Applied Energy 87 (2010) 327–339

Contents lists available at ScienceDirect

Applied Energy

journal homepage: www.elsevier.com/locate/apenergy

A new fuzzy adaptive hybrid particle swarm optimization algorithm for non-linear, non-smooth and non-convex economic dispatch problem

Taher Niknam *

Electronic and Electrical Department, Shiraz University of Technology, Modars Blvd. Shiraz, P.O. 71555-313, Iran

ARTICLE INFO

Article history:
Received 22 November 2008
Received in revised form 27 April 2009
Accepted 11 May 2009
Available online 13 June 2009

Keywords:
Economic dispatch
Fuzzy adaptive particle swarm optimization
Evolutionary algorithm
Nelder–Mead

ABSTRACT

Economic dispatch (ED) plays an important role in power system operation. ED problem is a non-smooth and non-convex problem when valve-point effects of generation units are taken into account. This paper presents an efficient hybrid evolutionary approach for solving the ED problem considering the valve-point effect. The proposed algorithm combines a fuzzy adaptive particle swarm optimization (FAPSO) algorithm with Nelder–Mead (NM) simplex search called FAPSO-NM. In the resulting hybrid algorithm, the NM algorithm is used as a local search algorithm around the global solution found by FAPSO at each iteration. Therefore, the proposed approach improves the performance of the FAPSO algorithm significantly. The algorithm is tested on two typical systems consisting of 13 and 40 thermal units whose incremental fuel cost functions take into account the valve-point loading effects.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

The ED problem is an important optimization problem in power system operation. The problem is used to determine the optimal power outputs of all generating units by minimizing the total fuel cost while the total generation should be equal to the total system demand plus the transmission network loss, the generation output of each unit should be between its minimum and maximum limits [1]. In the previous studies, various mathematical programming methods and optimization techniques have been utilized to solve the ED problem, including the lambda-iteration method, the base point and participation factors method, the gradient method, and Newton method [2–6]. These methods are based on the assumption that the incremental costs of the generators are monotonically increasing. Unfortunately, in practical situations this assumption may lead to infeasibility because of the non-linear characteristics of real generators. The non-linear characteristics of generators consist of prohibited zones, ramp rate limits, and non-smooth or non-convex cost functions. Dynamic programming, non-linear programming, and mix integer programming have been proposed in the literature to address this issue [2–10]. However, these methods suffer from the curse of dimensionality especially in dealing with modern power systems with large number of generators. Moreover, some assumptions may be needed in order to decrease the search space and avoid getting stuck in a local optimum. Recently, artificial intelligent-based techniques, including the genetic algorithms (GA) [11,13], the simulated annealing (SA) [14], evolutionary programming (EP) [15], tabu search (TS) [16], particle swarm optimization (PSO) [17–19], hybrid PSO and sequential quadratic programming (PSO-SQP) [20], differential evolution (DE) [3], hybrid DE and sequential quadratic programming (DE-SQP) [21], hybrid EP-SQP [3], variable scaling hybrid differential evolution (VSHDE) [4], hybrid GA (HGA) [2], evolutionary strategy optimization (ESO) [22–23], self organizing hierarchical PSO (SOH-PSO) [24] and new PSO(NPSO) [25] were applied to this problem.

The PSO algorithm is one of the modern evolutionary algorithms. This algorithm was first proposed by Kennedy and Eberhart. PSO was developed through simulation of a simplified social system, and has been found to be robust in solving continuous non-linear optimization problems [27–29]. The PSO algorithm can produce high-quality solutions within shorter calculation time and more stable convergence characteristics than other stochastic methods [27–29]. Recently, PSO has been successfully used to solve the ED problem while considering generator constraints and non-smooth cost constraints [17–19]. However, the performance of the traditional PSO significantly depends on its parameters, and it often suffers from the problem of being trapped in local optima. Also the final outputs have some stochastic characteristics. In order to avoid these problems, this paper presents a new hybrid evolutionary optimization algorithm based on combining the fuzzy adaptive particle swarm optimization (FAPSO) and Nelder–Mead (NM) algorithms, called FAPSO-NM. The proposed algorithm is employed to solve the ED problem with considering the valve-point effect. In the algorithm, the inertia weight and learning factors of PSO are dynamically adjusted using fuzzy IF/THEN rules. The proposed hybrid algorithm uses the FAPSO algorithm as the main optimizer and the NM algorithm as a local search technique around

* Tel.: +98 711 7264121; fax: +98 711 7353502.
E-mail addresses: niknam@sutech.ac.ir, taher_nik@yahoo.com

0306-2619/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.apenergy.2009.05.016

the best solution found by FAPSO at each iteration to reach the global minimum solutions. To evaluate the proposed hybrid algorithm, it is tested on two case studies with non-convex solution spaces. The results of the proposed FAPSO-NM are compared with those of the previous approaches, which show the effectiveness of the proposed algorithm in terms of solution quality. The proposed algorithm can be utilized in all non-linear, non-differentiable and discrete optimization problems.

The rest of the paper is as follows. In Section 2, the ED problem formulation is presented. Sections 3 and 4 briefly describe the basics of the FAPSO and NM algorithms, respectively. The proposed hybrid algorithm and its implementation to solve the ED problem are presented in Sections 5 and 6, respectively. In Section 7, the effectiveness of the proposed approach is demonstrated by comparing its results with those of the other algorithms. The paper concludes in References.

2. ED problem formulation

2.1. Formulation of objective function

The unit commitment (UC) problem is one of the critical problems in the economic operation of power systems. The UC problem determines the unit generation schedule by minimizing the operating cost and satisfying constraints such as load balance, system spinning reserve, ramp rate limits, fuel constraints, multiple emission requirements, and minimum up and down time limits for the predefined period [26]. The ED problem is a sub-problem of the UC problem. The ED problem must carry out the optimal generation dispatch among the operating units to meet the system load demand and operation constraints of generators while the scheduled combination units at each specific period of operation are determined. Generally, the objective function of the ED problem is non-differentiable at some points due to the valve-point effects. Therefore, the objective function should include a set of non-smooth cost functions. The valve-point effects include ripples in the heat rate curves, which increases the number of local optima. This paper considers non-smooth cost functions of generation units with valve-point effects. The objective function is usually defined as the superposition of sinusoidal functions and quadratic functions. Fig. 1 shows the cost function curve of a thermal generator.

The bold line shows the approximation of the cost function curve by a quadratic function:

$$F_i(P_{gi}) = a_i P_{gi}^2 + b_i P_{gi} + c_i \tag{1}$$

Since the thin line in Fig. 1 gives a more realistic approximation for the cost function of generators, it will be used instead of the quadratic function. The ripples in the thin-line input–output curve indicate the effects of the valves. As shown in Fig. 1, the curve contains higher order non-linearity and discontinuity compared with the smooth cost function due to the valve-point effects. In order to obtain a more accurate model, which takes into account of the valve-point effects, the cost function is modified to include the ripple curve. This can be done by adding sinusoidal functions to the quadratic function. Then the modified cost function of a generator will be [2–6]:

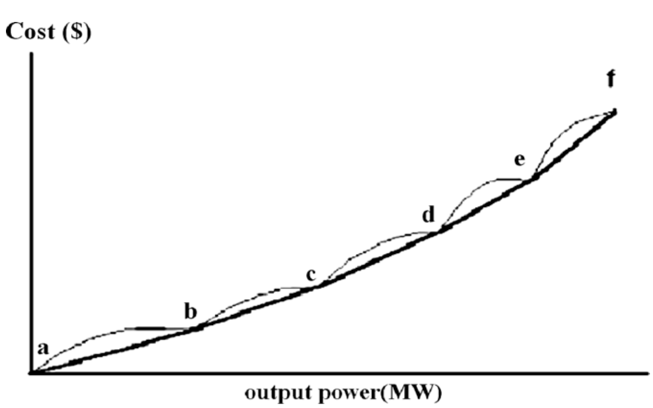

Fig. 1. Cost function of a generator with valve-points.

$$\begin{aligned} \min\ J(X) &= \sum_{i=1}^{Ng} (F_i(P_{gi}) + |e_i \sin(f_i(P_{gi,\min} - P_{gi}))|) \\ F_i(P_{gi}) &= a_i P_{gi}^2 + b_i P_{gi} + c_i \\ X &= [P_{g1}, P_{g2}, \ldots, P_{g,Ng}] \end{aligned} \tag{2}$$

where a_i, b_i, and c_i are the cost coefficients of the ith generator. e_i and f_i are two coefficients required for introducing valve-point discontinuities. $F_i(P_{gi})$ is the total cost generation of the ith generator. P_{gi} is the output of the ith generator. Ng is the number of generators. $P_{gi,\min}$ is the minimum generation limit (MW) of the ith generator. X is the control variable vector.

2.2. Constraints

The ED problem is subject to the following constraints.

