Electrical - Electronic HCMC University of Technology, Vietnam*
[2]*Universiti Teknologi Petronas, Malaysia*
[3]*Department of Power Systems, HCMC University of Technology, Vietnam*
Email: ledinhluong@gmail.com, pvasant@gmail.com, and vndieu@gmail.com

Abstract. This paper proposes a self-organizing hierarchical particle swarm optimization (SOH-PSO) algorithm for solving economic dispatch with multiple fuels (EDMF) and valve point effects. The PSO algorithm is simulating the behavior of birds or fish to find food. Each individual changes in velocity and position based on its experience improvement of itself and experience of both swarm. The PSO algorithm has been applied to solve many economic dispatch problems in power systems. In the new improved method, the conventional PSO algorithm is used with the variance coefficients to speed up the convergence to the global solution in a fast manner regardless of the shape of the cost function. The proposed SOH-PSO has been tested on various systems and the obtained numerical results have shown that the SOH-PSO method is more efficient and faster than many other methods reported in the literature for finding the optimal solution of EDMF. Therefore, the proposed SOH-PSO method can be a promising method for solving the practical EDMF problems.

Keywords: Economic dispatch; multiple fuels; self-organizing hierarchical particle swarm optimization.
PACS: 07.05.Mh

INTRODUCTION

The operation cost in power systems needs to be minimized at each time satisfying constraints via economic dispatch (ED) problem [1]. In practical power system operation conditions, many thermal generating units, especially those units which are supplied with multiple fuel sources like coal, natural gas, and oil require that their fuel cost functions may be segmented as piecewise quadratic cost functions to represent for different types of fuel. The ED problem with piecewise quadratic cost functions is to minimize total fuel cost among the available fuels for each unit satisfying load demand and generation limits. This is a non-convex and complicated optimization problem since it contains the discontinuous values at each boundary forming multiple local optimal. Therefore, the classical solution methods are difficult to deal with this problem. One approach for solving the problem with such units having multiple fuel options is linearization the segments and solving them by traditional methods [2]. A better approach is to retain the assumption of piecewise quadratic cost functions and proceed to solve them. A hierarchical approach based on the numerical method (HNUM) has been proposed in [3] as one way to approach to the problem. However, the major problem for the numerical methods is their exponentially growing time complexities for larger systems with non-convex constraints. Recently, many methods have been applied to solve the problem of multi-fuels economic dispatch such as the numerical method (HNUM) [4], the Hopfield Neural Network (HNN) [5], Enhanced Lagrangian Artificial Neural Network (ELANN) [6], Adaptive Hopfield Neural Network (AHNN) [7], Real Coded Genetic Algorithm (RCGA) [8], Hybrid Real Coded Genetic Algorithm (HRCGA) [8], Evolutionary Programming (EP), Conventional Evolutionary Programming (CEP) [9], Improved Evolutionary Programming (IEP) [10], Tabu search and Quadratic programming (ETQ) [11], Modified Particle Swarm Optimization (MPSO) [12], Improved Genetic Algorithm with the Adaptive Multiplier Updating Method (IGA AMUM) for power economic dispatch of units having multiple fuel options [13], and Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients [14]. However these methods are large number of iteration and easily influenced by parameters related controls. Recently, appeared PSO algorithm, this algorithm has several advantages compared to other methods of computational time faster and stable convergence. Scientists have applied PSO algorithm in many different areas of power system analysis such as system stability, coordination capacity, power system operation, etc and the PSO algorithm has produced good results than other methods.

This paper applies the advanced PSO algorithm to solve the large-scale economic dispatch problem with multi-fuel economic dispatch with valve point effects. The advanced PSO method is tested and validated by comparing results with other methods such as particle swarm optimization with time-varying inertia weight factor (PSO-TVIW), particle swarm optimization with time-varying acceleration coefficients (PSO-TVAC), improved genetic algorithm with multiplier updating (IGA_MU) [16], conventional genetic algorithm (CGA) with the MU (CGA_MU) [16].

