Economic Dispatch with Multiple Fuels by Self-Organizing Hierarchical Particle Swarm Optimization

Le Dinh Luong, Le Anh Dung, and Vo Ngoc Dieu

Abstract— *This paper proposes a self-organizing hierarchical particle swarm optimization (SOH-PSO) algorithm for solving economic dispatch with multiple fuels (EDMF) problem. 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— **About four key words or phrases in alphabetical order, separated by commas.**

1. INTRODUCTION

The operation cost in power systems needs to be minimized at each time satisfying constraints via economic dispatch (ED) problem. 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 to linearization the segments and solving them by traditional methods [1]. 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 [2] 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 as Genetic algorithm (GA), Evolutionary programming (EP), Nonlinear programming (NLP), Quadratic programming (QP), Tabu search (TS), Simulated Annealing (SA), Interior Point Methods (IP), Mixed Integer Programming (MIP)… 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… and has produced good results than other methods.

The purpose of this paper is to apply the advanced PSO algorithm to solve the problem of multi-fuel economic dispatch. Advanced PSO method is tested and confirmed by comparing results with other methods such as the numerical method (HNUM) [3], the Hopfield Neural Network (HNN) [4], Adaptive Hopfield Neural Network (AHNN) [6], Enhanced Lagrangian Artificial Neural Network (ELANN) [5], Improved Evolutionary Programming (IEP) [9], Modified Particle Swarm Optimization (MPSO) [11], Real Coded Genetic Algorithm (RCGA) [7], Hybrid Real Coded Genetic Algorithm (HRCGA) [7], Evolutionary Programming, Tabu search and Quadratic programming (ETQ) [10], Conventional Evolutionary Programming (CEP) [8].

2. PROBLEM FORMULATION

The main objective of 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 with an objective function with equality and inequality constraints.

Consider a system with N plants and each plant

Le Dinh Luong is with the Faculty of Mechanical - Electrical - Electronic, Ho Chi Minh City University of Technology, Viet Nam, E-mail: ledinhluong@gmail.com

Le Anh Dung is with Binh Duong Power Company, Vietnam.

Vo Ngoc Dieu is with Ho Chi Minh City University of Technology, Ho Chi Minh City, Viet Nam. He is now with the Department of Power Systems, E-mail: vndieu@gmail.com

Truong Phung Hiep Minh Phuong is with the Relay Division, Southern Electrical Testing Company (ETC2) Ho Chi Minh City, Viet Nam, E-mail: tphmphuong@yahoo.com

Nguyen Huu Thien An is Master student at Ho Chi Minh City University of Technology.

generates a P_i MW of capacity. The capacity of plant should be scheduled so that the total cost F is minimized.

Mathematically, the problem is formulated as follows.

$$\text{Min } F = \sum_{i=1}^{N} F_i(P_i) \qquad (1)$$

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

$$F_i(P_i) = \begin{cases} a_{i1} + b_{i1}P_i + c_{i1}P_i^2, \text{fuel } 1, P_{i,\min} \le P_i \le P_{i1} \\ a_{i2} + b_{i2}P_i + c_{i2}P_i^2, \text{fuel } 2, P_{i1} < P_i \le P_{i2} \\ \ldots \\ a_{ik} + b_{ik}P_i + c_{ik}P_i^2, \text{fuel } k, P_{ik-1} < P_i \le P_{i,\max} \end{cases} \qquad (2)$$

subject to

(1) Power balance constraint

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

where the power loss is approximately calculated by Kron's formula (4):

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

(2) Generator operating limits

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

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.

