Memetic Comp. (2017) 9:91–108
DOI 10.1007/s12293-016-0186-1

REGULAR RESEARCH PAPER

Swarm based mean-variance mapping optimization for convex and non-convex economic dispatch problems

T. H. Khoa[1] · P. M. Vasant[1] · M. S. Balbir Singh[1] · V. N. Dieu[2]

Received: 18 November 2014 / Accepted: 4 April 2016 / Published online: 27 April 2016
© Springer-Verlag Berlin Heidelberg 2016

Abstract In power system generation, the economic dispatch (ED) is used to allocate the real power output of thermal generating units to meet the required load demand so as the total cost of thermal generating units is minimized. This paper proposes a swarm based mean-variance mapping optimization ($MVMO^S$) for solving the ED problems with convex and nonconvex objective functions. The proposed method is the extension of the original single particle mean-variance mapping optimization by initializing a set of particles. The special feature of the proposed method is a mapping function applied for the mutation based on the mean and variance of n-best population. The proposed $MVMO^S$ is tested on various systems and the obtained results are compared to those from many other optimization methods in the literature. Test results have shown that the proposed method can obtain better solution quality than the other methods. Therefore, the proposed $MVMO^S$ is a potential method for efficiently solving the convex and nonconvex ED problems in power systems.

Keywords Mean-variance mapping optimization · Economic dispatch · Valve-point effects · Prohibited operating zone · Cubic fuel cost function · Quadratic fuel cost function

✉ T. H. Khoa
trhkhoa89@gmail.com

P. M. Vasant
pvasant@gmail.com

M. S. Balbir Singh
balbir@petronas.com.my

V. N. Dieu
vndieu@gmail.com

[1] Department of Fundamental and Applied Sciences, Universiti Teknologi PETRONAS, Tronoh, Malaysia

[2] Department of Power Systems, HCMC University of Technology, Ho Chi Minh City, Vietnam

Nomenclature

N	Total number of generating units
F	Total operation cost
a_i, b_i, c_i	Fuel cost coefficients of unit i
e_i, f_i	Fuel cost coefficients of unit i reflecting valve-point effects
B_{ij}, B_{0i}, B_{00}	B-matrix coefficients for transmission power loss
P_D	Total system load demand
P_i	Power output of generator i
$P_{i,max}$	Maximum power output of generator i
$P_{i,min}$	Minimum power output of generator i
P_s	Power output of slack unit
$P_{s,max}$	Maximum power output of slack unit
P_{ismin}	Minimum power output of slack unit
n_i	Number of prohibited operating zones of unit i
P_L	Total transmission loss
P_{ik}^l	Lower bound for prohibited zone k of generator i
P_{ik}^u	Upper bound for prohibited zone k of generator i
DR_i	Ramp down rate limit of unit i
UR_i	Ramp up rate limit of unit i
S_i	Spinning reserve from unit i
$S_{i,max}$	Maximum spinning reserve contribution of unit i
S_R	Total system spinning reserve requirement
n_var	Number of variable (generators)

n_par	Number of particles
mode	Variable selection strategy for offspring creation
archive zize	*n*-best individuals to be stored in the table
d_i	Initial smoothing factor
Δd_0^{ini}	Initial smoothing factor increment
$\Delta d_0^{\text{final}}$	Final smoothing factor increment
$f_{s\_ini}^*$	Initial shape scaling factor
$f_{s\_\text{final}}^*$	Final shape scaling factor
D_{min}	Minimum distance threshold to the global best solution
n_randomly	Initial number of variables selected for mutation
n_randomly_min	Final number of variables selected for mutation
indep.runs	*m* steps independently to collect a set of reliable individual solutions

1 Introduction

The economic dispatch (ED) is one of the power management tools that is used to determine real power output of thermal generating units to meet required load demand. The ED results in minimum fuel generation cost, minimum transmission power loss while satisfying all units, as well as system constraints [1,2].

The ED problems may be generally classified into convex and nonconvex optimization problem based on the nature characteristic of the generating units. In the convex ED, the operation cost function is usually approximated by quadratic function. However, the objective function of the ED is more accurate when the cubic function is considered to express the input-output characteristics of thermal generators. Several methods are proposed in the literature for solving ED with cubic fuel cost function such as iterative dynamic programming (DP) [3], Newton approach [4], genetic algorithm (GA) [5] particle swarm optimization (PSO) [6], λ-logic based algorithm [7,8] and firefly algorithm (FA) [9]. The ED problem can be represented more exactly by considering various nonconvex elements and nonlinearities in the objective function and constraints such as valve point effects, prohibited operating zones, ramp rate limits and spinning reserve. These effects can cause the input-output curve of thermal generators to become more complicated. For this reason, the practical ED problem should be formulated with a nonconvex objective function which is difficult to find global solution. Many heuristic search approaches are presented in the literature for solving the nonconvex ED problems such as hopfield neural network (HNN) [10–12], genetic algorithm (GA) [13–15], evolutionary programming (EP) [16], evolutionary algorithm (EA) [17], simulated annealing (SA) [18], artificial bee colony (ABC) [19], evolutionary algorithm [17], artificial immune system (AIS) [20], biogeography-based optimization (BBO) [21], and differential evolution (DE) [22,23], particle swarm optimization (PSO) [24,25]. Recently, PSO is the most popular method applied for solving the ED problems, especially for nonconvex problems. Several improvements of PSO method are developed for solving the ED problems such as self-organizing hierarchical PSO (SOH_PSO) [26], simulated annealing like particle swarm optimization (SA-PSO) [27], pseudo-gradient based particle swarm optimization (PGPSO) [2], new PSO with local random search (NPSO-LRS) [28], new adaptive particle swarm optimization (NAPSO) [29], Chaotic particle swarm optimization (CPSO) [30]. These improved PSO methods can obtain high quality solutions for the problem. The PSO method is continuously improved for dealing with large-scale and complex problems such as in [31,32]. Although, heuristic search methods can deal with complex optimization problems, their search ability often provides near global optimal solution. The nonconvex ED problems have been also solved by many hybrid optimization methods such as hybrid approach based on sequential combination of GA and active power optimization (APO) using Newton's second order approach (GA-APO, NSOA) [33], self-adaptive differential evolution with augmented Lagrange multiplier method (SADE-ALM) [34], integrated artificial intelligence (ETQ) [35], bacterial foraging optimization with Nelder –Mead algorithm (Adaptive BF with NM) [36]. These hybrid methods become powerful search methods for obtaining higher solution quality due to using the advantages of each element method to improve their search ability for the complex problems. However, the hydrid methods may be slower and more complicated than the single methods because of combination of several operations. The nonconvex optimization problem is still a challenge for solution methods. Hence, there is always a need for developing new techniques for solving nonconvex problems.

Recently, Mean-variance mapping optimization (MVMO) is a new meta-heuristic search algorithm which is developed by István Erlich [37]. This algorithm falls into the category of the so-called "population-based stochastic optimization technique". The similarities between MVMO and the other known stochastic algorithms are in three evolutionary operators including selection, mutation and crossover. The extensions of MVMO is also developed by Rueda & Erlich, which named swarm based mean-variance mapping optimization ($MVMO^S$) [38]. Unlike the single particle MVMO, the search process of $MVMO^S$ is started with a set of particles. In addition, two parameters of MVMO including the scaling factor and variable increment parameters have been extended to enhance the mapping. Hence, the ability for global search of ($MVMO^S$) is more effective than the original version. The $MVMO^S$ has been successfully implemented for solving the ED problems [39]. However, the ED was only considered as nonconvex problem which takes into account valve-point effects, multiple fuel options and prohibited operating zones. In this paper, the $MVMO^S$ is proposed as a new method for solving the ED problems in general. Both convex and nonconvex ED problems are considered in this study. For convex ED problem, the cubic fuel cost function is considered beside the quadratic fuel cost function which is the basic ED problem. For nonconvex ED problem, besides the formulated ED problems in [39], we have proposed a new problem which combines prohibited operating zones with valve-point loading effects. Moreover, the spinning reserve constraint and large-scale test systems have been proposed for the ED problem with prohibited operating zones which were not mentioned in [39]. The proposed $MVMO^S$ is tested on several convex and nonconvex systems and the obtained results are compared to those from many other methods in the literature. The comparisons have shown that the $MVMO^S$ method is more effective and provides better solution quality than the other methods in the literature for the problem in terms of optimal solution, especially for large-scale systems. Therefore, the $MVMO^S$ is a favorable method for solving the ED problems.