- Power balance constraint:
  The power balance constraint is based on the principle of equilibrium between the total system generation and total system loads (P_{load}) and losses (P_{loss}).

$$\sum_{i=1}^{Ng} P_{gi} = P_{load} + P_{loss} \tag{3}$$

  P_{loss} is calculated using B-coefficients and it is described by

$$P_{loss} = \sum_{i=1}^{Ng} \sum_{j=1}^{Ng} P_{gi} \times B_{ij} \times P_{gj} + \sum_{i=1}^{Ng} B_{i0} \times P_{gi} + B_{00} \tag{4}$$

  where B_{ij} is the i, jth element of the loss coefficient square matrix. B_{i0} is the ith element of the loss coefficient vector. B_{00} is the loss coefficient constant.
- Output generator constraints:
  The output power of each generating unit has a lower and upper limit. The output generator constraint is defined by a pair of inequality constraints as follows:

$$P_{gi,\min} \leqslant P_{gi} \leqslant P_{gi,\max} \tag{5}$$

  where $P_{gi,\max}$ is the maximum power output of the ith generating unit.

3. Fuzzy adaptive PSO

3.1. Original PSO

PSO is a stochastic optimization algorithm [27–29]. The main idea of the PSO is the mathematical modeling and simulation of the food searching activities of a flock of birds. In the multidimensional space, each particle is moved toward the optimal point by changing its position according to a velocity. The velocity of a particle is calculated by three components; inertia, cognitive, and social. The inertial component simulates the inertial performance of the bird to fly in the previous direction. The cognitive component models the memory of the bird about its previous best position. The social component models the memory of the bird about the best position among the particles. The particles move around the multidimensional search space until they find the optimal solution. Based on the above discussion, the mathematical model for PSO is as follows.

$$\begin{aligned} V_i^{(t+1)} &= \omega \cdot V_i^{(t)} + c_1 \cdot rand_1(\cdot) \cdot (Pbest_i - X_i^{(t)}) \\ &\quad + c_2 \cdot rand_2(\cdot) \cdot (Gbest - X_i^{(t)}) \\ X_i^{(t+1)} &= X_i^{(t)} + V_i^{(t+1)} \\ i &= 1, 2, 3, \ldots, N_{Swarm} \end{aligned} \tag{6}$$

where, i is the index of each particle, t is the current iteration number, $rand_1(\cdot)$ and $rand_2(\cdot)$ are random numbers between 0 and 1. $Pbest_i$ is the best previous experience of the ith particle that is recorded. *Gbest* is the best particle among the entire population. N_{Swarm} is the number of the swarms. Constants c_1 and c_2 are the weighting factors of the stochastic acceleration terms, which pull each particle towards the $Pbest_i$ and *Gbest*. ω is the inertia weight.

As indicated in (6), there are three tuning parameters; ω, c_1, and c_2 that each of them has a great impact on the algorithm performance. The inertia weight ω controls the exploration properties of the algorithm.

The learning factors c_1 and c_2 determine the impact of the personal best $Pbest_i$ and the global best *Gbest*, respectively. If $c_1 > c_2$, the particle has the tendency to converge to the best position found by itself $Pbest_i$ rather than the best position found by the population *Gbest*, and vice versa. Most implementations use a setting with $c_1 = c_2 = 2$ [27–31].

To implement the PSO algorithm to solve the ED problem, the following steps should be taken:

Step 1 The initial population and initial velocity for each particle should be generated randomly.
Step 2 The objective function is to be evaluated for each individual.
Step 3 The individual that has the minimum objective function should be selected as the global position.
Step 4 The ith individual is selected.
Step 5 The best local position (*Pbest*) is selected for the ith individual.
Step 6 The modified velocity for the ith individual needs to be calculated based on the local and global positions and Eq. (6).
Step 7 The modified position for the ith individual should be calculated based on Eq. (6) and then checked with its limit.
Step 8 If all individuals are selected, go to the next step, otherwise $i = i + 1$ and go to step 4.
Step 9 If the current iteration number reaches the predetermined maximum iteration number, the search procedure is stopped, otherwise go to step 2.

The last *Gbest* is the solution of the problem.

The Fig. 2 illustrates the flowchart of the original PSO to solve the ED problem.

It is probably impossible to define a unique set of parameters that work well in all cases. However, the following fuzzy adaptive PSO (FAPSO) algorithm has been useful to work in practice.

3.2. FAPSO

From experience, it is known that [29–31]:

(i) when the best fitness is found at the end of the run, low inertia weight and high learning factors are often preferred;
(ii) when the best fitness is stayed at one value for a long time, the number of generations for unchanged best fitness is large. The inertia weight should be increased and learning factors should be decreased.

According to this knowledge, a fuzzy system is utilized to tune the inertia weight and learning factors with the best fitness (*BF*) and the number of generations for the best unchanged fitness (*NU*) as the input variables, and the inertia weight (ω) and learning factors (c_1 and c_2) as the output variables.

The *BF* value determines the performance of the best candidate solution found so far. The optimization problems have different ranges of the *BF* values. To use a FAPSO, which is applicable to a various range of problems, the ranges of the *BF* and *NU* values are normalized into [0, 1.0]. The *BF* values can be normalized using the following formula:

$$NBF = \frac{BF - BF_{\min}}{BF_{\max} - BF_{\min}} \tag{7}$$

where, $BF_{\max}$ and $BF_{\min}$ are the maximum and minimum values of *BF* value.

NU values are normalized in a similar way. Other converting methods are possible as well. The bound values for ω, c_1, and c_2 are: $0.2 \leqslant \omega \leqslant 1.2$, $1 \leqslant c_1$ and $c_2 \leqslant 2$.

For fuzzification of every input and output, the membership functions shown in Fig. 3 are used.

In Fig. 1 PS (positive small), PM (positive medium), PB (positive big) and PR (positive bigger) are the linguistic values for the inputs and outputs.

The Mamdani-type fuzzy rule is used to formulate the conditional statements that comprise fuzzy logic. For example

R_i: IF (*NBF* is PB) and (*NU* is PM),
THEN (ω is PB), (c_1 is PM) and (c_2 is PM).

The fuzzy rules in Tables 1–3 [32] are used to adjust the inertia weight (ω) and learning factors (c_1 and c_2), respectively. Each rule represents a mapping from the input space to the output space.

To obtain a deterministic control action, a defuzzification strategy is required. In this paper, the centroid method has been used.

To apply the FAPSO algorithm to solve the ED problem, the following steps should be taken:

Step 1 The initial population and initial velocity for each particle should be generated randomly.
Step 2 The objective function is to be evaluated for each individual.
Step 3 The individual that has the minimum objective function should be selected as the global position.
Step 4 The ith individual is selected.
Step 5 The best local position (*Pbest*) is selected for the ith individual.
Step 6 Update the FAPSO parameters.
Step 7 Calculate the next position for each individual based on the FAPSO parameters and Eq. (6) and then checked with its limit.
Step 8 If all individuals are selected, go to the next step, otherwise $i = i + 1$ and go to step 4.
Step 9 If the current iteration number reaches the predetermined maximum iteration number, the search procedure is stopped, otherwise go to step 2.

The last *Gbest* is the solution of the problem.

The Fig. 4 illustrates the flowchart of the original PSO to solve the ED problem.