3. SOH-PSO FOR EDMF

3.1 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, position and velocity of individual i is described by vectors $X_i = (x_{i1},\ldots, x_{id})$ and $V_i = (v_{i1},\ldots, v_{id})$. $Pbest_i = (x_{i1}^{Pbest},......,x_{id}^{Pbest})$ is the best location for the current instance i and $Gbest_i = (x_{i1}^{Gbest},......,x_{id}^{Gbest})$ is the best location for 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:

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

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

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

where

X_i^k	position of individual i at loop k
X_i^{k+1}	position of individual i in the loop k+1
V_i^k	velocity of individual i in the loop k
V_i^{k+1}	velocity of individual i in the loop k+1
ω	constant weight inertia
c_1	individual experience coefficient
c_2	coefficient of social relations of individual
$rand_1$, $rand_2$	random numbers between [0, 1]
$Pbest_i^k$	best position of individual i in the loop k
$Gbest^k$	best position of swarm in the loop 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 loop k, we need to determine the position of individual i at the next loop 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. Firstly, 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; Secondly, $c_1 rand_1 \times \left(Pbest_i^k - X_i^k\right)$ shows the individual experience based on the previous search, toward the best position $Pbest_i^k$; and lastly $c_2 rand_2 \times \left(Gbest^k - X_i^k\right)$ 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 loop k+1. After obtaining the velocity vector V_i^{k+1}, it is combined with the position vector X_i^k at loop k to obtain the position vector X_i^{k+1} of individual i in the loop k +1.

3.2 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_{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 [1] 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 is 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.

3.3 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 is 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:

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

$$c_1 = (c_{1f} - c_{1i}) . \frac{Iter}{Iter_{\max}} + c_{1i} \quad (16)$$

$$c_2 = (c_{2f} - c_{2i}) . \frac{Iter}{Iter_{\max}} + c_{2i} \quad (17)$$

where c_{1f}, c_{1i}, c_{2f}, c_{2i} are constant inertias and

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

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

3.4 PSO with Self Organizing Hierarchical

As we know, most of the improved PSO algorithm based on the linear change of the inertia coefficient and the penalty coefficient method. However, in some complex function, controlling the diversity of the population coefficient changes linearly inertia will cause the individual may soon converge to the optimal solution locally. Therefore, 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

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

$$c_1 = (c_{1f} - c_{1i}).\frac{Iter}{Iter_{\max}} + c_{1i} \quad (20)$$

$$c_2 = (c_{2f} - c_{2i}).\frac{Iter}{Iter_{\max}} + c_{2i} \quad (21)$$

where c_{1f}, c_{1i}, c_{2f}, c_{2i} are constant inertias and

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

3.5 Implementation of SOH-PSO to EDMF

The objective function:

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

Neglecting transmission losses, power balance constraint

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

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 [2].

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 $i = 1, \ldots, d$ is the number of individuals in the swarm.

$$x_i = [\, P_{i1}, P_{i2}, \ldots, P_{i,n-1}, P_n \,]$$

$$v_i = [\, v_{i1}, v_{i2}, \ldots, v_{i,n-1}]$$

where P_{ik} is a generation capacity of the plant, with $k = 1, \ldots, n_i$ (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 $i = 1, \ldots, d$. Calculated function fitness.

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

Step 6: Setting loop $t = 1$

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

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 limited capacity 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 Pbesti, Gbest if objective function value is better than old objective function value

Step 11: Increase number of loop $t = t+1$

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

4. NUMERICAL RESULTS

To validate the effectiveness of proposed advanced PSO method, two improved PSO algorithm (PSO TVIW, TVAC PSO) was tested on 10 plant systems with different load demands and each plant has cost function including many quadratic to many different types of fuel. The results from these two methods were compared with HNUM [3], HNN [4], AHNN [6], ELANN [5], IEP [9], CEP [8], FEP [8], IFEP [8], MPSO [11], RCGA [7], HRCGA [7], and ETQ [10] for the load changes from 2400MW to 2700MW.

- *Setting parameter for algorithm PSO-TVIW*

Number of loop: $Iter_{max}$=100

Number of individuals in the swarm: d=20

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 loop: $Iter_{max}$=100

Number of individuals in the swarm: d=20

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 loop: $Iter_{max}$=100

Number of individuals in the swarm: d=20

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.