The remaining organization of this paper is as follows. Section 2 presents the formulation of the ED including convex and nonconvex problems. Handling of constraints and implementation of the proposed $MVMO^S$ to ED problem are addressed in Sect. 3. Section 4 reports results of the proposed $MVMO^S$ method. A number of case studies using standard test systems are used to test the proposed method. The comparisons of results between the proposed method and existing methods are also carried out in this section. The discussion is followed in Sect. 5. After all, the conclusion is given.

2 Problem formulation

2.1 Convex economic dispatch problem

The objective function of the ED problem is to minimize the total production cost, which be written as:

$$Minimize\ F_T = \sum_{i=1}^{N} F_i(P_i) \qquad i = 1, 2, \ldots, N \tag{1}$$

Mathematically, the fuel cost of a thermal generation unit is represented as quadratic function [1]:

$$F_i(P_i) = a_i + b_i P_i + c_i P_i^2 \tag{2}$$

The solution of ED can be highly improved by introducing higher order generator cost functions. Cubic cost function displays the actual response of thermal generators more accurately. The cubic fuel cost function of a thermal generating unit is represented as follows [4]:

$$F_i(P_i) = a_i + b_i P_i + c_i P_i^2 + d_i P_i^3 \tag{3}$$

subject to

Real power balance equation The total active power output of generating units must be equal to total power load demand plus power loss:

$$\sum_{i=1}^{N} P_i = P_D + P_L \tag{4}$$

where the power loss P_L is calculated by the below formulation [1]:

$$P_L = \sum_{i=1}^{N}\sum_{j=1}^{N} P_i B_{ij} P_j + \sum_{i=1}^{N} B_{0i} P_i + B_{00} \tag{5}$$

Generator capacity limits The active power output of generating units must be within the allowed limits:

$$P_{i,\min} \le P_i \le P_{i,\max} \tag{6}$$

2.2 Nonconvex economic dispatch problems

2.2.1 ED problem with valve point effects

The valve point effects (VPE) is considered as practical operation of thermal generating units. When each steam valve in a turbine of the thermal unit starts to open, produces a rippling effect on the input-output curve. The ripples are shown in Fig. 1. The VPE makes the fuel cost function highly nonlinear and having multiple local optimum. The fuel cost function is described as the superposition of sinusoidal functions and quadratic functions. The model of ED problem with VPE is formulated as follows [2]:

$$F_i(P_i) = a_i + b_i P_i + c_i P_i^2 + \left| e_i \sin(f_i(P_{i,\min} - P_i)) \right| \tag{7}$$

subject to the real power balance constraint in Eq. (4) and generator capacity limits in Eq. (6).

2.2.2 ED problem with prohibited operating zones

The prohibited operating zones (POZ) are the range of output power where the thermal generating unit must avoid operating because it causes undue vibration of the turbine shaft and might bring damage to the shaft and bearings. The cost curve function of units with prohibited zones is described in Fig. 2 [2].

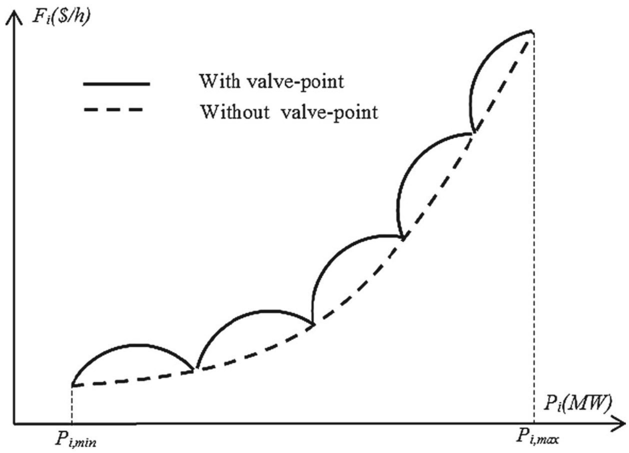


Fig. 1 Fuel cost curve of units with valve-point effects

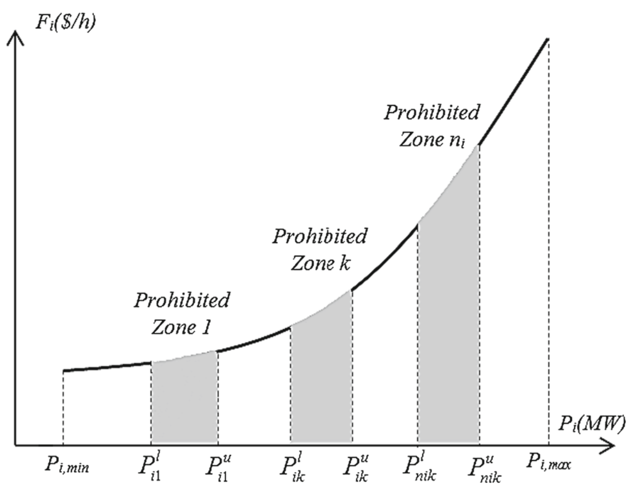


Fig. 2 Fuel cost curve of units with prohibited zones

The objective function for the ED problem with POZ can be a quadratic function in Eq. (2) or a quadratic function with VPE in Eq. (7). For units without POZ, only the equality constraint in Eq. (4) and inequality constraint in Eq. (6) are considered. As for units operating with POZ, more constraints are added to the constraints mentioned above as follows:

Prohibited operating zone constraint The feasible operating points should be located at one of the sub-regions as follows [2]:

$$P_i \in \begin{cases} P_{i,\min} \le P_i \le P^l_{i1} \\ P^u_{ik-1} \le P_i \le P^l_{ik}, k = 2, \ldots, n_i \\ P^u_{in_i} \le P_i \le P_{i,\max} \end{cases} \quad (8)$$

Spinning reserve constraint The spinning reserve constraint for all units is defined as [17]:

$$\sum_{i=1}^{N} S_i \ge S_R \quad (9)$$

where the operating margin of each unit Si is determined by:

$$S_i = \min\left\{P_{i,\max} - P_i, S_{i,\max}\right\}; \ \forall i \notin \Omega \quad (10)$$

$$S_i = 0; \quad \forall i \in \Omega \quad (11)$$

Ramp rate limit constraints The increased or decreased power output of a unit from its initial operating point to the next one should not exceed its ramp up and down rate limits. The ramp rate constraints are determined by [2]:

$$P_i - P_{i0} \le UR_i, \text{ if generation increases} \quad (12)$$

$$P_{i0} - P_i \le DR_i, \text{ if generation decreases} \quad (13)$$

To handle the ramp rate limits, the highest and lowest possible power outputs of units are determined based on their power output limits, the generator capacity limits in Eq. (6) can be rewritten as follows [2]:

$$\max(P_{i,\min}, P_{i0}-DR) \le P_i \le \min(P_{i,\max}, P_{i0}+UR) \quad (14)$$

3 MVMO^S for ED problems

3.1 MVMO^S

MVMO^S is an extension of the original version MVMO. The difference between MVMO and MVMO^S is the initial search process with particles. MVMO starts the search with single particle while MVMO^S starts the search with a set of particles. MVMO is extended to two parameters: the scaling factor f_s and variable incrementΔdparameter to enhance its mapping. Therefore, the global search ability of the MVMO^S is strengthened. The detail of MVMO^S algorithm is described in [38].