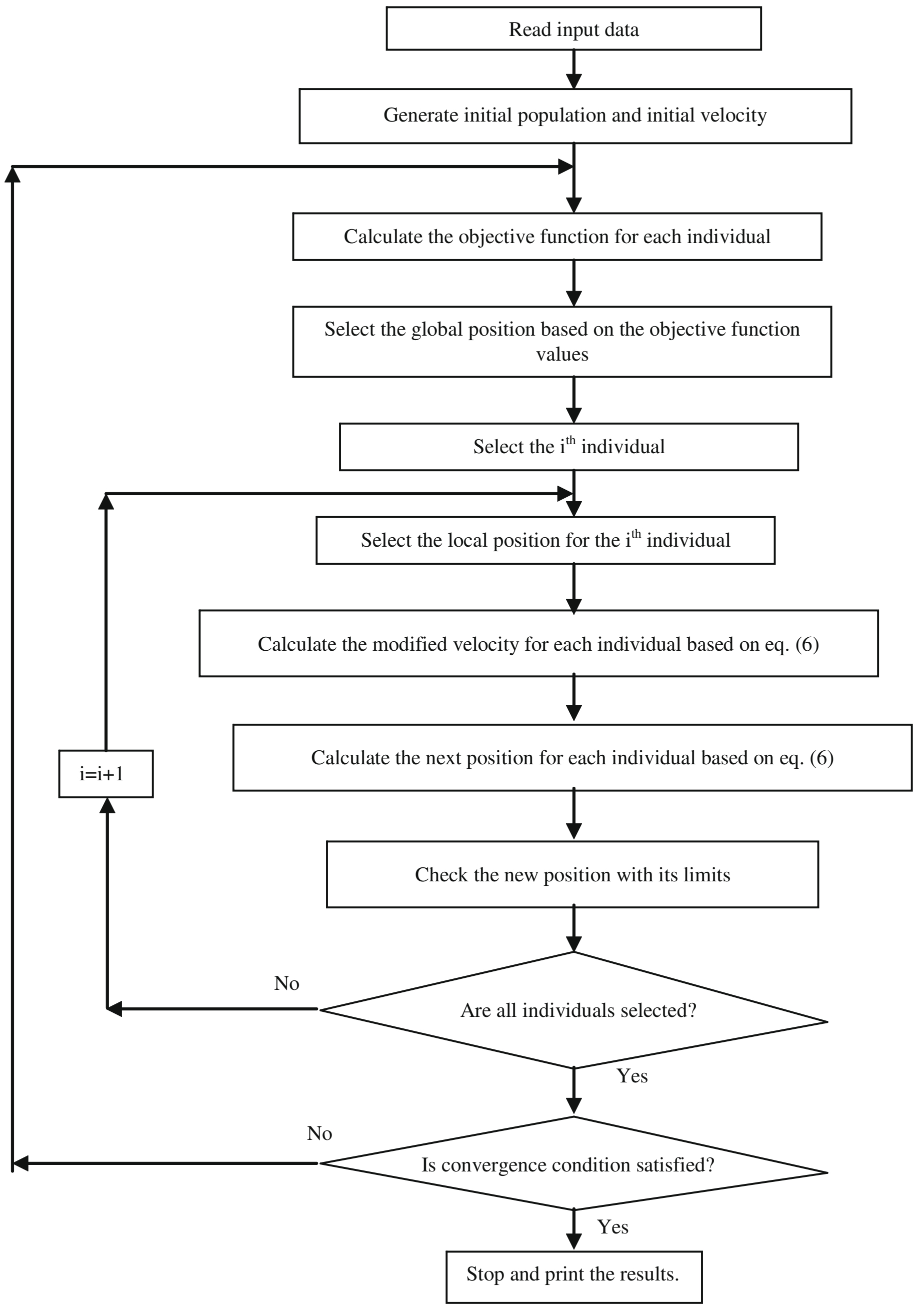

Fig. 2. Flowchart of the original PSO algorithm.

4. Nelder-Mead method

The Nelder-Mead method is a generally used non-linear optimization algorithm. It is a numerical method for minimizing an objective function in a multidimensional space [33–35].

The operations of the method are to rescale the simplex based on the local behavior of the function by using four basic procedures: reflection, expansion, contraction, and shrinkage [33–35]. Through these procedures, the simplex can successfully improve itself and get closer to the optimum solution. The original NM simplex procedure is outlined by the following steps:

Step 1 Initialization

Generate $N+1$ vertex points randomly to form an initial N-dimensional simplex. Evaluate the functional value at each vertex point of the simplex. $N+1$ vertex points have been sorted ascendingly based on the objective function values as below:

$$\begin{bmatrix} X_{low} & J_{low} \\ \cdot & \cdot \\ \cdot & \cdot \\ X_{highs} & J_{highs} \\ X_{high} & J_{high} \end{bmatrix}_{(N+1)\times(n+1)} \tag{8}$$

where X_{low}, X_{high}, and X_{highs} are the vertices with the lowest, the highest and, the second highest function values, respectively. J_{low}, J_{high}, and J_{highs} represent the corresponding

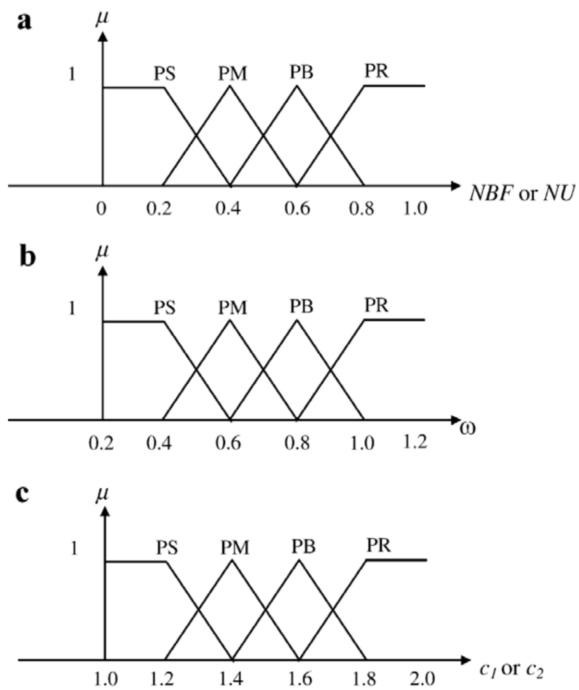

Fig. 3. Membership functions of inputs and outputs (a) *NBF* or *NU*, (b) ω, and (c) c_1 and c_2.

Table 1
Fuzzy rules for the inertia weight.

ω		*NU*			
		PS	PM	PB	PR
NBF	PS	PS	PM	PB	PB
	PM	PM	PM	PB	PR
	PB	PB	PB	PB	PR
	PR	PB	PB	PR	PR

Table 2
Fuzzy rules for learning factor c_1.

c_1		*NU*			
		PS	PM	PB	PR
NBF	PS	PR	PB	PB	PB
	PM	PB	PM	PM	PS
	PB	PB	PM	PS	PS
	PR	PM	PM	PS	PS

Table 3
Fuzzy rules for learning factor c_2.

c_2		*NU*			
		PS	PM	PB	PR
NBF	PS	PR	PB	PM	PM
	PM	PB	PM	PS	PS
	PB	PM	PM	PS	PS
	PR	PM	PS	PS	PS

observed function values, respectively. n is the number of state variables.

Step 2 Reflection

Find X_{cent}, the center of the simplex without X_{high} in the minimization case. Generate a new vertex X_{refl} by reflecting the worst point according to the following equation:

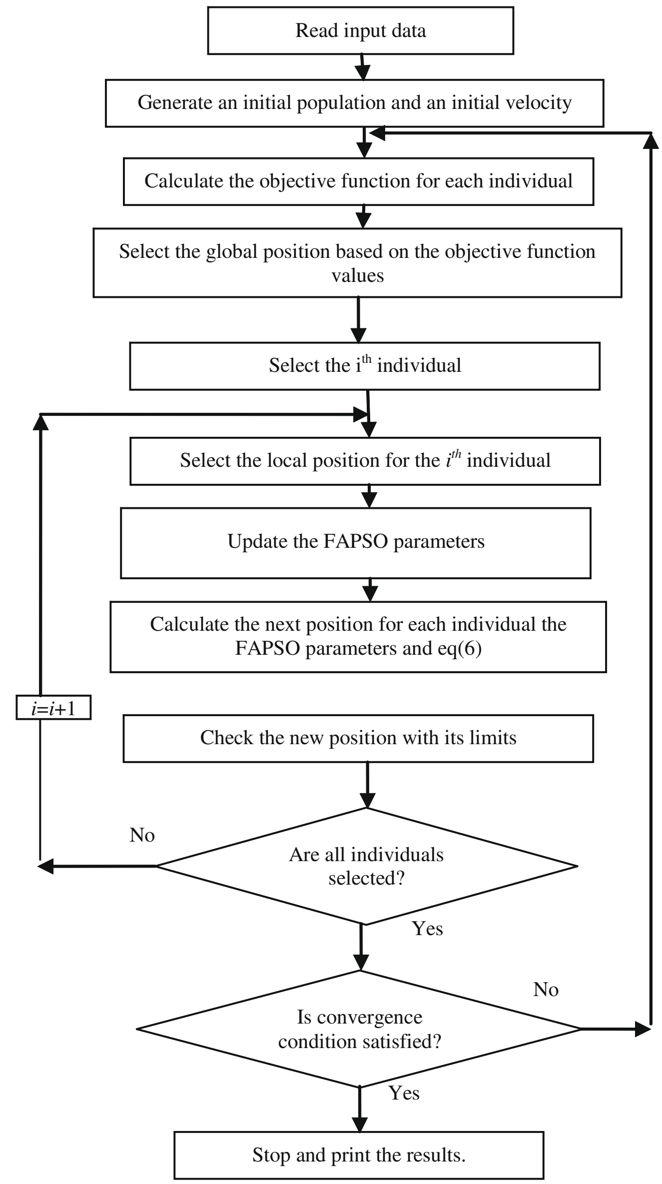

Fig. 4. Flowchart of the FAPSO algorithm.

$$X_{cent} = \frac{1}{N}\sum_{\substack{j=1\\ X_j \neq X_{high}}}^{N+1} X_j$$
$$X_{refl} = (1+\alpha)\times X_{cent} - \alpha \times X_{high} \tag{9}$$

where α is the reflection coefficient ($\alpha > 0$). Nelder and Mead suggested that $\alpha = 1$. If $J_{low} < J_{refl} < J_{highs}$, accept the reflection by replacing X_{high} with X_{refl}, and step 2 is repeated again for a new iteration. If $J_{refl} < J_{low}$, go to step 3. If $J_{high} > J_{refl} > J_{highs}$, replace X_{high} with X_{refl} and go to step 4. If $J_{high} < J_{refl}$, go to step 4 without the replacement of X_{high} by X_{refl}.