PROBLEM FORMULATION

The main objective of the EDMF problem is to minimize total cost of thermal power plants with many different fuels satisfying many different operating constraints including power balance and the limited capacity of the generating units. Therefore, it can be mathematically modeled as an objective function with equality and inequality constraints.

Consider a system with N plants and each plant generates a P_i MW of capacity. The capacity of plants should be scheduled so that their total cost F is minimized. Mathematically, the problem is formulated as follows:

$$\mathrm{Min}\, F = \sum_{i=1}^{N} F_i(P_i) \quad (1)$$

where the fuel cost function of each generating unit is represented by:

$$F_i(P_i) = \begin{cases} a_{i1} + b_{i1}P_i + c_{i1}P_i^2 + \left|e_{i1}.\sin(f_{i1}.(P_{i1}^{\min} - P_{i1}))\right|, \ forfuel1, P_i^{\min} \le P_i \le P_{i1} \\ a_{i2} + b_{i2}P_i + c_{i2}P_i^2 + \left|e_{i2}.\sin(f_{i2}.(P_{i2}^{\min} - P_{i2}))\right|, \ forfuel2, P_{i1} \le P_i \le P_{i2} \\ \vdots \\ a_{ik} + b_{ik}P_i + c_{ik}P_i^2 + \left|e_{ik}.\sin(f_{ik}.(P_{ik}^{\min} - P_{ik}))\right|, \ forfuelk, P_{ik-1} \le P_i \le P_i^{\max} \end{cases} \quad (2)$$

subject to

1. Power balance constraint

$$\sum_{i=1}^{N} P_i - P_L - P_D = 0 \quad (3)$$

where the power loss is approximately calculated by Kron's formula [15]:

$$P_L = \sum_{i=1}^{N}\sum_{j=1}^{N} P_i B_{ij} P_j + \sum_{i=1}^{N} B_{0i} P_i + B_{00} \quad (4)$$

2. Generator operating limits

$$P_{i,\min} \le P_i \le P_{i,\max} \ ; \ i = 1,..., N \quad (5)$$

where

$F_i(P_i)$	Fuel cost function of generating unit i
a_{ik}, b_{ik}, c_{ik}	Cost coefficients for fuel cost function k of unit i
B_{ij}, B_{0i}, B_{00}	Transmission loss formula coefficients
N	Number of online generating units
P_D	Total load demand of the system (MW)
P_L	Total network loss of the system (MW)
P_i	Output power of unit i (MW)
$P_{i,min}$, $P_{i,max}$	Lower and upper generation limits of unit i (MW)

With this formulation, the ED problem with multiple fuels becomes a non-convex optimization problem with multiple minima. For obtaining optimal solution for this problem, solution methods have to search for optimal solution in a very large search space, leading to time consuming. Therefore, it is necessary to limit the search space of the problem to reduce computational time, especially for large-scale systems.

SOH-PSO FOR EDMF

CONVENTIONAL PSO

In PSO algorithm, the individual in the swarm approaches the target through its optimal speed, previous experience of itself and neighbor individual experience. In the search space of d-dimensional, the position and velocity of individual i is described by vectors $X_i = (x_{i1}, \ldots, x_{id})$ and $V_i = (v_{i1}, \ldots, v_{id})$, respectively. $Pbest_i = (x_{i1}^{Pbest}, ..., x_{id}^{Pbest})$ is the best location for the current instance i and $Gbest_i = (x_{i1}^{Gbest}, ..., x_{id}^{Gbest})$ is the best location of the swarm.

Consider a swarm with p individuals in d-dimensional space. Position vector X_i^k of each individual i is updated by the following expression:

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (6)$$

where V_i^{k+1} is a new velocity calculated by the formula:

$$V_i^{k+1} = \omega V_i^k + c_1 rand_1 \times \left(Pbest_i^k - X_i^k\right) + c_2 rand_2 \times \left(Gbest^k - X_i^k\right) \quad (7)$$

where

X_i^k	position of individual i at iteration k
X_i^{k+1}	position of individual i in the iteration k+1
V_i^k	velocity of individual i in the iteration k

V_i^{k+1} velocity of individual *i* in the iteration *k*+1
ω constant weight inertia
c_1 individual experience coefficient
c_2 social relations coefficient of individual
$rand_1$, $rand_2$ random numbers between [0, 1]
$Pbest_i^k$ best position of individual *i* in the iteration *k*
$Gbest^k$ best position of swarm in the iteration *k*.