3.2 Calculation of power output for slack unit

Slack variable method is usually used for handling equality constraints in optimization problems where the value of variables is calculated from the others based on the equality constraints [27]. This method is used for calculation of the power output for a slack unit from the power outputs of the remaining units in the system based on the power balance constraint in Eq. (4). The power output of the slack unit is as follows:

$$P_s = P_D + P_L - \sum_{\substack{i=1 \\ i \ne s}}^{N} P_i \quad (15)$$

Equation (15) represents the calculation of slack unit s which is randomly selected from the available N units and the limit violations of the slack unit will be penalized in the fitness function in Eq. (18). The first unit of all test system is selected as slack unit in this paper.

3.3 Implementation of MVMO^S to ED

3.3.1 Initialization of algorithm

The parameters for MVMO^S have to be initialized including $iter_{max}$, *n_var*, *n_par*, *mode*, d_i, Δd_0^{ini}, $\Delta d_0^{\text{final}}$, *archive zize* (n), $f^*_{s\_ini}$, $f^*_{s\_\text{final}}$, *n_randomly*, *n_randomly_min*, *indep.runs*(m), D_{min}..

3.3.2 Normalization and de-normalization of variables

In the proposed MVMO^S method, each particle represents a solution. A set of particles is used for finding best solution for the problem. The initial optimization variables are normalized to the [0,1] bound as follows:

$$x\_normalized = rand(n\_par, n\_var) \tag{16}$$

The search space of the algorithm is always restricted inside the [0,1] range. However, the function evaluation is carried out in the original scales $[P_{i,min}, P_{i,max}]$. The denormalization of the optimization variables is carried as follows:

$$P_i = P_{i,min} + (P_{i,max} - P_{i,min}).x\_normalized(\iota, :) \tag{17}$$

This initial solution is further checked for POZ violation. If the violation is found, the repairing strategy in [2] is used to move the operating point to a feasible region. After that, the power output for the slack generator is calculated by using Eq. (15). The fitness function includes the objective function in Eq. (1), penalty terms for the slack unit if the generator capacity limits constraint in Eq. (6) is violated and the penalty terms for spinning reserve constraint in Eq. (9). The fitness function F_T to be minimized for the considered problem is calculated as follows:

$$\begin{aligned} F_T = & \sum_{i=1}^{N} F_i(P_i) + K_s.[(\max(0, P_s - P_{s,\max})) \\ & + (\max(0, P_{s,\min} - P_s))] \\ & + K_p.\max\left(0, S_R - \sum_{i=1}^{N} S_i\right) \end{aligned} \tag{18}$$

where K_s and K_p are the penalty factor for the slack unit and spinning reserve constraint, respectively.

MVMO^S utilizes swarm implementation to enhance the power of global searching of the classical MVMO by starting the search process with a set of particles, each having its own memory and represented by the corresponding archive and mapping function. At the beginning of the optimization process, each particle performs m steps independently to collect a set of reliable individual solutions. Then, the particles start to communicate and to exchange information. It is worthless when particles are very close to each other since this would entail redundancy. To avoid closeness between particles, the normalized distance of each particle's local best solution $x^{lbest,i}$ to the global best x^{gbest} is calculated by Eq. (19). This normalized distance is employed to attempt to reduce the swarm size provided. The i-th particle is discarded from the optimization process if the distance D_i is less than a certain user defined threshold D_{min} [38]. The information exchange between particles and swarm reduction are aimed at enhancing the global search ability while avoiding redundancy in the search process.

#	Fitness	x_1	x_1	... x_i...	x_N
1					
...					
n					
Mean	---	$\overline{x_1}$	$\overline{x_2}$	$\overline{x_i}$	$\overline{x_N}$
Variance	---	v_1	v_2	v_i	v_N

Fig. 3 The archive is used to store n-best population

$$D_i = \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(x_j^{gbest} - x_j^{lbest,i}\right)^2} \tag{19}$$

3.3.3 Solution archive

The best fitness and variables are stored in the archive table which is described as Fig. 3. The archive size (n) is taken to be a minimum of two. The table of best individuals is filled up progressively over iterations in a descending order of the fitness. When the table is filled with n members, an update is performed only if the fitness of the new population is better than those in the table.

Mean $\bar{x}_i$ and variance v_i are computed from the archive as follows [37]:

$$\bar{x}_i = \frac{1}{n}\sum_{j=1}^{n} x_i(j) \tag{20}$$

$$v_i = \frac{1}{n}\sum_{j=1}^{n} (x_i(j) - \bar{x}_i)^2 \tag{21}$$

Where j goes from 1 to n (archive size). At the beginning $\bar{x}_i$ corresponds with the initialized value of x_i and the variance v_i is set to 1.

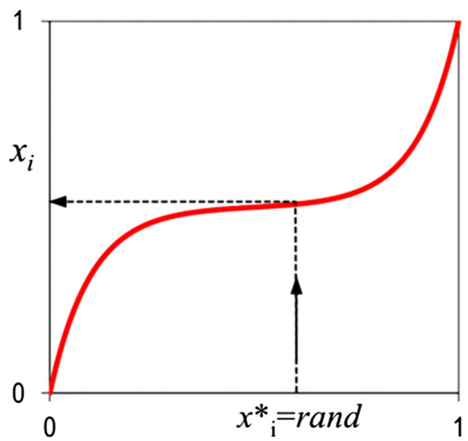


Fig. 4 Variable mapping

3.3.4 Offspring creation

The individual with the best fitness, f_{best}, and its corresponding optimization values, x_{best}, are stored in the memory of the parent population for that iteration. This parent is used for creation of the next offspring. Three common evolutionary operations in offspring creation are: selection, mutation and crossover operators.

Selection

Among N variables of the optimization problem, d variables are selected for mutation operation. There are four strategies which are described in [37] for selecting the variables.

Mutation

For each of the d selected dimension, mutation is used to assign a new value of that variable. Given a uniform random number $x_i^* \in [0,1]$, the transformation of x_i^* to x_i via mapping function is calculated in Eq. (22) and depicted as Fig. 4. The transformation mapping function, h, is calculated by the mean $\bar{x}$ and shape variables s_{i1} and s_{i2} as in Eq. (24) [37]:

$$x_i = h_x + (1 - h_1 + h_o).x_i^* - h_o \tag{22}$$

where h_x, h_1, h_0 are the outputs of transformation mapping function in Eq. (24) based on different inputs given by [37]:

$$h_x = h(x = x_i^*), h_o = h(x = 0), h_1 = h(x = 1) \tag{23}$$

$$h(\bar{x}_i, s_{i1}, s_{i2}, x) = \bar{x}_i.(1 - e^{-x.s_{i1}}) + (1 - \bar{x}_i).e^{-(1-x).s_{i2}} \tag{24}$$

where

$$s_i = -\ln(v_i).f_s \tag{25}$$

The scaling factor f_s in Eq. (25) is a MVMO parameter which can be used to change the shape of the function during iteration. In MVMO^S, this factor is extended due to the need for exploring the search space more globally at the beginning of the iterations. At the end of the iterations, the focus should be on exploitation procedures. In [38], the factor f_s is given as follows:

$$f_s = f_s^*. (1 + rand()) \tag{26}$$

where

$$f_s^* = f_{s\_\text{ini}}^* + \left(\frac{i}{i_{\text{final}}}\right)^2 (f_{s\_\text{final}}^* - f_{s\_\text{ini}}^*) \tag{27}$$