Step 3 Expansion

Should reflection produce a function value smaller than J_{low} (i.e., $J_{refl} < J_{low}$), the reflection is expanded in order to extend the search space in the same direction and the expansion point is calculated by the following equation:

$$X_{\exp} = \gamma \times X_{refl} + (1-\gamma)\times X_{cent} \tag{10}$$

where γ is the expansion coefficient ($\gamma > 1$). Nelder and Mead suggested $\gamma = 2$. If $J_{\exp} < J_{low}$, the expansion is accepted by replacing X_{high} with $X_{\exp}$; otherwise, $X_{\exp}$ replaces X_{high}. The algorithm continues with a new iteration in step 2.

Step 4 Contraction.

The contraction vertex is calculated by the following equation:

$$X_{cont} = \gamma \times X_{high} + (1 - \beta) \times X_{cent} \quad (11)$$

where β is the contraction coefficient ($0 < \beta < 1$). Nelder and Mead suggested $\beta = 0.5$. If $J_{cont} < J_{low}$, the contraction is accepted by replacing X_{high} with X_{cont} and then a new iteration begins with step 2. If $J_{cont} > J_{high}$ then go to step 5.

Step 5 Shrinkage

In this step, shrink the entire simplex except X_{low} by

$$X_i = \gamma \times X_{low} + (1 - \delta) \times X_{low} \\ 0 < \delta < 1 \quad (12)$$

Nelder and Mead suggested $\delta = 0.5$. Exit the algorithm if the stopping criteria are satisfied; otherwise return to step 2.

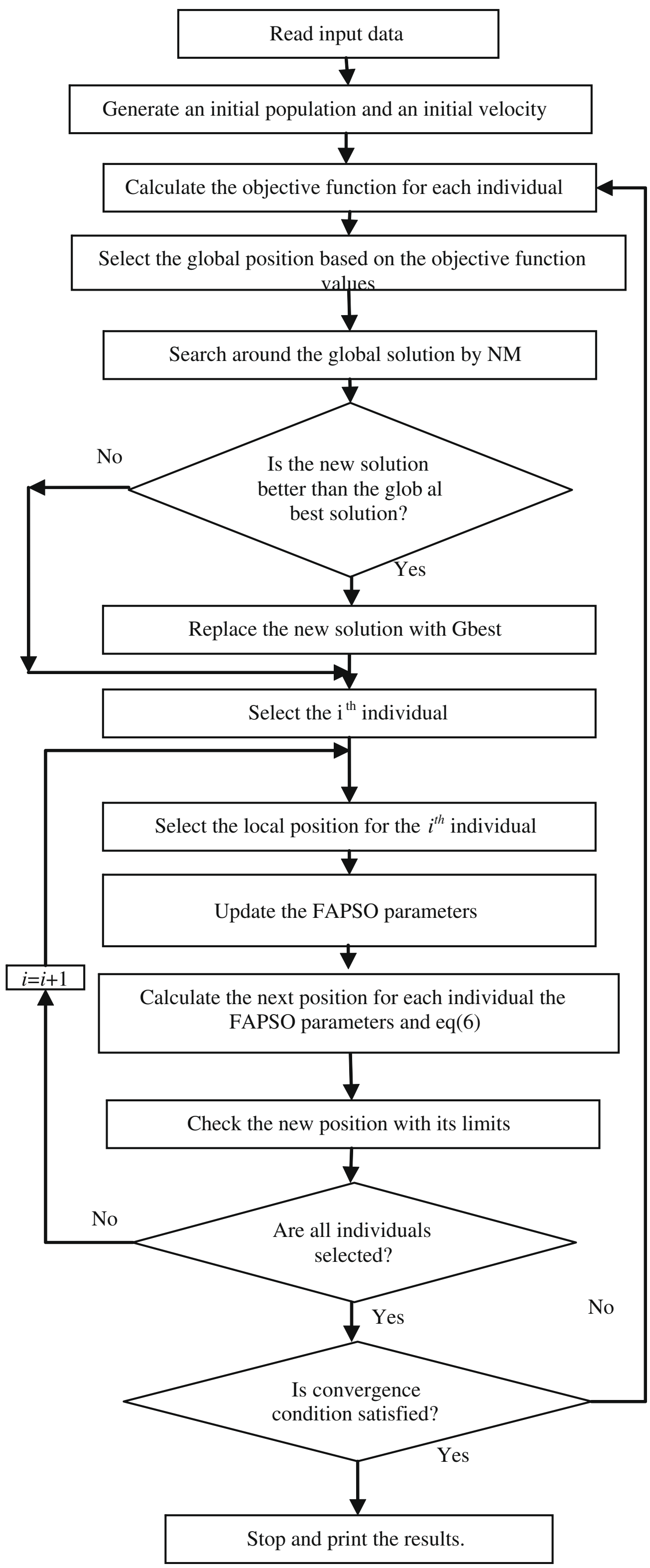

Fig. 5. Flowchart of FAPSO-NM.

5. Application of the hybrid FAPSO-NM algorithm on the ED problem

As mentioned in the previous sections, previous studies show that PSO has strong global search ability but, as a stochastic search algorithm, cannot guarantee convergence to the global optimal solution at the end. Also, the results are highly sensitive to the chosen values of parameters. On the other hand, the NM algorithm is a powerful local search algorithm. The FAPSO algorithm can be used to tune of PSO's parameters. This section presents and implements a new hybrid evolutionary algorithm which makes full use of the strong global search ability of the FAPSO algorithm and the strong local search ability of the NM algorithm. Therefore, they compensate the weaknesses of each other. In the proposed algorithm the FAPSO is considered as a main optimization algorithm and the NM algorithm is considered as a local search technique.

To implement the algorithm, the following steps should be taken:

Step 1 The input data including cost coefficients of the generators, output generator constraints, transmission loss matrix coefficients, and loads should be read.

Step 2 The proposed ED problem needs to be transformed into an unconstrained one by constructing an augmented objective function incorporating penalty factors for any value violating the constraints:

Table 4
Comparison of fuel costs for power demand of 1800 MW.

Method	Minimum cost ($/h)	Maximum cost ($/h)	Average cost ($/h)
FAPSO-NM	17963.84	**17964.21**	**17963.9577**
FAPSO	17963.84	17976.35	17969.9187
PSO	18030.72	18401.35	18205.9247
CEF [2]	18048.21	18404.04	18190.23
EP [15]	17994.07	–	18127.06
EP-PSO [2–3]	17991.03	–	18106.93
PSO-SQR [20]	17969.93	–	18029.99
DE [3]	**17963.83**	17975.36	17965.48
HGA [2]	**17963.83**	–	17988.04
FEP [2,15]	18018	18453.82	18200.79
MFEP [2,15]	18028.09	18416.89	18192
IFEP [2,15]	17994.07	18267.42	18127.06

$$H(X) = J(X) + k_1\left(\sum_{j=1}^{N_{eq}}(h_j(X))^2\right) + k_2\left(\sum_{j=1}^{N_{ueq}}(Max[0, -g_j(X)])^2\right) \tag{13}$$

$J(X)$ is the objective function value of the ED problem. N_{eq} and N_{ueq} are the number of equality and inequality constraints, respectively. $h_i(X)$ and $g_i(X)$ are the equality and inequality constraints, respectively. k_1 and k_2 are the penalty factors,. Since the constraints should be met, the value of the k_1 and k_2 parameters should be high. The chosen values for them are 10,000.

In the ED problem, the augmented objective function is calculated as follows:

At first, the distribution load flow is run based on the control variables. According to the results of the distribution load flow, the objective function value ($J(X)$) is calculated and the constraints are checked. Then the augmented objective function is calculated using the values of objective function and constraints.

Step 3 Generate an initial population and an initial velocity

An initial population, X_i, and an initial velocity, V_i, which must meet constraints, are generated randomly.

$$Population = \begin{bmatrix} X_1 \\ X_2 \\ \dots \\ X_{N_{Swarm}} \end{bmatrix}$$
$$X_i = [x_i^1, x_i^2, \dots, x_i^{Ng}], \tag{14}$$
$$x_i^j = rand(\cdot) \times (P_{gj,\max} - P_{gj,\min}) + P_{gj,\min}$$
$$j = 1, 2, 3, \dots, Ng$$
$$i = 1, 2, 3, \dots, N_{Swarm}$$