Expressions (6) and (7) show the principle search of PSO algorithm by using the change velocity and position of the individual. In particular, X_i^k is the position of individual *i* in iteration *k*, we need to determine the position of individual *i* at the next iteration X_i^{k+1}.

To determine X_i^{k+1}, the value of V_i^{k+1} needs to be calculated in advance. The vector V_i^{k+1} consists of three components. The first one, ωV_i^k, shows the inertial searching of the individual. During the search process, each individual tends to follow the inertia of the previous searches; the second one, $c_1 rand_1 \times \left(Pbest_i^k - X_i^k\right)$, shows the individual experience based on the previous search, toward the best position $Pbest_i^k$; and the last one, $c_2 rand_2 \times \left(Gbest^k - X_i^k\right)$, shows the ability to communicate by learning the best individual in the swarm, toward the best position of the swarm has been to present $Gbest^k$. Synthesis of the three-element vector above, the vector V_i^{k+1} is obtained for the velocity vector of individual *i* in the iteration *k*+1. After obtaining the velocity vector V_i^{k+1}, it is combined with the position vector X_i^k at iteration *k* to obtain the position vector X_i^{k+1} of individual *i* in the iteration *k* +1.

PSO with Time-Varying Inertia Weight Factor

The expression velocity vector update function of the individual is shown as follows:

$$V_i^{k+1} = \omega_{new} V_i^k + c_1 rand_1 \times \left(Pbest_i^k - X_i^k\right) + c_2 rand_2 \times \left(Gbest^k - X_i^k\right) \quad (8)$$

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{Iter_{\max}} k \quad (9)$$

$$\omega_{new} = \omega_{\min} + \omega \times rand_3 \quad (10)$$

where

ω_{new} inertia weight factors
ω_{max}, ω_{min} maximum and minimum inertia weight factors
k current iteration
$Iter_{max}$ maximum number of iterations
$rand_1$, $rand_2$, $rand_3$ random numbers in [0,1].

The function for updating the position remains the same as the basic PSO algorithm.

By experiment, Shi and Eberhart [2] found that the optimal solution can be improved by changing the inertia coefficient from 0.9 during the search to 0.4 at the end the search. The overall procedure of the PSO-TVIW is presented as follows:

Step 1: Created swarm includes all elements with position and velocity of the random *d*-dimensional search space.

Step 2: Calculate objective function value of each element.

Step 3: Comparing the objective function values of the elements to those from the previous iteration. For each individual, if the value of the objective function value at the current iteration is better than that from the previous iteration, the position corresponding to the current objective function will set to P_{besti}. Otherwise, the position in the previous iteration is set to P_{besti}.

Step 4: Recognize elements in the value swarm best objective function and objective function value assigned to the variable G_{best}. Store the values of P_{besti} and G_{best}.

Step 5: Change the velocity and position of the element by the expression:

$$V_i^{k+1} = \omega_{new} V_i^k + c_1 rand_1 \times \left(Pbest_i^k - X_i^k\right) + c_2 rand_2 \times \left(Gbest^k - X_i^k\right) \quad (11)$$

with:

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{Iter_{\max}} k \quad (12)$$

$$\omega_{new} = \omega_{\min} + \omega rand_3 \quad (13)$$

$$X^{k+1} = X^k + V^{k+1} \quad (14)$$

Step 6: If the stopping criteria are not met, increase the iteration counter and return to Step 2. Otherwise, stop.

PSO WITH TIME-VARYING ACCELERATION COEFFICIENTS

PSO with time-varying acceleration coefficients (PSO-TVAC) is another improvement of the conventional PSO. In the PSO-TVAC, the experience factors of individual and society will change with respect to the number of iterations. The overall procedure of the PSO-TVAC is addressed as follows:

Step 1: Created swarm includes all the elements with position and velocity of the random *d*-dimensional search space.