In Eq. (27), f_s^* denotes the smallest value of f_s and the variable i represents the iteration number. $f_{s\_ini}^*$ and $f_{s\_\text{final}}^*$ are the initial and final values of f_s^*, respectively. The recommended range of $f_{s\_ini}^*$ is from 0.9 to1.0, and range of $f_{s\_\text{final}}^*$ is from 1.0 to 3.0. When $f_{s\_\text{final}}^* = f_{s\_\text{ini}}^* = 1$, which means that the option for controlling the f_s factor is not used [38]. The shape variables s_{i1} and s_{i2} in Eq. (24) are determined by using the following algorithm[38]:

```
s_i1 = s_i2 = s_i
if s_i > 0 then
Δd = (1 + Δd_o) + 2 . Δd_o (rand() – 0.5)
  if s_i > d_i
    d_i = d_i . Δd
  else
    d_i = d_i / Δd
  end if
  if rand() ≥ 0.5 then
     s_i1 = s_i ; s_i2 = d_i
     else
     s_i1 = d_i ; s_i2 = s_i
     end if
 end if
```

The above procedure fully exploits the asymmetric characteristic of the mapping function by using different values for s_{i1} and s_{i2}.Δd_0 is calculated in Eq. (28), this factor is allowed to decrease from 0.4 to 0.01.

$$\Delta d_0 = \Delta d_0^{\text{ini}} + \left(\frac{i}{i_{\text{final}}}\right)^2 \left(\Delta d_0^{\text{final}} - \Delta d_0^{\text{ini}}\right). \tag{28}$$

Crossover

The crossover operation is done for the remaining unmutated dimensions where the genes of the parents are inherited. In other words, the values of these unmutated dimensions are clones of the parent. Here, the crossover is done by direct cloning of certain genes. In this way, the offsprings are created by combining the vector x_{best}, and vector of d mutated dimensions.

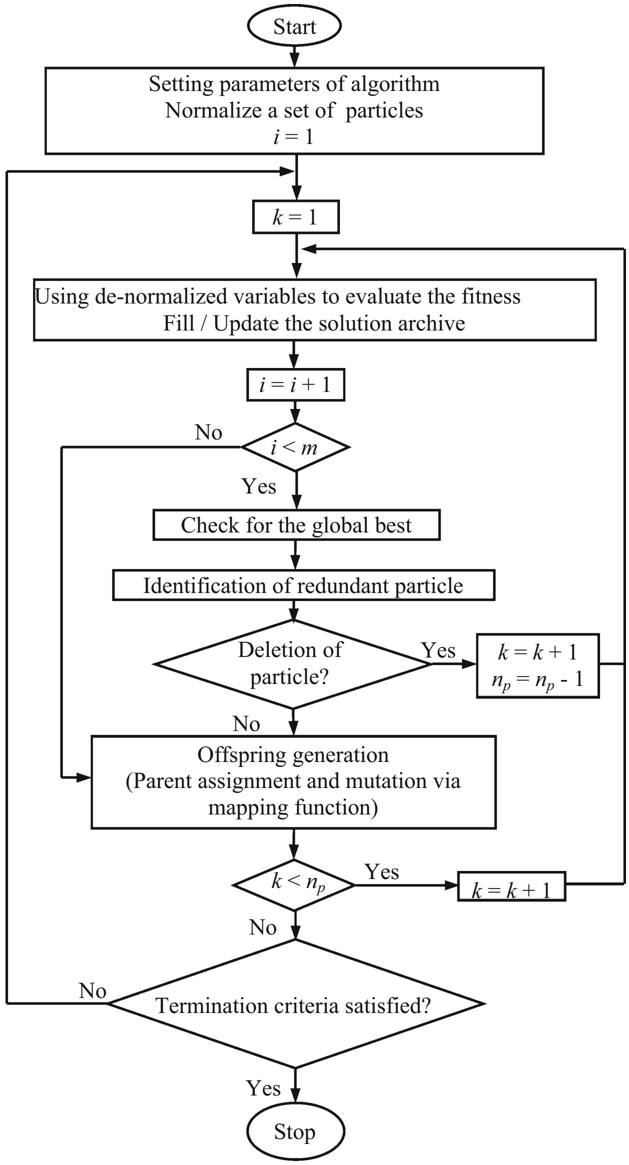


Fig. 5 The flowchart of MVMO^S

3.3.5 Termination criteria

The algorithm of the proposed MVMO^S is terminated when the maximum number of iterations $iter_{max}$ is reached.

3.3.6 Overall procedure

The flowchart of MVMO^S is depicted in Fig. 5 [36]:

The steps of procedure of MVMO^S for the ED problem are described as follows:

Step 1 Setting the parameters for MVMO^S including $iter_{max}$, n_var, n_par, $mode$, d_i, Δd_0^{ini}, $\Delta d_0^{\text{final}}$, archive zize, $f_{s\_ini}^*$, $f_{s\_\text{final}}^*$, n_randomly, n_randomly_min, indep.runs(m),D_{min} Set $i = 1$, i denotes the function evaluation

Step 2 Normalizing initial variables to the range [0,1] (i.e. swarm of particles) by using Eq. (16).

Step 3 Set $k = 1$, k denotes particle counters.

Step 4 De-normalizing variables using Eq. (17), check for POZ violation and repair,calculate power output for the slack generator using Eq. (15), evaluate fitness function in Eq. (18), store f_{best} and x_{best} in archive.

Step 5 Increase $i = i + 1$. If $i < m$(independent steps), go to Step 6. Otherwise, go to Step 7.

Step 6 Check the particles for the global best, collect a set of reliable individual solutions. The i-th particle is discarded from the optimization process if the distance D_i is less than a certain user defined threshold D_{min}. If the particle is delected, increase $k = k + 1$, $n_p = n_p - 1$ and go to step 4. Otherwise, go to Step 7.

Step 7 Create offspring generation through three evolutionary operators: selection, mutation and crossover.

Step 8 if $k < n_p$, increase $k = k + 1$ and go to step 4. Otherwise, go to step 9.

Step 9 Check termination criteria. If stoping criteria is satisfied, stop. Otherwise, go to step 3. The algorithm of the proposed MVMO^S is terminated when the maximum number of iterations $iter_{max}$ is reached.

4 Numerical results

The proposed MVMO^S is mplemented to different convex and nonconvex ED problems. The MVMO^S is more effective than the original MVMO in term of the global search ability [38]. For this confirmation, the single particle MVMO is also implemented to the ED problems by set the number of particles n_p to 1. Different systems corresponding to the formulated problems are used for testing the proposed method. The implementation of the MVMO^S is coded in Matlab R2013a platform and executed for 50 independent trials on a core i5-3470 CPU 3.2 GHz PC with 4GB of RAM for each case.

4.1 Selection of parameters

Since different parameters of the proposed method have effects on the performance of MVMO^S. Hence, it is important to determine a set of optimal parameters of the proposed methods for dealing with ED problems. For each problem, the selection of parameters is carried out by varying only one parameter at a time while fixing the others. The parameter is first fixed at the low value and then increased. The obtained result after one run is compared to the previous one. If the obtained result after

Table 1 Parameter setting of MVMO^S

Parameters	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6			Case 7
$iter_{max}$	10,000	4000	40,000	3000	30,000	50,000	70,000	90,000	1,500,000
n_var (generators)	20	3	26	6	15	30	60	90	140
n_p	5	20	10	5	20	5	5	5	5
archive size	4	4	4	4	4	4	4	4	4
mode	4	4	4	4	4	4	4	4	4
indep.runs (m)	200	200	200	200	800	800	800	800	800
n_randomly	7	2	12	3	7	10	25	30	20
n_randomly_min	6	2	10	2	6	5	20	25	10
$f^*_{s\_ini}$	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
$f^*_{s\_\text{final}}$	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5
d_i	1	1	1	1	1	1	1	1	1
Δd_0^{ini}	0.25	025	0.25	0.25	0.25	0.25	0.25	0.25	0.25
$\Delta d_0^{\text{final}}$	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
D_{min}	0	0	0	0	0	0	0	0	0

one run is considerably better than the previous one, their obtained value is chosen as the proper value. Otherwise, their value will be increased. Multiple runs are carried out to choose the suitable set of parameters. By experiments, the set of optimal parameters for each test system are shown in Table 1. The typical parameters are selected as follows:

- $iter_{max}$: maximum number of iterations depend on the scale of test systems and complexity of the problem. The maximum number of iterations is selected in the range from 2000 to 150000 iterations.
- *n_var*: number of optimization variables. This parameter denotes the number of generating units in system.
- *n_par*: number of particles is varied from 5, 10, 20, 30, 40 and 50, respectively. The number of particles is chosen by experiments for each case.
- *mode* : There are four variable selection strategy for off-spring creation . Afer all simulations, stragy 4 (*mode* = 4) is suporior to the other stragy.
- Δd_0^{ini}, $\Delta d_0^{\text{final}}$: The range of Δd_0 in Eq. (28) is [0.01 – 0.4]. By experiments, Δd_0^{ini} and $\Delta d_0^{\text{final}}$ is set to 0.25 and 0.02, respectively for all cases.
- $f^*_{s\_ini}$, $f^*_{s\_\text{final}}$: The range of values of $f^*_{s\_ini}$ is from 0.9 to 1.0 and for values of $f^*_{s\_\text{final}}$ is from 1.0 to 3.0. For all cases, $f^*_{s\_ini}$ is set to 0.95 in the range [0.9, 1.0] and $f^*_{s\_\text{final}}$ is set to 2.5 in the in the range [1.0, 3.0].
- *indep.runs(m)*: The maximum number of independent runs can be selected in the range from 200 to 800.
- D_{min} is set to 0 for all cases.

The penalty factor K_s for the slack unit and the penalty factor K_p spinning reserve constraint are large enough and set to 10^3 for all systems.

4.2 Convex ED problems

The fuel cost function is in turn as the quadratic and cubic form for the convex ED problems.

4.2.1 Case 1: 20-unit test system with quadratic cost function

This system supplies to a total load demand of 2500 MW. The input data of the 20-unit system are from [10] including fuel cost coefficients, generator capacity limits and B loss matrix. The results obtained by the proposed MVMO^S are compared to those from lamda-iteration [10], HNN [10], BBO [21] and MVMO in Table 2. The comparison shows that the total cost and power loss obtained by the proposed MVMO^S are very close to the other methods in Table 2. It is noted that the MVMO^S achieves the optimal solution with a high probability (the standard deviation is 0 %) while the solutions of lamda-iteration and HNN mismatch with the total power demand and power loss.

4.2.2 Case 2: 3-unit test system with cubic cost function

The data information for 3-generating units with cubic cost function are given in [3]. The total power load demand for this system is 1400 MW. The transmission power loss is considered in this case. The B loss matrix for the calculation of power loss is referred from [1]. The obtained results by the MVMO and MVMO^S are presented in Table 3 along with the solutions of SA1, SA2 [18] and IGA_MU [13]. The total cost and power loss obtained by the proposed MVMO^S are very close to the other methods except SA1. This system has two local optimum (about 6639 and 6642) [13], and SA1 are likely to get stuck in a local solution.

Table 2 Results and comparisons for 20-unit system with quadratic fuel cost function

Method Unit	Lamda iteration [10] P_i (MW)	HNN [10] P_i (MW)	BBO [21] P_i (MW)	MVMO P_i (MW)	$MVMO^S$ P_i (MW)
1	512.7805	512.7804	513.0892	512.7921	512.8032
2	169.1033	169.1035	173.3533	169.0396	169.0356
3	126.8898	126.8897	126.9231	126.8966	126.9038
4	102.8657	102.8656	103.3292	102.8659	102.8794
5	113.6836	113.6836	113.7741	113.6836	113.6801
6	73.5710	73.5709	73.06694	73.5790	73.5880
7	115.2878	115.2876	114.9843	115.2981	115.3130
8	116.3994	116.3994	116.4238	116.4029	116.4186
9	100.4062	100.4063	100.6948	100.4000	100.3979
10	106.0267	106.0267	99.99979	106.0532	106.0461
11	150.2394	150.2395	148.9770	150.2495	150.2079
12	292.7648	292.7647	294.0207	292.7497	292.7671
13	119.1154	119.1155	119.5754	119.1091	119.1058
14	30.8340	30.8342	30.54786	30.8429	30.8156
15	115.8057	115.8056	116.4546	115.8041	115.8156
16	36.2545	36.2545	36.22787	36.2551	36.2531
17	66.8590	66.8590	66.85943	66.8609	66.8630
18	87.9720	87.9720	88.54701	87.9606	87.9716
19	100.8033	100.8033	100.9802	100.8077	100.7986
20	54.3050	54.3050	54.2725	54.3146	54.2975
Total power (MW)	2591.9671	2591.9670	2592.1011	2591.9653	2591.9615
Power loss (MW)	91.9670	91.9669	92.1011	91.9653	91.9615
Power mismatch (MW)	−0.000187	0.000021	0.0	0.0	0.0
Min cost ($/h)	62,456.9391	62,456.6341	62,456.7926	62,456.6331	62,456.6331
Average cost ($/h)	–	–	62,456.7928	62,456.6331	62,456.6331
Max cost ($/h)	–	–	62,456.7935	62,456.6331	62,456.6331
Standard deviation($/h)	–	–	–	0.0	0.0
Average CPU time (s)	0.033757	0.006355	0.29282	6.334	6.365

Table 3 Results and comparisons for 3-unit system with cubic fuel cost function by MVMO and $MVMO^S$

Method Unit	SA1 [18] P_i (MW)	SA2 [18] P_i (MW)	IGA_MU[13] P_i (MW)	MVMO P_i (MW)	$MVMO^S$ P_i (MW)
1	359.546	376.123	365.4085	365.3941	365.5376
2	406.734	100.052	100.000	100.0000	100.0000
3	677.152	986.273	997.3436	997.3585	997.2108
Total power (MW)	1443.434	1462.448	1462.7521	1462.7526	1462.3517
Power loss (MW)	43.434	62.448	62.7521	62.7526	62.7484
Min cost ($/h)	6642.657	6639.504	6639.1849	6639.1853	6639.1853
Average cost ($/h)	–	–	–	6642.4732	6640.1013
Max cost ($/h)	–	–	–	6642.6830	6642.6830
Standard deviation ($/h)	–	–	–	0.8391	1.5308
Average CPU time (s)	–	–	–	1.758	1.823

Table 4 Results and comparisons for 26-unit system with cubic fuel cost function by MVMO and MVMO^S

Method Unit	GA [6] P_i (MW)	PSO [6] P_i (MW)	MVMO P_i (MW)	MVMO^S P_i (MW)
1	2.40	2.40	2.3938	2.3937
2	2.40	2.40	2.400	2.400
3	2.40	2.40	2.400	2.400
4	2.40	2.40	2.400	2.400
5	2.40	2.40	2.400	2.400
6	4.00	4.00	4.000	4.000
7	4.00	4.00	4.000	4.000
8	4.00	4.00	4.000	4.000
9	4.00	4.00	4.000	4.000
10	15.20	15.20	16.5434	15.2000
11	15.20	15.20	15.2000	15.2002
12	15.20	15.20	15.2058	15.2006
13	15.20	15.20	15.3700	18.3375
14	25.00	25.00	25.0000	25.0000
15	25.00	25.00	25.0000	25.0000
16	25.00	25.00	25.0000	25.0000
17	129.71	129.69	143.1421	131.2265
18	124.71	124.69	124.7450	110.5722
19	120.42	120.40	134.7322	123.5438
20	116.72	116.70	134.3684	114.8961
21	68.95	68.95	68.9500	68.9500
22	68.95	68.95	68.9500	68.9500
23	68.95	68.95	68.9500	68.9500
24	337.76	337.85	290.8493	345.9793
25	400.00	400.00	400.0000	400.0000
26	400.00	400.00	400.0000	400.0000
Total power (MW)	2000.00	2000.00	2000.00	2000.00
Min cost ($/h)	27,671.2441	27,671.2276	27,261.8836	27,252.1262
Average cost ($/h)	–	–	27,342.9673	27,292.5052
Max cost ($/h)	–	–	27,741.6415	27,345.2012
Std. deviation($/h)	–	–	76.5519	22.7301
Average CPU time (s)	–	–	21.303	21.565

4.2.3 Case 3: 26-unit test system with cubic cost function

The data of the test system including 26 thermal generating units with cubic fuel cost function can be found in [8]. The system load demand for this case is 2000 MW neglecting transmission power loss. The obtained results by the MVMO^S are compared to those from GA, PSO [6] and MVMO as given in Table 4. The proposed MVMO^S provides the total cost less than GA, PSO and MVMO.