- The data system has 10 plant in following table

Table 1. The data of 10 plant system

Point	Branch	Fuel	a_i	b_i	c_i	P_{min}	P_{max}
1	1	1	26.97	-0.3975	0.2176	100	196
1	2	2	21.13	-0.3059	0.1861	196	250
2	1	2	1.865	-0.0399	0.1138	50	114
2	2	3	13.65	-0.1980	0.1620	114	157
2	3	1	118.4	-1.269	0.4194	157	230
3	1	1	39.79	-0.3116	0.1457	200	332
3	2	3	-2.876	0.0339	0.8035	332	388
3	3	2	-59.14	0.4864	0.1176	388	500
4	1	1	1.983	-0.0311	0.1049	99	138
4	2	2	52.85	-0.6348	0.2758	138	200
4	3	3	266.8	-2.338	0.5935	200	265
5	1	1	13.92	-0.0873	0.1066	190	338
5	2	2	99.76	-0.5206	0.1597	338	407
5	3	3	-53.99	0.4462	0.1498	407	490
6	1	2	1.983	-0.0311	0.1049	85	138
6	2	1	52.85	-0.6348	0.2758	138	200
6	3	3	266.8	-2.338	0.5935	200	265
7	1	1	18.93	-0.1325	0.1107	200	331
7	2	2	43.77	-0.2267	0.1165	331	391
7	3	3	-43.35	0.3559	0.2454	391	500
8	1	1	1.983	-0.0311	0.1049	99	138
8	2	2	52.85	-0.6348	0.2758	138	200
8	3	3	266.8	-2.338	0.5935	200	265
9	1	3	14.23	-0.0182	0.6121	130	213
9	2	1	88.53	-0.5675	0.1554	213	370
9	3	3	14.23	-0.0182	0.6121	370	440
10	1	1	13.97	-0.0994	0.1102	200	362
10	2	3	46.71	-0.2024	0.1137	362	407
10	3	2	-61.13	0.5084	0.4164	407	490

Case study 1

In this example, load demand is 2400MW.

Table 2. Costs and Processing Time

Method	Minimum Cost ($)	Maximum Cost ($)	Average Cost ($)	Time (s)
PSO TVIW	481.7226	485.1255	481.7566	0.33
PSO TVAC	481.7226	487.9564	481.7922	0.34
SOH- PSO	481.7226	484.1453	481.7468	0.31

Table 3. Costs and Capacity of Plant

Method	Plant	2400 MW			
		P_i(MW)	Fuel	Branch	Cost
PSO TVIW	1	189.746	1	1	481.7226
	2	202.341	1	3	
	3	253.901	1	1	
	4	233.044	3	3	
	5	241.855	1	1	
	6	233.046	3	3	
	7	253.269	1	1	
	8	233.041	3	3	
	9	320.375	1	2	
	10	239.381	1	1	
PSO TVAC	1	189.863	1	1	481.7226
	2	202.347	1	3	
	3	254.109	1	1	
	4	232.896	3	3	
	5	241.738	1	1	
	6	233.061	3	3	
	7	253.298	1	1	
	8	233.044	3	3	
	9	320.226	1	2	
	10	239.418	1	1	
SOH- PSO	1	189.608	1	1	481.7226
	2	202.272	1	3	
	3	253.987	1	1	
	4	233.013	3	3	
	5	241.892	1	1	
	6	233.139	3	3	
	7	253.252	1	1	
	8	233.065	3	3	
	9	320.178	1	2	
	10	239.595	1	1	

Table 4. Comparison of Total Cost and Processing time in Case 1

Method	P (MW)	Cost/hour($/h)	Time (s)
HNUM [3]	2401.2	488.50	1.08
HNN [4]	2399.8	487.87	60
AHNN [6]	2400	481.72	4
ELANN [5]	2400	481.74	11.53
IEP [9]	2400	481.779	-
MPSO [11]	2400	481.723	-
RCGA [7]	2400	481.723	49.92
HRCGA [7]	2400	481.722	6.1
ETQ [10]	2400	481.72	86.3
PSO TVIW	2400	481.7226	0.33
PSO TVAC	2400	481.7226	0.34
SOH- PSO	2400	481.7226	0.31

Case study 2

In this example, load demand is 2500MW.