Step 2: Calculate objective function value of each element.

Step 3: Comparing the objective function values of the elements to those from the previous iteration. For each individual, if the value of the objective function value at the current iteration is better than that from the previous iteration, the position corresponding to the current objective function will set to P_{besti}. Otherwise, the position in the previous iteration is set to P_{besti}.

Step 4: Recognize elements in the value swarm best objective function and objective function value assigned to the variable G_{best}.

Step 5: Change the velocity and position of the element in the expression:

$$V_i^{k+1} = \omega V_i^k + c_1 rand_1 \times \left(Pbest_i^k - X_i^k\right) + c_2 rand_2 \times \left(Gbest^k - X_i^k\right) \quad (15)$$

$$c_1 = (c_{1\max} - c_{1\min}).\frac{k}{Iter_{\max}} + c_{1\min} \quad (16)$$

$$c_2 = (c_{2\max} - c_{2\min}).\frac{k}{Iter_{\max}} + c_{2\min} \quad (17)$$

where

c_{1max}, c_{1min} maximum and minimum individual experience coefficient

c_{2max}, c_{2min} maximum and minimum social relations coefficient of individual

ω constant weight inertia

$$X^{k+1} = X^k + V^{k+1} \quad (18)$$

Step 6: If the stopping criteria are not met, increase the iteration counter and return to Step 2. Otherwise, stop.

PSO WITH SELF ORGANIZING HIERARCHICAL

As we know, most of the improved PSO algorithms have been based on the linear change of the inertia coefficient and the penalty coefficient method. However, in some complex functions, controlling the diversity of the population coefficient changes will cause the individual may locally converge to the optimal solution soon. In this paper, we proposed the improved PSO algorithm does not need the velocity of the previous iteration. We found this algorithm is simple but very effective when solving optimization problems for complex problems. The expression velocity vector update function of the individual is shown as follows

$$V_i^{k+1} = c_1 rand_1 \times \left(Pbest_i^k - X_i^k\right) + c_2 rand_2 \times \left(Gbest^k - X_i^k\right) \quad (19)$$

$$c_1 = (c_{1\max} - c_{1\min}).\frac{k}{Iter_{\max}} + c_{1\min} \quad (20)$$

$$c_2 = (c_{2\max} - c_{2\min}).\frac{k}{Iter_{\max}} + c_{2\min} \quad (21)$$

where

c_{1max}, c_{1min} maximum and minimum individual experience coefficient

c_{2max}, c_{2min} maximum and minimum social relations coefficient of individual

$$X^{k+1} = X^k + V^{k+1} \quad (22)$$

Implementation of SOH-PSO to EDMF

The objective function of the problem is:

$$\mathrm{Min}\, F = \sum_{i=1}^{N} F_i(P_i) \quad (23)$$

Neglecting transmission losses, power balance constraint is formulated by:

$$\sum_{i=1}^{N} P_i = P_D \quad (24)$$

By using the slack variable method, the capacity of the unit *N* is calculated as follow:

$$P_N = P_D - \sum_{i=1}^{N-1} P_i \quad (25)$$

The proposed SOH-PSO for the problem has been explained in detail in [3].

The overall procedure of SOH-PSO for the EDMF is as follows:

Step 1: Determine the number of plants in the power system. Determine the number of fuel of each plant, maximum capacity and minimum capacity of each plant, and the cost coefficient of each plant.

Step 2: Setting the initial parameters for the PSO algorithm: Number of individuals in the swarm, maximum number of iterations $Iter_{max}$, maximum and minimum inertia coefficient (ω_{max} and ω_{min}), maximum and minimum experience coefficient (c_{1i},

c_{1f}), maximum and minimum coefficient of social relation (c_{2i}, c_{2f}).

Step 3: Created position vector x_i and velocity vector v_i of the individual, with i = 1, …, d is the number of individuals in the swarm.

$$x_i = [\, P_{i1}, P_{i2}, \dots, P_{i,N-1}, P_N \,]$$
$$v_i = [\, v_{i1}, v_{i2}, \dots, v_{i,n-1}]$$

where P_{ik} is a generation capacity of the plants, with k = 1,…, n_i (n_i is the number of fuels for unit i). P_N is calculated by the expression (25).