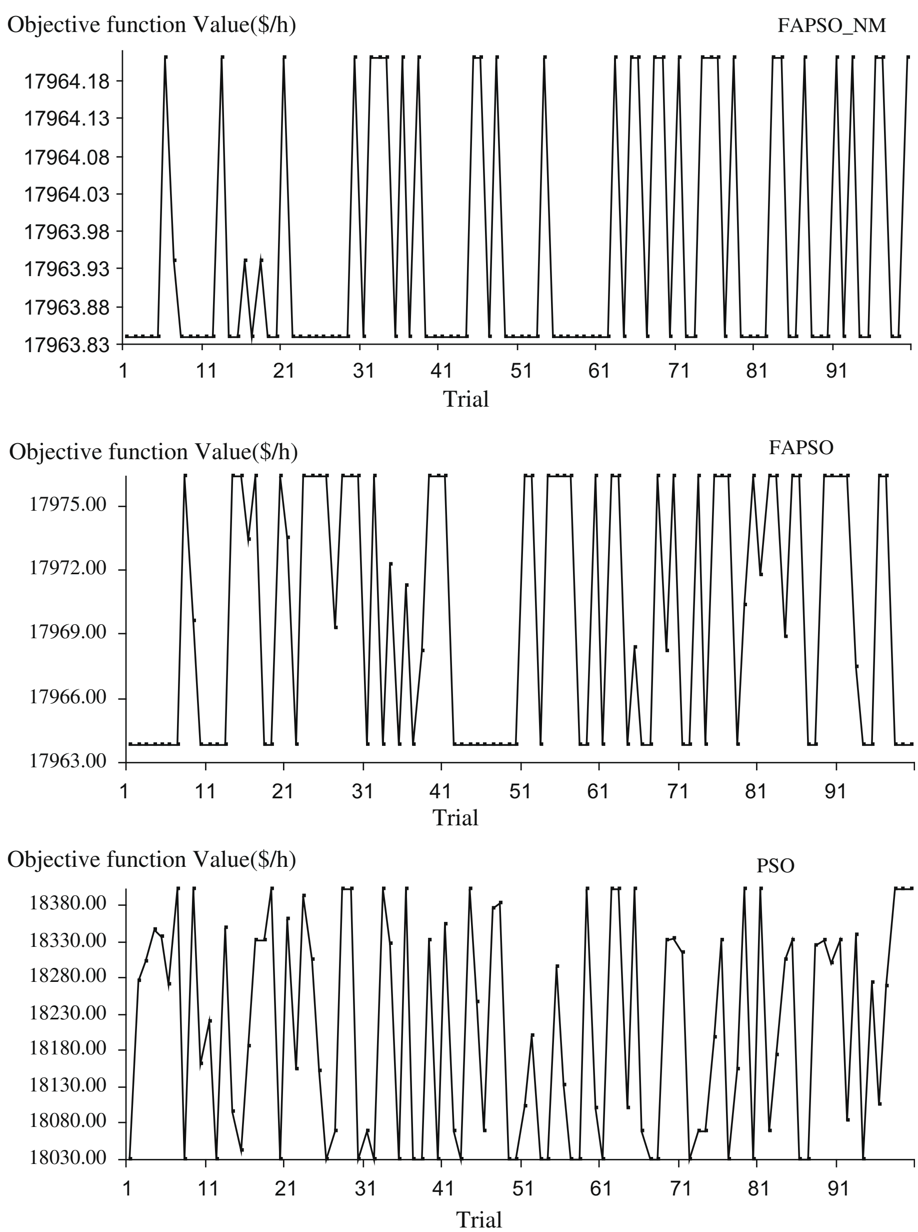

Fig. 6. Distribution of total costs of the FAPSO-NM, FAPSO and PSO algorithms for a load demand of 1800 MW for 100 different trials.

Table 5
Dispatch result of the proposed algorithm for power demand of 1800 MW.

Generator	Output power (MW)
Pg1	628.32
Pg2	222.75
Pg3	149.6
Pg4	109.87
Pg5	109.87
Pg6	109.87
Pg7	109.87
Pg8	60
Pg9	109.87
Pg10	40
Pg11	40
Pg12	55
Pg13	55

$$Velocity = \begin{bmatrix} V_1 \\ V_2 \\ \cdots \\ V_{N_{Swarm}} \end{bmatrix}$$
$$V_i = [v_i^1, v_i^2, ..., v_i^n], \quad v_i^j = 0.1 * (rand(\cdot) \times (P_{gj,\max} - P_{gj,\min}) + P_{gj,\min}) \quad j = 1,2,3,...,Ng \quad i = 1,2,3,...,N_{Swarm} \tag{15}$$

Table 6
CPU time of PSO, FAPSO and FAPSO-NM algorithms for power demand of 1800 MW.

Algorithm	CPU time (s)
FAPSO-NM	~6.8
FAPSO	~9
PSO	~11.89

where, $rand(\cdot)$ is a random function generator between 0 and 1.

Step 4 The augmented objective function (Eq. (13)) has to be evaluated for each individual using the result of distribution load flow.
Step 5 The individual, which has the minimum objective function, should be selected as the global position.
Step 6 Apply NM to search around the global solution.
Step 7 i = 1.
Step 8 The best local position ($Pbest_i$) is selected for the ith individual.
Step 9 Update the FAPSO parameters as described in the previous section.
Step 10 The modified velocity for the ith individual needs to be calculated based on the local and global positions, the FAPSO parameters, and Eq. (6).

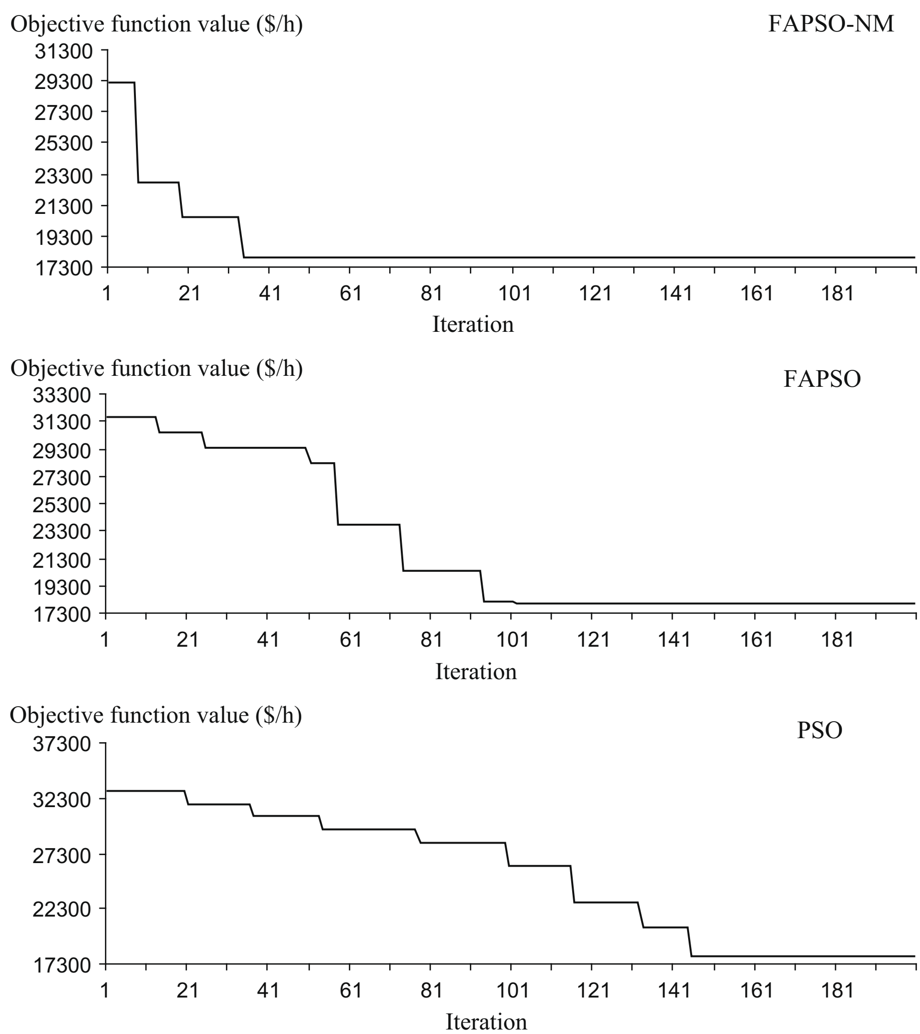

Fig. 7. Comparative convergence behaviors of the FAPSO-NM, FAPSO and PSO algorithms for a load demand of 1800 MW.

Table 7
Comparison of fuel costs for power demand of 2520 MW.

Method	Minimum cost ($/h)	Maximum cost ($/h)	Average cost ($/h)
FAPSO-NM	24169.92	**24170.5**	**24170.0017**
FAPSO	24170.93	24176.4	24173.0069
PSO	24262.73	24277.81	24271.9231
GA-SA [2,14]	24275.71	–	–
HGA [2]	24169.92	–	–
ESO [22]	24179.59	–	–
EP-PSO [2,23]	24266.44	–	–
PSO-SQR [20]	24261.05	–	–
DE [3]	**24169.9177**	–	–
GA [2,12]	24398.23	–	–
SA [2,14]	24970.91	–	–

Table 8
Dispatch result of the proposed algorithm for power demand of 2520 MW.

Generator	Output power (MW)
Pg1	628.32
Pg2	299.2
Pg3	299.98
Pg4	159.73
Pg5	159.73
Pg6	159.73
Pg7	159.73
Pg8	159.73
Pg9	159.73
Pg10	77.4
Pg11	77.4
Pg12	87.69
Pg13	92.4

Step 11 The modified position for *i*th individual should be calculated based on Eq. (6) and then checked with its limit.

Step 12 If all individuals are selected, go to the next step, otherwise $i = i + 1$ and go to step 7.