4.3 Nonconvex ED problems

The proposed method is tested on different nonconvex problem including VPE and POZ characteristics. In order to demonstrate the efficiency of the proposed approach, it is also tested on large-scale systems.

4.3.1 Case 4: ED problem with valve point effects

The proposed MVMO^S are applied to IEEE test systems including 30 bus and 6 thermal generating units with VPE. The fuel cost coefficients, generator capacity limits and B loss matrix for this test systems are refered from [33]. The test system here is for 283.4 MW load demand. Table 5 shows the results obtained by the MVMO and MVMO^S. The comparisons of fuel costs obtained by the proposed MVMO^S and other methods are listed in Table 6. The proposed MVMO^S can obtain better fuel costs than GA, GA-APSO, NSOA [33], DE [23] and λ-logic [40], and close to MVMO.

Table 5 Results for 6-unit system with VPE by MVMO and MVMO^S

Method Unit	MVMO *Pi* (MW)	MVMO^S *Pi* (MW)
1	199.5997	199.5997
2	20.0000	20.0000
3	18.7881	21.1230
4	12.9487	10.4509
5	30.6650	30.1236
6	12.0010	12.5795
Total power (MW)	294.0026	293.8766
Power loss (MW)	10.6026	10.4766
Min cost ($/h)	927.7827	926.9800
Average cost ($/h)	949.2403	938.2780
Max cost ($/h)	998.7118	943.9712
Standard deviation ($/h)	19.8170	5.2368
Average CPU time (s)	1.649	1.747

4.3.2 ED problem with prohibited operating zones

(a) *Case 5: 15-unit system with POZ* The proposed MVMO^S is testes on 15-unit test system [17]. This system consists of 15 thermal generating units, in which four units have POZ. The system supplies to a power load demand of 2650 MW with a system spinning reserve requirement of 200 MW. Power loss and ramp rate constraint are neglected.

In order to show the efficiency of the proposed method, the MVMO^S is also tested on a system with some modified units' data of above 15-unit system. The modifed data system is found in [14]. Table 7 shows the solutions of the MVMO and MVMO^S for the original and modified cases.

For the orginal case, the fuel cost of the MVMO^S are compared to those of $\lambda - \delta$ iterative method [41], IHNN [11], EHNN [12], EP [42], FCEPA [42], QEA [17], IQEA [17] and MVMO. From the Table 8, the fuel cost of MVMO and MVMO^S are less than IHNN and slightly lower than other methods except EHNN. Although the fuel cost from the EHNN is slightly lower than that of the MVMO and MVMO^S, power balance constraint in the EHNN is not satisfied, where 0.8 MW is not dispatched yet.

For the modified case, the fuel cost obtained by the MVMO^S are compared to that of SGA [14], DCGA [14], ETQ [35], CEP [16], FEP [16], IFEP [16] and MVMO. As shown in the Table 9, the cost from the proposed method is slightly lower than that from SGA and DCGA in [14], and close to that from the other methods.

The 15-unit test system above includes the ramp rate constraint and neglects spinning reserve required. The power loss is considered for this case. The power load demand for this system is 2630 MW. The fuel cost coefficients, generator capacity limits, prohibited operating zones and B loss matrix are from [24]. Table 10 presents the solutions obtained by the MVMO and MVMO^S. The solution comparison is shown in Table 11, where the fuel costs of the MVMO^S are slightly lower than SA-PSO [27], PGPSO [2], ABC [19], CSA [41] and MVMO, and less than the other methods.

(b) Case 6: Large-scale systems with prohibited operating zones In order to demonstrate the applicable capability to large-scale systems, the proposed MVMO^S is test on 30-unit system, 60-unit system and 90-unit system with POZ. These large-scale systems are built from the basic 15-unit system [17] which supplies to the power load demand of 2650 MW with a required spinning reserve of 200M. The load demand is proportionally adjusted to the size of each large-scale system. Table 12 shows the fuel cost and CPU times obtained by the MVMO and MVMO^S.

The MVMO^S are compared to the convention GA (CGA), improved GA with multiplier updating method (IGAMUM) [15] and MVMO in term of average total costs and CPU times in Table 13. The comparison shows that the MVMO^S can obtain better solution quality than other methods.

4.3.3 Case 7: ED problem with both valve point effects and prohibited operating zones

The test system here is also a large-scale system including 140 generating units which supplies to the power load demand of 49342 MW neglecting transmission power loss. In order to show the efficiency of the proposed MVMO^S,

Table 6 Comparisons for 6-unit system with VPE

Method	Min cost ($/h)	Average cost ($/h)	Max cost ($/h)	Power loss (MW)	CPU (s)
GA [33]	996.0369	–	1117.1285	8.7060	0.578
GA-APO [33]	996.0369	–	1101.491	10.7563	0.156
NSOA [33]	984.9365	–	992.4815	10.4395	0.015
DE [23]	963.0010	–	–	9.3425	0.6558
λ-logic [40]	961.5654	–	–	6.4195	0.064
MVMO	927.7827	949.2403	998.7118	10.6026	1.649
MVMO^S	926.9800	938.2780	943.9712	10.4766	1.747

Table 7 Results for 15-unit system with POZ by MVMO and MVMO^S

Method	Original		Modified units' data	
Unit	MVMO P_i (MW)	MVMO^S P_i (MW)	MVMO P_i (MW)	MVMO^S P_i (MW)
1	450.0000	450.0000	455.0000	455.0000
2	450.0000	450.0000	455.0000	455.0000
3	130.0000	130.0000	130.0000	130.0000
4	130.0000	130.0000	130.0000	130.0000
5	335.0000	335.0000	259.6383	259.0906
6	455.0000	455.0000	459.9992	460.0000
7	465.0000	465.0000	465.0000	465.0000
8	60.0000	60.0000	60.0000	60.0000
9	25.0000	25.0000	25.0000	25.0000
10	20.0000	20.0000	26.5966	20.0000
11	20.0000	20.0000	64.3295	60.6415
12	55.0000	55.0000	64.4364	75.2679
13	25.0000	25.0000	25.0000	25.0000
14	15.0000	15.0000	15.0000	15.0000
15	15.0000	15.0000	15.0000	15.0000
Total power (MW)	2650	2650	2650	2650
Min cost ($/h)	32,544.9704	32,544.9704	32,506.5807	32,506.2863
Average cost ($/h)	32,550.8295	32,545.9417	32,509.0947	32,507.7312
Max cost ($/h)	32,561.5595	32,547.9949	32,524.4595	32,508.5308
Std. deviation ($/h)	5.5653	1.2882	2.7784	0.4414
Average CPU time (s)	8.712	9.445	8.617	9.310

Table 8 Comparisons of best cost for 15-unit system with POZ for original case

Method	Total cost ($/h)	Total power (MW)
$\lambda - \delta$ iterative method [41]	32,549.80	2650
IHNN [11]	32,568.00	2650
EHNN [12]	32,536.90	2649.2
EP [42]	32,545.20	2650
FCEPA [42]	32,544.97	2650
QEA [17]	32,548.48	2650
IQEA [17]	32,544.97	2650
MVMO	32,544.97	2650
MVMO^S	32,544.97	2650