Step 4: Calculated the total cost of fuel $F(x_i)$ with i = 1,…, d. Calculated function fitness.

Step 5: Assigning value $Pbest_i = x_i$, find the value and position Gbest.

Step 6: Setting iteration counter $t = 1$.

Take steps from 7 to 12 for each individual $i = 1, \dots, d$

Step 7: Update v_{id}

Using expression (11), (12) and (13) if we want to use PSO-TVIW method

Using expression (15), (16) and (17) if we want to use PSO-TVAC method

Using expression (19), (20) and (21) if we want to use SOH-PSO method

Step 8: Update x_{id} by expression (6)

With i is number of plants and d number of individuals in the swarm.

Check the capacity limits of the plants. If $x_{id} > x_{idmax}$ then $x_{id} = x_{idmax}$, if $x_{id} < x_{idmin}$ then $x_{id} = x_{idmin}$.

Step 9: Calculated value P_N by expression (25)

Step 10: Calculated total cost of the fuel $F(x_i)$ and update $Pbest_i$, Gbest if objective function value is better than old objective function value

Step 11: Increase number of iteration counter $t = t+1$.

Step 12: Considering the conditions stop the program if $Iter > Itert_{max}$. Otherwise return to step 7.

NUMERICAL RESULTS

To validate the effectiveness of proposed advanced PSO method, three improved PSO algorithm (PSO-TVIW, TVAC PSO, SOH-PSO) was tested on 10, 20, 40, 80, 160 plant systems. Each plant has cost function with multiple fuels and valve point effects. These algorithms are implemented in Matlab 9.0 and run on a Dell Studio Laptop Core (TM) 2 Duo CPU T6400@2.0 GHz, Ram 4G. Stopping criterion is the maximum number of iterations.

- *Setting parameter for algorithm PSO-TVIW*

Number of iterations: $Iter_{max}$= 500

Number of individuals in the swarm: d = 25

Maximum acceleration coefficient: ω_{max}= 0.9

Minimum acceleration coefficient: ω_{min}= 0.4

Individual experience coefficient: c_1= 2.3

The coefficient of social relations of individual: c_2= 0.5

- *Setting parameter for algorithm PSO-TVAC*

Number of iteration: $Iter_{max}$= 500

Number of individuals in the swarm: d = 25

Acceleration coefficient: ω = 0.75

Minimum individual experience coefficient: c_{1min} = 0.2

Maximum individual experience coefficient: c_{1max} = 2.5

The minimum coefficient of social relations of individual: c_{2mim} = 0.2

The maximum coefficient of social relations of individual: c_{2max} = 2.5.

- *Setting parameter for algorithm SOH-PSO*

Number of iterations: $Iter_{max}$ = 500

Number of individuals in the swarm: d = 25

Minimum individual experience coefficient: c_{1min} = 0.2

Maximum individual experience coefficient: c_{1max} = 2.5

The minimum coefficient of social relations of individual: c_{2mim} = 0.2

The maximum coefficient of social relations of individual: c_{2max} = 2.5.

Case study 1: 10-unit power systems with multiple fuels and valve point effects.

To verify the feasibility of the proposed SOH-PSO method, the 10-unit system [16] were tested. The input data and the cost coefficients for 10-generating units are given in [16]. The total demanded load P_D of this problem is 2700 MW.

The optimal dispatch of the generators is listed in Table 1. Table 2 shows the minimum, mean, maximum cost achieved by the PSO-TVIW, PSO-TVAC and SOH-PSO algorithm in 100 runs. Three improved PSO algorithm has succeeded in finding a global optimal solution. The optimum active power is in their secure values and is far from the min and max limits. It is also clear from the optimum solution that the PSO-TVIW, PSO-TVAC and SOH-PSO easily prevent the violation of all the active constraints. The results obtained from the SOH-PSO are compared to those from CGA-MU, IGA-MU, PSO-TVAC, and PSO-TVIW as given in Table 3. As seen in Table 3, SOH-PSO method provide a minimum total cost less than that from other methods including CGA_MU [16] and IGA_MU [16]. The computation time of the SOH-PSO method is also faster than TVIW PSO, PSO-TVAC, CGA_MU [16], ad IGA_MU [16].