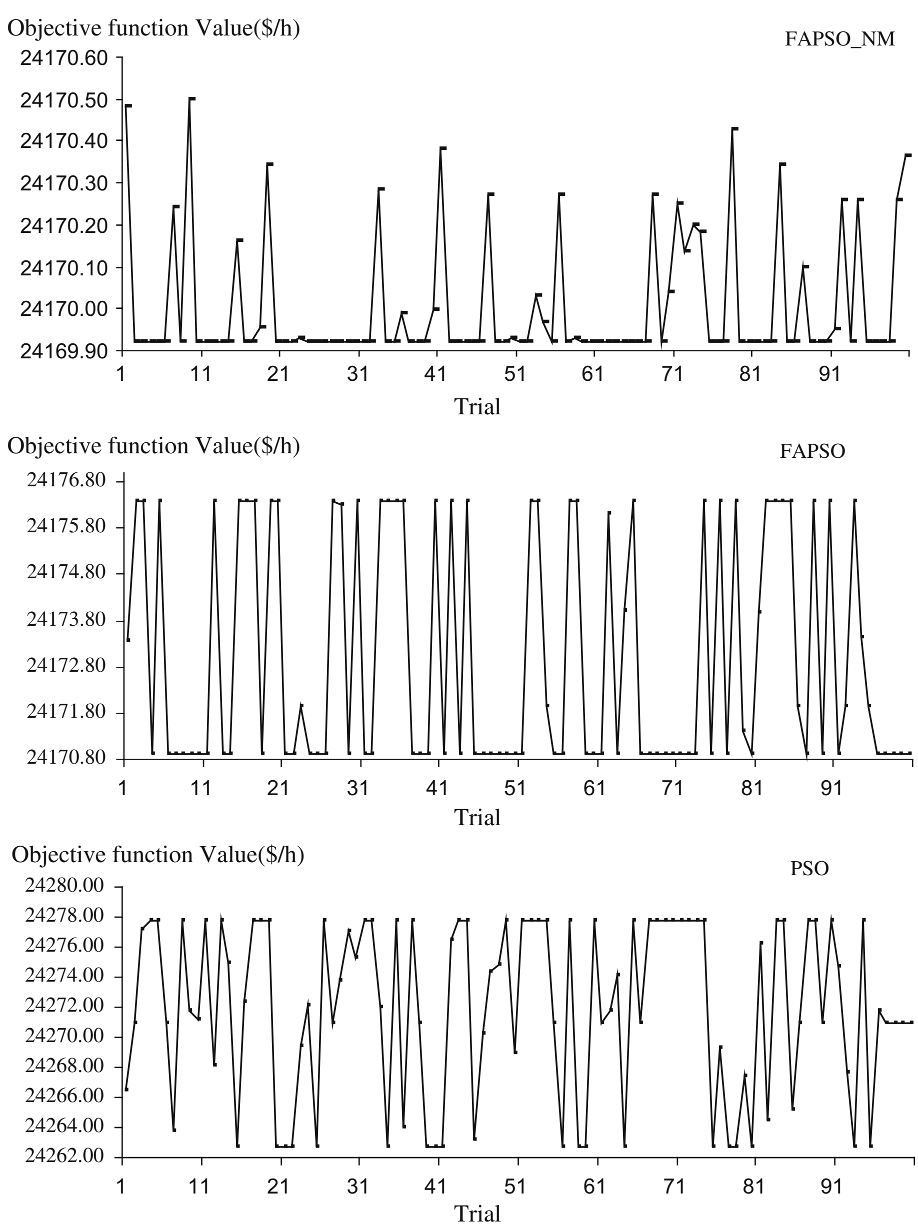

Fig. 8. Distribution of total costs of the FAPSO-NM, FAPSO and PSO algorithms for a load demand of 2520 MW for 100 different trials.

Step 13 If the current iteration number reaches the predetermined maximum iteration number, the search procedure is stopped, otherwise go to step 4.

The flowchart of the algorithm is shown in Fig. 5.

6. Simulation results

In order to evaluate the performance of proposed hybrid approach in solving the ED problem, two case studies (13 and 40 thermal units or generators) are considered in this section. MATLAB 7.0 was used to implement the algorithm.

6.1. Case 1: 13 thermal generators

The first case study includes 13 thermal generation units with the effects of valve-point loading. The expected power demands to be satisfied by the 13 generating units are 1800 MW [15] and 2520 MW [21]. The system data is given in [15]. The problem has a number of local optimum solutions and any method may be trapped in one of them. The problem is solved for two different power demands in order to illustrate the efficiency of the proposed method in obtaining high-quality solutions.

Due to the randomness of the evolutionary algorithms, their performances cannot be judged by the result of a single run. Many trials with different initializations should be made to reach a valid conclusion about the performance of the algorithms. An algorithm is robust, if it can guarantee an acceptable performance level under different conditions. In this paper, 100 different runs have been carried out. The population size of the proposed algorithm (N_{swarm}) and the number of iterations are 26 and 300, respectively. The simulation results of the proposed hybrid algorithm and others for different trials in terms of minimum cost, maximum cost, and average cost for power demand of 1800 MW have been shown in Table 4. The bold prints in the table illustrate the best solutions among all solutions found by the algorithms.

Fig. 6 illustrates the distribution of the results obtained by the proposed algorithm, FAPSO and the original PSO for 100 different runs.

From the computational results of Table 4, the minimum cost, maximum cost and the mean cost values obtained by the proposed algorithm are 17963.84 $/h, 17964.21 $/h, and 17964.01 $/h, respectively, which are slightly lower than those obtained by the

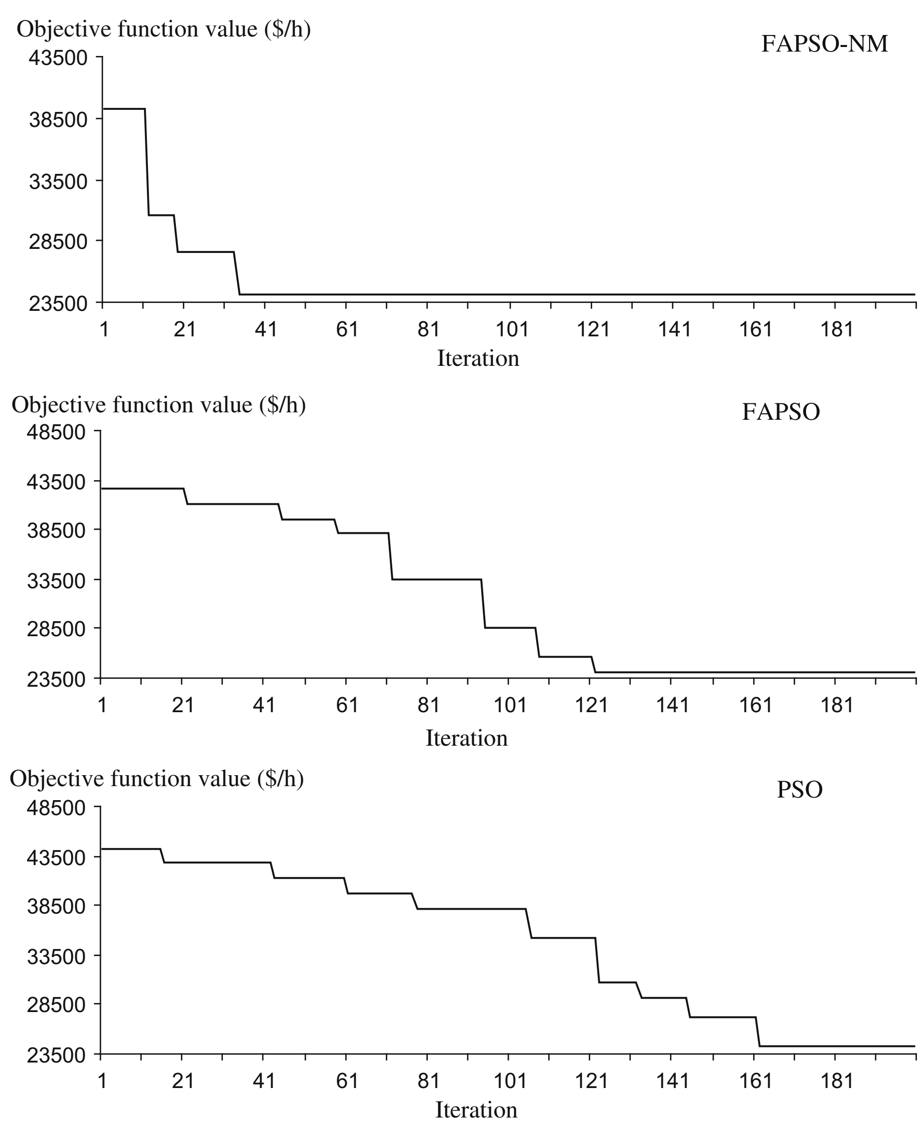

Fig. 9. Comparative convergence behaviors of the FAPSO-NM, FAPSO and PSO algorithms for a load demand of 2520 MW.