Table 5. Costs and Processing Time

Method	Minimum Cost ($)	Maximum Cost ($)	Average Cost ($)	Time (s)
PSO TVIW	526.2389	526.2427	526.24003	0.38
PSO TVAC	526.239	526.2545	526.2418	0.37
SOH- PSO	526.2388	526.242	526.23938	0.37

Table 6. Costs and Capacity of Plant

Method	Plant	2700 MW			
		P_i(MW)	Fuel	Branch	Cost
PSO TVIW	1	206.582	1	1	526.2389
	2	206.351	1	3	
	3	265.921	1	1	
	4	236.014	3	3	
	5	257.858	1	1	
	6	236.01	3	3	
	7	268.799	1	1	
	8	235.913	3	3	
	9	331.736	1	2	
	10	254.815	1	1	
PSO TVAC	1	206.415	1	1	526.239
	2	206.479	1	3	
	3	265.892	1	1	
	4	236.038	3	3	
	5	257.763	1	1	
	6	236.085	3	3	
	7	268.645	1	1	
	8	235.993	3	3	
	9	331.562	1	2	
	10	255.127	1	1	
SOH- PSO	1	206.627	2	2	526.2388
	2	206.432	1	3	
	3	265.803	1	1	
	4	236.056	3	3	
	5	258.02	1	1	
	6	235.96	3	3	
	7	268.769	1	1	
	8	235.982	3	3	
	9	331.435	1	2	
	10	254.917	1	1	

Table 7. Comparison of Total Cost and Processing time in Case 2

Method	P (MW)	Cost/hour($/h)	Time (s)
HNUM [3]	2500.1	526.7	-
HNN [4]	2499.8	526.13	60
AHNN [6]	2500	526.230	4
ELANN [5]	2500	526.27	12.25
IEP [9]	2500	526.304	-
MPSO [11]	2500	526.239	-
RCGA [7]	2500	526.239	49.92
HRCGA [7]	2500	526.238	6.1
CEP [8]	2500	526.246	0.495
FEP [8]	2500	526.262	0.394
IFEP [8]	2500	526.246	0.558
PSO TVIW	2500	526.2389	0.38
PSO TVAC	2500	526.239	0.37
SOH- PSO	2500	526.2388	0.37

Case study 3

In this example, load demand is 2600MW.

Table 8. Costs and Processing Time

Method	Minimum Cost ($)	Maximum Cost ($)	Average Cost ($)	Time (s)
PSO TVIW	574.3809	574.7441	574.5649	0.38
PSO TVAC	574.381	574.7453	574.5622	0.37
SOH- PSO	574.3808	574.7432	574.41714	0.37

Table 9. Costs and Capacity of Plant

Method	Plant	2600 MW			
		P_i(MW)	Fuel	Branch	Cost
PSO TVIW	1	216.597	1	1	574.3809
	2	210.962	1	3	
	3	278.38	1	1	
	4	239.075	3	3	
	5	275.285	1	1	
	6	239.295	3	3	
	7	286.194	1	1	
	8	239.053	3	3	
	9	343.294	1	2	
	10	271.865	1	1	
PSO TVAC	1	216.65	1	1	574.381
	2	210.976	1	3	
	3	278.52	1	1	
	4	239.116	3	3	
	5	275.56	1	1	
	6	239.019	3	3	
	7	285.821	1	1	
	8	239.225	3	3	
	9	343.322	1	2	
	10	271.79	1	1	
SOH- PSO	1	216.544	2	2	574.3808
	2	210.886	1	3	
	3	278.342	1	1	
	4	239.102	3	3	
	5	275.598	1	1	
	6	239.162	3	3	
	7	285.677	1	1	
	8	239.176	3	3	
	9	343.497	1	2	
	10	272.016	1	1	

Table 10. Comparison of Total Cost and Processing time in Case 3

Method	P (MW)	Cost/hour($/h)	Time (s)
HNUM [3]	2599.3	574.03	-
HNN [4]	2599.8	574.26	60
AHNN [6]	2600	574.37	4
ELANN [5]	2600	574.41	9.99
IEP [9]	2600	574.473	-
MPSO [11]	2600	574.381	-
RCGA [7]	2600	574.396	37.57
HRCGA [7]	2600	574.38	5.4
PSO TVIW	2600	574.3809	0.38
PSO TVAC	2600	574.381	0.37
SOH- PSO	2600	574.3808	0.37

Case study 4

In this example, load demand is 2700MW.