TABLE 1. Costs and capacity of plant in Case study 1

Method	Plant	2700 MW			
		P_i(MW)	Fuel	Segment	Cost
PSO TVIW	1	218.129	2	2	623.8444
	2	209.955	1	3	
	3	278.615	1	1	
	4	240.355	3	3	
	5	276.438	1	1	
	6	236.967	3	3	
	7	286.051	1	1	
	8	240.089	3	3	
	9	437.98	3	3	
	10	275.421	1	1	
PSO TVAC	1	219.134	2	2	623.8399
	2	209.679	1	3	
	3	278.636	1	1	
	4	239.015	3	3	
	5	276.5	1	1	
	6	238.855	3	3	
	7	287.743	1	1	
	8	241.299	3	3	
	9	434.88	3	3	
	10	274.26	1	1	
SOH-PSO	1	219.133	2	2	623.8362
	2	209.432	1	3	
	3	282.673	1	1	
	4	239.821	3	3	
	5	279.768	1	1	
	6	238.989	3	3	
	7	287.727	1	1	
	8	239.418	3	3	
	9	424.111	3	3	
	10	278.928	1	1	

TABLE 2. Costs and processing time in Case study 1

Method	Min cost ($)	Max. cost ($)	Avg. cost ($)	Time (s)
PSO-TVIW	623.8444	624.3465	623.942159	3.92
PSO-TVAC	623.8399	624.1303	623.923907	3.95
SOH-PSO	623.8362	624.1799	623.910038	3.8

TABLE 3. Comparison of total cost and processing time in Case study 1

Method	P (MW)	Total cost ($/h)	Time (s)
CGA_MU [16]	2700	624.7193	7.25
IGA_MU [16]	2700	624.5178	26.17
PSO-TVIW	2700	624.8444	3.92
PSO-TVAC	2700	623.8399	3.95
SOH-PSO	2700	623.8362	3.8

Case study 2: Large-scale power systems with multiple fuels and valve point effects.

The algorithms of PSO-TVIW, PSO-TVAC and SOH-PSO have been tested in large-scale systems including 20, 40, 80, 160 plants with total demand load is 2700*(N/10) MW [16], where N is the number of plants. Each plant has fuel costs with multiple fuels and valve point effects. Large-scale power systems are tested system based on 10 plants in case study 1 by multiplying the number of units. Results obtained from the PSO method are compared with other methods CGA_MU [16] and IGA_MU [16].

Table 4 shows the best costs, computational times, and the number of units for the systems with 20, 40, 80, 160 units. We have observed that the SOH-PSO approach for solving the economic dispatch problem with multiple fuels and valve point effects have obtained better results than the other methods. Therefore, the proposed SOH-PSO method has the advantages over other methods for faster computation times, larger-scale power system implementation, and more stable convergence.

TABLE 4. Comparison of total cost and processing time in Case study 2

Method	Number of units (N)	Cost ($/h)	Time (s)
CGA_MU [16]	20	1249.3893	80.48
	40	2500.9220	157.39
	80	5008.1426	309.41
	160	10143.7263	621.30
IGA_MU [16]	20	1249.1179	21.64
	40	2499.8243	43.71
	80	5003.8832	85.67
	160	10042.4742	174.62
PSO-TVIW	20	1247.8481	16.35
	40	2499.7118	17.88
	80	5042.7687	21.51
	160	10145.1804	28.89
PSO-TVAC	20	1248.132	16.4
	40	2500.5642	17.9
	80	5029.3577	21.55
	160	10201.8615	28.91
SOH-PSO	20	1247.8839	16.45
	40	2496.045	17.94
	80	5000.524	21.65
	160	10038.533	28.96

CONCLUSION

In this paper, the self-organizing hierarchical particle swarm optimization (SOH-PSO) algorithm has been presented to solve the economic dispatch problem with multiple fuels (EDMF) and valve point effects. In the new improved method, the conventional PSO algorithm is used with the variant coefficients to speed up the convergence to the global solution in a fast manner regardless of the shape of the cost function. In the improved method of PSO algorithm, SOH-PSO method can obtain the best results and the fastest computational times compared to PSO-TVAC, PSO-TVIW, CGA-MU, and IGA-MU for the test systems, especially for large systems. The advantages of SOH-PSO method are conceptually simple, easy to implement, and better convergence of the previous method. Therefore, the proposed SOH-PSO has a great potential for applying to large-scale optimization problems in electrical power systems.