Table 9
CPU time of PSO, FAPSO and FAPSO-NM algorithms for power demand of 2520 MW.

Algorithm	CPU time (s)
FAPSO-NM	~6.8
FAPSO	~9
PSO	~11.89

other methods. The results of Fig. 4 show that variations of the total cost obtained by the FAPSO-NM are in small range for different runs. Also the results of the figure show that FAPSO-NM and FAPSO converge to the global solution of 17963.84 in 70 and 46 times, respectively, while the original PSO algorithm does not converge to this solution and converges to the global solution of 18030.72.

It is obvious from Table 4 and Fig. 6 that the results of the FPSO-NM algorithm are very close to minimum value. In other words, FAPSO-NM has the small standard deviation for different runs. That means that FAPSO-NM is robust.

Table 10
Comparison of fuel costs for 40 generators.

Method	Minimum cost ($/h)	Maximum cost ($/h)	Average cost ($/h)
FAPSO-NM	121418.3	**121419.8**	**121418.803**
FAPSO	121712.4	121873.17	121778.246
EP	122624.35	–	123,382
EP-SQP	122323.97	–	122379.63
MPSO	122252.27	–	–
PSO	123930.45	124312.63	124,155
PSO-SQP	122094.67	–	122295.13
DEC-SQP	121074.98	–	122295.13
HGA	**121418.27**	–	121784.04
IFEP	122624.35	125740.63	123,382
ESO	122122.16	123143.07	122524.07
NPSO	121704.74	122995.09	122221.37
SOHPSO	121501.14	122446.3	121853.57
CEF	123488.29	126902.89	124793.48
FEP	122679.71	127245.59	124119.37
MFEP	122647.57	124356.47	123484.74
HDE	121698.51	–	122304.30
DE	121416.29	121431.47	121422.72

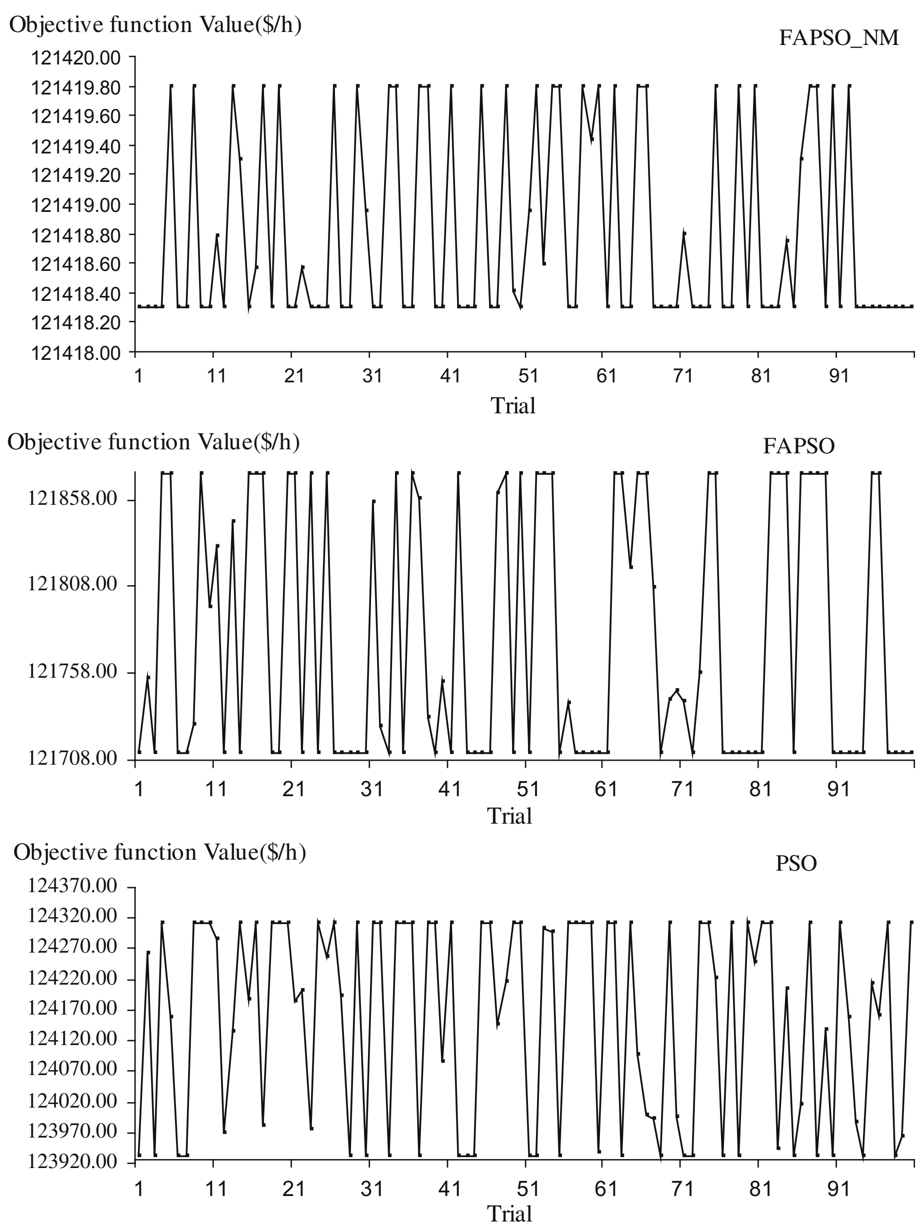

Fig. 10. Distribution of total costs of the FAPSO-NM, FAPSO and PSO algorithms for 40 generators for 100 different trials.

Table 5 shows the simulation results of FPSO-NM for the best solution.

The speed of convergence of the proposed algorithm was also tested. Fig. 7 shows the required number of iterations for the proposed hybrid algorithm, the FAPSO, and the original PSO in order to converge to the best solution found by respective algorithms.

The results of the convergence test suggest that the FAPSO-NM algorithm is the best compared to the others in terms of the required number of iterations.

Table 6 reveals the CPU time of the FAPSO-NM, FAPSO and PSO algorithms.

The simulation results in Table 6 illustrate that the execution time of the FAPSO-NM is significantly short with respect to FAPSO and PSO and provides a general idea that the FAPSO-NM can be utilized without any restriction in practical networks.

Table 7 illustrates the simulation results of the different methods and the proposed method for a load demand of 2520 MW. The best solutions among all solutions have been shown in the bold prints.

Table 8 shows that among the implemented methods, the FPSO-NM algorithm has the best performance in terms of the mean cost value, the minimum cost, and the average cost. To verify the solution quality obtained by the proposed algorithm, Fig. 8 shows the distribution of the results obtained by the proposed algorithm, FAPSO and the original PSO for 100 different runs.

Table 8 illustrates the dispatch results of FPSO-NM for the best solution for power demand of 2520 MW.

Fig. 9 shows the convergence characteristics of the FPSO-NM, FPSO, and PSO for power demand of 2520 MW. The figure demonstrates that the hybrid algorithm converges to the global optimal solution after 35 iterations while the FPSO and PSO algorithm converge to global solutions after 120 and 164 iterations, respectively.

Table 9 shows the execution time of FAPSO, FAPSO and PSO.

The execution time of the proposed algorithm is less than the FAPSO and original PSO algorithms. The reason is that the number of particle swarms in the FAPSO-NM algorithm is 26 while in the FAPSO and original PSO is 35.

6.2. Case2. 40 thermal units

The second case study includes 40 generators with valve-point loading effects and has a total load of 10,500 MW [15]. The system has several local minima, and the global minimum is very difficult to find. To utilize the FAPSO algorithm for this case study, 100 different runs have been carried out. The population size of the proposed algorithm and the number of iterations are 60 and 1000, respectively. In order to compare results with other algorithms, it is assumed that the test system is lossless.

The final fuel costs obtained using the FPSO-NM, FAPSO, PSO [2–3], EP [2–3], EP–SQP [2–3], DEC-SQP [2–3], PSO–SQP [2–3], HGA [2–3], modified PSO, HDE [2–3], GA [2–3], NPSO [24], SOHPSO [25], HGA [2–3], IFEP [2,3,24,25], MFEP [2,3,24,25], DEC-SQP [2,3] and DE [2,3] are summarized in Table 10. The best solutions among all solutions have been illustrated in the bold prints.

From Table 10, it is clear that the results of the proposed algorithm are better than those obtained by the other algorithms. Also, the maximum cost obtained by the proposed algorithm is even better than the minimum cost obtained by several algorithms. The minimum cost obtained by DE is better than others. However, the average cost and maximum cost obtained by the proposed algorithm are better than those obtained by DE.

Fig. 10 shows the variations of output results of obtained by FAPSO-NM, FAPSO and PSO for 100 different runs.

The simulation results indicate that the variation of results of the FAPSO-NM algorithm is in the small range with respect to the others. Also, the proposed algorithm converges to global solution in 70 times.

Table 11
Dispatch result of the proposed algorithm for 40 generators.