Table 11. Costs and Processing Time

Method	Minimum Cost ($)	Maximum Cost ($)	Average Cost ($)	Time (s)
PSO TVIW	623.8092	623.8997	623.82531	0.38
PSO TVAC	623.8092	623.8353	623.8133	0.37
SOH- PSO	623.8092	623.8353	623.81199	0.37

Table 12. Costs and Capacity of Plant

Method	Plant	2700 MW			
		P_i(MW)	Fuel	Branch	Cost
PSO TVIW	1	218.23	1	1	623.8092
	2	211.707	1	3	
	3	280.84	1	1	
	4	239.669	3	3	
	5	278.513	1	1	
	6	239.679	3	3	
	7	288.456	1	1	
	8	239.58	3	3	
	9	428.554	1	2	
	10	274.772	1	1	
PSO TVAC	1	218.266	1	1	623.8092
	2	211.609	1	3	
	3	280.864	1	1	
	4	239.648	3	3	
	5	278.159	1	1	
	6	239.614	3	3	
	7	288.786	1	1	
	8	239.633	3	3	
	9	428.411	1	2	
	10	275.012	1	1	
SOH-PSO	1	218.393	2	2	623.8092
	2	211.733	1	3	
	3	280.698	1	1	
	4	239.683	3	3	
	5	278.474	1	1	
	6	239.451	3	3	
	7	288.529	1	1	
	8	239.405	3	3	
	9	428.596	3	3	
	10	275.036	1	1	

Table 13. Comparison of Total Cost and Processing time in Case 4

Method	P (MW)	Cost/hour($/h)	Time (s)
HNUM [3]	2702.2	625.18	-
HNN [4]	2699.7	626.12	60
AHNN [6]	2700	626.24	4
ELANN [5]	2700	623.88	21.36
IEP [9]	2700	623.851	-
MPSO [11]	2700	623.809	-
RCGA [7]	2700	623.809	44.56
HRCGA [7]	2700	623.809	6.47
PSO TVIW	2700	623.8092	0.38
PSO TVAC	2700	623.8092	0.37
SOH- PSO	2700	623.8092	0.37

The compared results from Table 2 to Table 13 show that the SOH-PSO 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 SOH-PSO easily prevent the violation of all the active constraints.

Tables 2, 5, 8 and 11 show the minimum, mean, maximum cost achieved by the SOH-PSO algorithm in 100 runs. Obviously, the minimum costs acquired by the proposed methods are all lower than that obtained HNUM [3], HNN [4], AHNN [6], ELANN [5], IEP [9], CEP [8], FEP [8], IFEP [8], MPSO [11], RCGA [7], HRCGA [7], and ETQ [10] in all cases. Total costs of the proposed method are close to those from the others for the rest cases. Note the power balance constraint in HNUM [3] and HNN [4] are not satisfied. These results show that the proposed methods are feasible and indeed capable of acquiring better solution. The optimal dispatches of the generators are listed in Tables 3, 6, 9 and 12. Also note that all outputs of generator are within its permissible limits.

5. CONCLUSION

In this paper, the self-organizing hierarchical particle swarm optimization (SOH-PSO) algorithm has been presented to solve the economic dispatch with multiple fuels (EDMF) problem. 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.

Numerical results demonstrate that the SOH-PSO algorithm has more advantages for solving the EDMF than the previous methods in all the test case in terms of total costs and computational times. The results show that the proposed algorithm is efficient for solving nonconvex fuel cost functions. This paper is the first step in the study of EDMF function. A further direction for this study will be to apply other large-scale power systems with valve point effects.