ACKNOWLEDGMENTS

This research paper is supported by ERGS research grant of UTP and MOHE of Malaysia.

REFERENCES

1. A.J. Wood and B.F. Wollenberg, *Power Generation, Operation, and Control*, 2nd ed., John Wiley, New York, 1996.
2. R. C. Eberhart and Y. Shi, "Comparison between genetic algorithms and particle swarm optimization", *Evolutionary Programming VII: Proc. 7th Ann. Conf. on Evolutionary Programming Conf.*, San Diego, CA, 1998.
3. Asanga Ratnaweera, Saman K. Halgamuge, and Harry C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients", *IEEE Trans. Evolutionary Computation* **8**, 240-255 (2004).
4. C. E. Lin and G. L. Viviani, "Hierarchical economic dispatch for piecewise quadratic cost functions", *IEEE Trans. Power Apparatus and Systems* **PAS-103**, 1170 - 1175 (1984).
5. 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 Systems* **8**, 1030-1038 (1993).
6. S. C. Lee and Y. H. Kim, "An enhanced Lagrangian neural network for the ELD problems with piecewise quadratic cost functions and nonlinear constraints" *Electric Power Systems Research* **60**,167–177 (2002).
7. K. Y. Lee, A. Sode-Yome, and J. H. Park, "Adaptive Hopfield neural networks for economic load dispatch," *IEEE Trans. Power Systems* **13**, 519- 526 (1998).
8. S. Baskar, P. Subbaraj, and M.V.C. Rao, "Hybrid real coded genetic algorithm solution to economic dispatch problem", *Computers and Electrical Engineering* **29**, 407-419 (2003).
9. T. Jayabarathi, K. Jayaprakash, D. N. Jeyakumar, and T. Raghunathan, "Evolutionary programming techniques for different kinds of economic dispatch problems", *Electric Power Systems Research* **73**, 169-176 (2005).
10. Y. M. Park, J. R. Wong, and J. B. Park, "A new approach to economic load dispatch based on improved evolutionary programming", *Eng. Intell. Syst. Elect. Eng Commun.* **6**, 103-110 (1998).
11. W.-M. Lin, F.-S. Cheng, and M.-T. Tsay, "Nonconvex economic dispatch by integrated artificial intelligence", *IEEE Trans. Power Systems* **16**, 307-311 (2001).
12. J.-B. Park, K.-S. Lee, and K. W. Lee, "A particle swarm optimization for economic dispatch with nonsmooth cost function", *IEEE Trans. Power Systems* **12**, 34-42 (2005).
13. C.L. Chiang and C.T. Su, "Adaptive-improved genetic algorithm for the economic dispatch of units with multiple fuel options", *Cybernetics and Systems: An International Journal* **36**, 687-704 (2005).
14. Marco A. Montes de Oca, Thomas Stutzle, Mauro Birattari, and Marco Dorigo, "A comparison of particle swarm optimization algorithms based on run-length distributions", *Proc. the 5th international conference on Ant Colony Optimization and Swarm Intelligence,* ANTS'06, 2006, pp. 1-12.
15. Asanga Ratnaweera, Saman K. Halgamuge and Harry C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients", IEEE Trans. Evolutionary Computation, vol. 8, no. 3, Jun. 2004, pp. 240 - 255.
16. Chao-Lung Chiang, "Improved genetic algorithm for power economic dispatch of units with valve point effects and multiple fuels", *IEEE Trans. Power Systems* **20**, 1690 – 1699 (2005).