Table 9 Comparisons of best cost and CPU time for 15-unit system with modified units' data neglecting ramp rate constraint and power loss

Method	Total cost ($/h)	Total power (MW)	CPU (s)
SGA [14]	32,517.00	2649.900	454.3
DCGA [14]	32,515.00	2649.900	398.5
ETQ [35]	32,507.50	2650.000	15.8
CEP [16]	32,507.55	2649.997	3.746
FEP [16]	32,507.55	2649.995	2.769
IFEP [16]	32,507.46	2649.994	3.318
MVMO	32,506.85	2650.000	8.617
MVMO^S	32,506.29	2650.000	9.310

both nonconvex characteristics including VPE and POZ are consider for this case. The data of this system can be found in [25] where 12 units have VPE and 4 other units have POZ. A comparison of fuel costs and CPU times from the MVMO^S and PSO methods [25] is shown in Table 14. The comparison shows that the proposed MVMO^S dominates PSO methods including conventional PSO with the constraint treatment strategy (CTPSO), PSO with chaotic sequences (CSPSO), PSO with crossover operation (COPSO) and PSO with both chaotic sequences and crossover operation (CCPSO) in term of optimal solution quality. Note that the maximum fuel cost obtained by MVMO^S is even better than the minimum fuel costs obtained by PSO methods. The optimal dispatch solution for 140-unit system is shown in Table 15.

5 Discussion

The proposed MVMO^S has been implemented to different ED problems including convex and nonconvex characteristics. The issues from the numerical results are given as follows:

Table 10 Results for 15-unit system with POZ including ramp rate constraint and power loss by MVMO and $MVMO^S$

Method Unit	MVMO *Pi* (MW)	$MVMO^S$ *Pi* (MW)
1	455.0000	455.0000
2	380.000	380.0000
3	130.0000	130.0000
4	130.0000	130.0000
5	170.0000	170.0000
6	460.0000	460.0000
7	430.0000	430.0000
8	86.0128	74.5596
9	45.3576	56.1155
10	159.3699	159.9994
11	79.9999	79.9997
12	80.0000	80.0000
13	25.0000	25.0000
14	15.0000	15.0000
15	15.0000	15.0000
Total power(MW)	2660.7402	2660.6742
Power loss (MW)	30.7402	30.6742
Min cost ($/h)	32,705.0250	32,704.4721
Average cost ($/h)	32,711.5821	32,706.6475
Max cost ($/h)	32,725.4457	32,709.3152
Standard deviation ($/h)	5.4956	1.1143
Average CPU time (s)	14.242	14.820

Table 11 Comparisons for 15-unit system with POZ

Method	Min cost ($/h)	Average cost ($/h)	Max cost ($/h)	Power loss (MW)	CPU (s)
GA [24]	33,113.00	33,228.00	33,337.00	38.2782	49.31
PSO [24]	32,858.00	33,039.00	33,331.00	32.4306	26.59
AIS [20]	32,854.00	32,873.25	32,892.00	32.4075	–
CPSO1 [30]	32,835.00	33,021.00	–	32.1302	–
CPSO2 [30]	32,834.00	33,021.00	–	32.1303	–
SOH-PSO [26]	32,751.00	32,878.00	32,945.00	32.2800	0.0936
PSO-MSAF [43]	32,713.09	32,759.64	32,798.2	30.4900	19.15
SA-PSO [27]	32,708.00	32,732.00	32,789.00	30.9080	12.79
PGPSO [2]	32,705.75	32,716.84	32,726.08	30.6644	1.631
ABC [19]	32,707.85	32,707.95	32,708.27	30.9591	11.02
CSA [41]	32,706.66	–	–	30.8577	2.226
MVMO	32,705.05	32,711.58	32,725.45	30.7402	14.242
$MVMO^S$	32,704.47	32,706.65	32,709.32	30.6742	14.820

5.1 Solution quality

From Tables 2, 3, 4, 6, 8, 9, 11, 13 and 14, the proposed $MVMO^S$ provides better total fuel cost than other reported methods. Especially for large-scale system, as seen from Table 14, the proposed $MVMO^S$ dominates PSO methods in term of all minimum, average and maximum total fuel cost. Moreover, the power outputs obtained by $MVMO^S$ are always between the minimum and maximum generator capacity limits and the total power output of generating units always equals to the power load demand. It is indicated that the equality and inequality constraints always satisfy. Consequently, the $MVMO^S$can obtain very good solution quality for ED problems.

Table 12 Fuel costs and CPU times for large-scale systems with POZ

Methods	MVMO			$MVMO^S$		
Number of units	30	60	90	30	60	90
Min cost ($/h)	65,086.3370	130,170.8046	195,258.6600	65,086.2051	130,170.7797	195,258.4951
Average cost ($/h)	65,090.2023	130,175.0956	195,263.5962	65,089.2153	130,175.0130	195,263.5819
Max cost ($/h)	65,096.4193	130,184.1467	195,273.9091	65,090.7489	130,181.1126	195,269.1161
Standard deviation ($/h)	1.6844	3.1922	3.2920	1.4061	2.9197	3.1618
Average CPU time (s)	17.051	30.030	41.574	18.096	31.325	43.633

Table 13 Comparison of average total costs and CPU times for large-scale systems with POZ

Method	No. of units	Average cost ($)	CPU time (s)
CGA [15]	30	65,784.740	275.73
	60	131,992.310	563.81
	90	198,831.690	940.93
IGAMUM [15]	30	65,089.954	79.80
	60	130,180.030	162.58
	90	195,274.060	255.45
MVMO	30	65,090.2023	17.051
	60	130,175.0956	30.030
	90	195,263.5962	41.574
$MVMO^S$	30	65,089.2153	18.096
	60	130,175.0130	31.325
	90	195,263.5819	43.633

Table 14 Comparisons of fuel costs for 140-unit system with both VPE and POZ

Method	Min cost ($/h)	Average cost ($/h)	Max cost ($/h)	Standard deviation($/h)	CPU (s)
CTPSO [25]	1,657,962.73	1,657,964.06	1,658,002.79	7.3150	100
CSPSO [25]	1,657,962.73	1,657,962.74	1,657,962.85	0.0235	99
COPSO [25]	1,657,962.73	1,657,962.73	1,657,962.73	0.0002	150
CCPSO [25]	1,657,962.73	1,657,962.73	1,657,962.73	0.0000	150
MVMO	1,655,787.4348	1,656,002.8755	1,656,280.0410	122.8641	57.908
$MVMO^S$	1,655,758.0755	1,655,998.2312	1,656,268.6523	127.0692	64.460

5.2 Computational efficiency

In this paper, both the total cost and computational time are used for result comparison. However, it is difficult for the computational time comparison among the methods for optimization problems due to different computer processors and programming languages used. Therefore, the key factor for result comparison is usually the objective function rather than the computational time. The proposed $MVMO^S$ has been tested on large-scale systems and the obtained results including total cost and computational time have compared to those from many other methods. The comparison has indicated that the proposed method can obtain better solution quality than other methods. Although the proposed method is not as fast as some methods, it is also faster than many other methods. Moreover, the computational time from the proposed method is not too long compared to the faster methods. In fact, the large-scale and complex optimization problems are always a challenge for solution methods in both global optimal solution and computational time. In future, the proposed method can be further improved for efficiently dealing with different large-scale and complex problems.