6. REFERENCES

[1] Eberhart, R. C. and Shi, Y. (1998)(b). Comparison between genetic algorithms and particle swarm optimization. In V. W. Porto, N. Saravanan, D. Waagen, and A. E. Eiben, Eds. *Evolutionary Programming VII: Proc. 7th Ann. Conf. on Evolutionary Programming Conf.*, San Diego, CA. Berlin: Springer-Verlag.

[2] Asanga Ratnaweera, Saman K. Halgamuge, Member, IEEE, and Harry C. Watson. Self-Organizing Hierarchical Particle Swarm Optimizer With Time-Varying Acceleration Coefficients. *IEEE Transactions on Evolutionary Computation*, Vol. 8, No. 3, June 2004, pp. 240-255.

[3] Lin, C. E. and Viviani, G. L. 1984. Hierarchical economic dispatch for piecewise quadratic cost functions. *IEEE Transactions on Power Apparatus and Systems*, PAS-103(6): 1170 -1175.

[4] Park, J. H.; Kim, Y. S.; Eom, I. K.; and Lee, K. Y. 1993. Economic load dispatch for piecewise quadratic cost function using Hopfield neural network. *IEEE Transactions on Power Systems*, 8(3): 1030-1038.

[5] Lee, S. C. and Kim, Y. H. 2002. An enhanced Lagrangian neural network for the ELD problems

with piecewise quadratic cost functions and nonlinear constraints. *Electric Power Systems Research 60*: 167–177.

[6] Lee, K. Y.; Sode-Yome, A.; and Park, J. H. 1998. Adaptive Hopfield neural networks for economic load dispatch. *IEEE Transactions on Power Systems* 13(2): 519- 526.

[7] Baskar, S.; Subbaraj P.; and Rao, M.V.C. 2003. Hybrid real coded genetic algorithm solution to economic dispatch problem. *Computers and Electrical Engineering* 29: 407-419.

[8] Jayabarathi, T.; Jayaprakash, K.; Jeyakumar, D. N.; and Raghunathan,T. 2005. Evolutionary programming techniques for different kinds of economic dispatch problems. *Electric Power Systems Research* 73: 169-176.

[9] Park, Y. M.; Wong, J. R.; and Park, J. B. 1998. A new approach to economic load dispatch based on improved evolutionary programming. *Eng. Intell. Syst. Elect. Eng Commun*. 6(2): 103-110.

[10] Lin, W.-M.; Cheng, F.-S.; and Tsay, M.-T. 2001. Nonconvex economic dispatch by integrated artificial intelligence. *IEEE Trans. Power Systems* 16(2): 307-311.

[11] Park, J.-B.; Lee, K.-S.; and Lee, K. W. 2005. A particle swarm optimization for economic dispatch with nonsmooth cost function. *IEEE Transactions on Power Systems*, 12(1): 34-42.

[12] 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(7) (2005), pp. 687-704.

[13] Marco A. Montes de Oca, Thomas Stutzle, Mauro Birattari, and Marco Dorigo. A comparision of particle swarm optimization algorithms based on run-length distributions. *IRIDIA, Code*, Universite Libre de Bruxelles, Brussels, Belgium.

[14] Asanga Ratnaweera, Saman K. Halgamuge and Harry C. Watson. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Transactions On Evolutionary Computation*, Vol.8, No.3, June 2004.

[15] Chao-Lung Chiang. Improved genetic algorithm for power economic dispatch of units with valve point effects and multiple fuels. *IEEE Transactions on Power Systems*, Vol.20, No.4, November 2005

[16] K.T.Chaturvedi, Manjaree Pandit and Laxmi Srivastava. Self-organizing Hierarchical particle swarm optimization for nonconvex economic dispatch. *IEEE Transactions on Power Systems*, Vol.23, No.3, August 2008

[17] J.-P. Chiou. Variable scaling hybrid differential evolution for large-scale economic dispatch problems. *Elect. Power Syst*. Res. 77 (3/4) (2007) 212–218.