Generator	Output power (MW)	Generator	Output power (MW)
Pg1	111.38	Pg21	523.33
Pg2	110.93	Pg22	523.48
Pg3	97.41	Pg23	523.33
Pg4	179.33	Pg24	523.33
Pg5	89.22	Pg25	523.33
Pg6	140	Pg26	523.33
Pg7	259.62	Pg27	10
Pg8	284.66	Pg28	10
Pg9	284.66	Pg29	10
Pg10	130	Pg30	88.7
Pg11	168.82	Pg31	190
Pg12	168.82	Pg32	190
Pg13	214.75	Pg33	190
Pg14	394.28	Pg34	165
Pg15	304.54	Pg35	166
Pg16	394.3	Pg36	165
Pg17	489.29	Pg37	110
Pg18	489.29	Pg38	110
Pg19	511.28	Pg39	110
Pg20	511.29	Pg40	511.3

Table 12
CPU time of PSO, FAPSO and FAPSO-NM algorithms for 40 generators.

Algorithm	CPU time (s)
FAPSO-NM	~40
FAPSO	~87
PSO	~152

Table 11 shows the simulation results obtained by FAPSO-NM for the best solution.

The CPU execution time of proposed algorithm, FAPSO and PSO is shown in Table 12.

The simulation results for different case studies show the superiority of the FAPSO-NM algorithm over other methods. In other words, the simulation results show that the FAPSO-NM algorithm converges to global solution has a short run time and small standard deviation for different trails.

7. Conclusion

In this paper, a new hybrid optimization algorithm based on the combination of FAPSO and NM, called FAPSO-NM, was presented for solving economic dispatch problem in power systems considering the valve-point effects. In general, the cost function of the generating units is non-smooth and non-convex. The valve-point effects have modeled and imposed as rectified sinusoid components. PSO is an efficient tool for solving complex optimization problems. It is utilized to solve different problems in diverse fields. Also, it has been successfully used to solve complex problems related to the field of power systems such as the ED problem. The results of PSO greatly depend on the parameter values and the method often suffers from the problem of being trapped in local optima. To overcome these drawbacks, in this paper, the parameters of PSO have been adjusted using fuzzy IF/THEN rules, and in order to reach the global solution, the algorithm has used the NM algorithm as a local search technique around the best solution found by FAPSO. The effectiveness of the hybrid algorithm has been verified by computer experiments. Two typical ED problems with valve-point affects were considered and the performances of different algorithms were compared. The numerical results reveal the superiority and feasibility of the proposed hybrid approach compared to other methods.

The simulation results indicate that this optimization method is very accurate and converges very rapidly so that it can be used in the practical optimization problems. The objective functions of the problems can be considered differentiable, non-differentiable, convex and non-convex and the variables can be considered continuous and discrete.

References

[1] Vahidinasab V, Jadid S. Multiobjective environmental/techno-economic approach for strategic bidding in energy markets. Appl Energy 2009;86:496–504.
[2] He DK, Wang FL, Mao ZZ. Hybrid genetic algorithm for economic dispatch with valve-point effect. Electr Power Syst Res 2008;78:626–33.
[3] Subbaraj P, Rengaraj R, Salivahanan S. Enhancement of combined heat and power economic dispatch using self adaptive real-coded genetic algorithm. Appl Energy 2009;86:915–21.
[4] Ashwani K, Wenzhong G. Pattern of secure bilateral transactions ensuring power economic dispatch in hybrid electricity markets. Appl Energy 2009;86:1000–1010l.
[5] Noman N, Iba H. Differential evolution for economic load dispatch problems. Electr Power Syst Res 2008;78:1322–31.
[6] Chiou JP. Variable scaling hybrid differential evolution for large-scale economic dispatch problems. Electr Power Syst Res 2007;77:212–8.
[7] Selvakumar AI, Thanushkodi K. Anti-predatory particle swarm optimization: solution to nonconvex economic dispatch problems. Electr Power Syst Res 2008;78:2–10.
[8] Papageorgiou LG, Fraga ES. A mixed integer quadratic programming formulation for the economic dispatch of generators with prohibited operating zones. Electr Power Syst Res 2007;77:1292–6.
[9] Wang S-K, Chiou J-P, Liu C-W. Non-smooth/nonconvex economic dispatch by a novel hybrid differential evolution algorithm. IET Gen Trans Distrib 2007;1(5):793–803.
[10] Cai J, Ma X, Li L, Yang Y, Peng H, Wang X. Chaotic ant swarm optimization to economic dispatch. Electr Power Syst Res 2007;77:1373–80.
[11] Chen PH, Chang H-C. Large-scale economic dispatch by genetic algorithm. IEEE Trans Power Syst 1995;10(4):1919–26.
[12] Walters DC, Sheble GB. Genetic algorithm solution of economic dispatch with valve point loading. IEEE Trans Power Syst 1993;8(3):1325–32.
[13] Baskar S, Subbaraj P, Rao MVC. Hybrid real coded genetic algorithm solution to economic dispatch problem. Comput Electr Eng 2003;29(3):407–19.
[14] Wong KP, Wong YW. Genetic and genetic/simulated-annealing approaches to economic dispatch. IEE Proc Gen Transm Distrib 1994;141(5):507–13.
[15] Sinha N, Chakrabarti R, Chattopadhyay PK. Evolutionary programming techniques for economic load dispatch. IEEE Trans Evol Comput 2003;7(1):83–94.
[16] Lin VM, Cheng FS, Tsay MT. An improved tabu search for economic dispatch with multiple minima. IEEE Trans Power Syst 2002;17(1):108–12.
[17] Park JB, Lee K-S, Shin J-R, Lee KY. A particle swarm optimization for economic dispatch with non-smooth cost functions. IEEE Trans Power Syst 2005;20(1):34–42.
[18] Gaing ZL. Particle swarm optimization to solving the economic dispatch considering the generator constraints. IEEE Trans Power Syst 2003;18(3):1187–95.
[19] Yuan X, Su A, Yuan Y, Nie H, Wang L. An improved PSO for dynamic load dispatch of generators with valve-point effects. Energy J 2009;34(1):67–74.
[20] Victoire TAA, Jeyakumar AE. Hybrid PSO-SQP for economic dispatch with valve-point effect. Electr Power Syst Res 2004;71(1):51–9.
[21] Dos L, Coelho S, Mariani VC. Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect. IEEE Trans Power Syst 2006;21(2):989–96.
[22] Neto AP, Unsihuay C, Saavedra OR. Efficient evolutionary strategy optimisation procedure to solve the nonconvex economic dispatch problem with generator constraints. IEE Proc Gen Transm Distrib 2005;152(5):653–60.
[23] Swarup KS, Kumar PR. A new evolutionary computation technique for economic dispatch with security constraints. Int J Electr Power Energy Syst 2006;28(4):273–83.
[24] Niknam T, Khodaei A, Fallahi F. A new decomposition approach for the thermal unit commitment problem. Appl Energy 2009;86(9):1667–74.
[25] Selvakumar AI, Khanushkodi T. A new particle swarm optimization solution to nonconvex economic dispatch problems. IEEE Trans Power Syst 2007;22(1):42–51.
[26] Chaturvedi KT, Pandit M, Srivastava L. Self-organizing hierarchical particle swarm optimization for nonconvex economic dispatch. IEEE Trans Power Syst 2008;23(3):1079–87.
[27] Kennedy J, Eberhart R. Particle swarm optimization. IEEE Int Conf Neural Networks 1995;4:1942–8 [Piscataway, NJ].
[28] Niknam T, Nayeripour M, Olamaei J, Arefi A. An efficient hybrid evolutionary optimization algorithm for daily Volt/Var control at distribution system including DGs. Int Rev Electr Eng 2008;3(3):1–11.
[29] Olamaei J, Niknam T, Gharehpetian G. Application of particle swarm optimization for distribution feeder reconfiguration considering distributed generators. Appl Math Comput J 2008;200(1–2):575–86.
[30] Saber AY, Senjyu T, Yona A, Funabashi T. Unit commitment computation by fuzzy adaptive particle swarm optimization. IET Gen Transm Distrib 2007;1(3):456–65.
[31] Bajpai P, Singh SN. Fuzzy adaptive particle swarm optimization for bidding strategy in uniform price spot market. IEEE Trans Power Syst 2007;22(4):2152–60.
[32] Zhang W, Liu Y. Multi-objective reactive power and voltage control based on fuzzy optimization strategy and fuzzy adaptive particle swarm. Int J Electr Power Energy Syst 2008;30(9):525–32.
[33] Zahara E, Kao YT. Hybrid Nelder–Mead simplex search and particle swarm optimization for constrained engineering design problems. Expert Syst Appl J 2009;36(2):3880–6.
[34] Shu-Kai SF, Yun-Chia L, Zahara E. A genetic algorithm and a particle swarm optimizer hybridized with Nelder–Mead simplex search. Comput Indus Eng 2006;50(4):401–25.
[35] Chelouah R, Siarry P. A hybrid method combining continuous tabu search and Nelder–Mead simplex algorithms for the global optimization of multiminima functions. Eur J Operation Res 2005;161(3):636–54.