5.3 Convergence characteristic

Figures 6, 7, 8, 9 and 10 depict the convergence characteristic of the MVMO and $MVMO^S$ for case 1 (ED with quadratic fuel cost function), case 3 (ED with cubic fuel cost function), case 4 (ED with VPE), case 5 (ED with POZ) and case 7 (ED with both VPE and POZ), respectively. All the figures

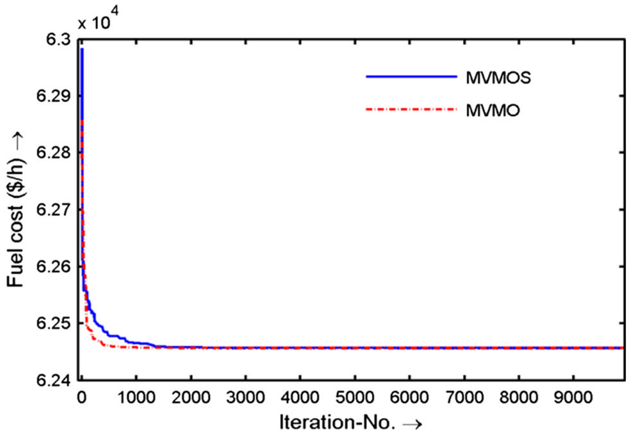


Fig. 6 Convergence property of MVMO and MVMO^S for case 1 (ED with qudratic fuel cost function)

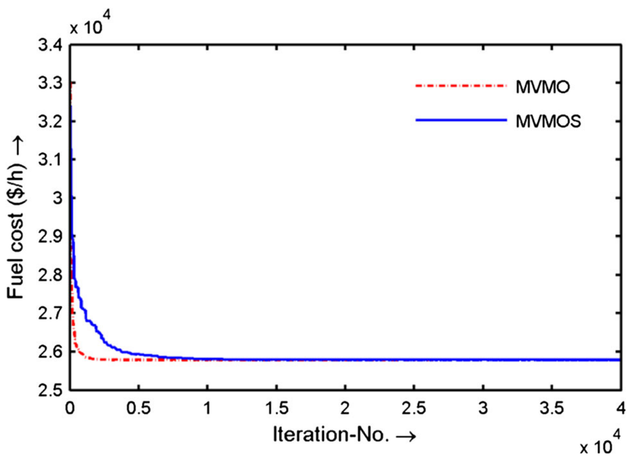


Fig. 7 Convergence property of MVMO and MVMO^S for case 3 (ED with cubic fuel cost function)

shows that both MVMO and MVMO^S provide the solution without premature convergence or trapping in local optima. It indicates their search ability can balance between exploration and exploitation. The MVMO converges faster than the MVMO^S. This is because the MVMO starts the search process with single particle while MVMO^S starts the search process with a set of particles. It makes the MVMO^S take more time than the original MVMO to converge. However, from all numerical results, the MVMO^S achieves better solution than MVMO. It is confirmed that the global search ability of the MVMO^S is better than the MVMO.

5.4 Robustness analysis

The MVMO and the MVMO^S are run 50 independent trials. The minimum cost, maximum cost, average cost and standard deviation obtained by the MVMO and MVMO^S to evaluate the robustness characteristic of the proposed method for ED problems. For case 1, the proposed MVMO^S achieves the

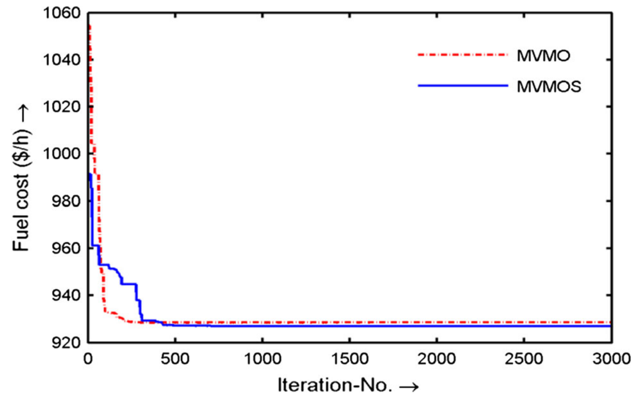


Fig. 8 Convergence property of MVMO and MVMO^S for case 4 (ED with VPE)

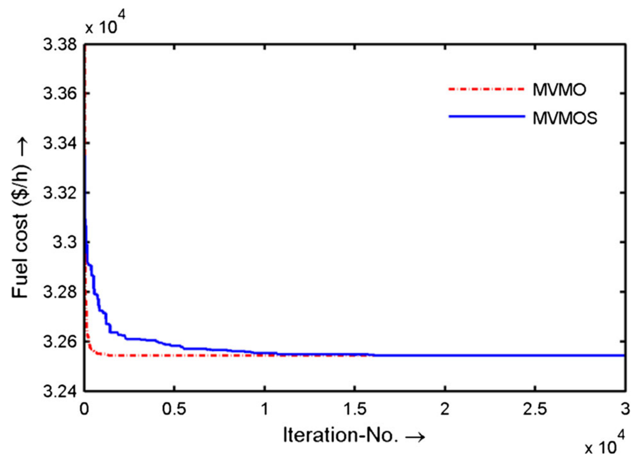


Fig. 9 Convergence property of MVMO and MVMO^S for case 5 (original case) ED with POZ

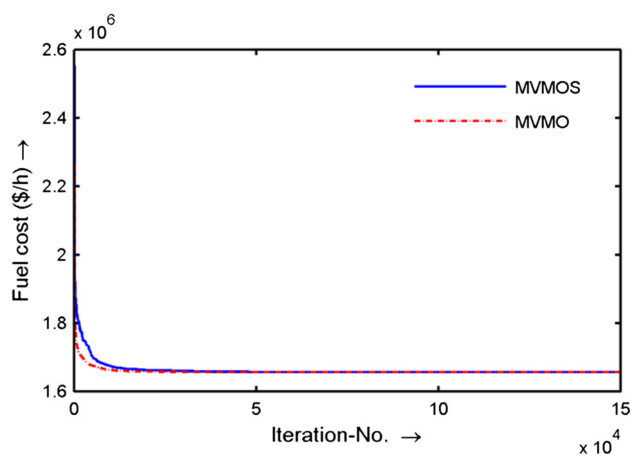
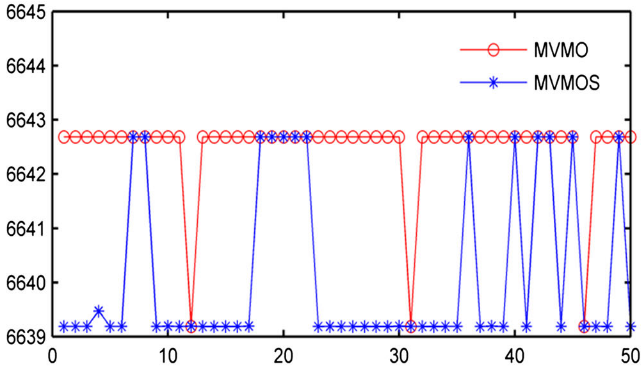


Fig. 10 Convergence property of MVMO and MVMO^S for case 7 (ED with both VPE and POZ)

optimal solution with a high probability (the standard deviation is 0 %). As observed from Tables 1, 6, 11 and 13, the proposed MVMO^S robust than most of the other methods in literature. Figures 11, 12, 13 and 14 show the distribution of



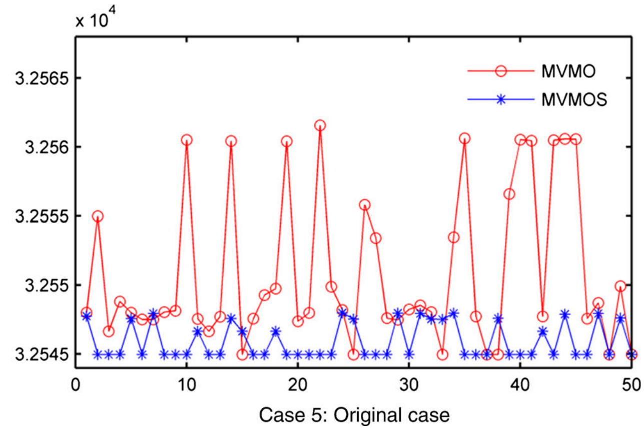
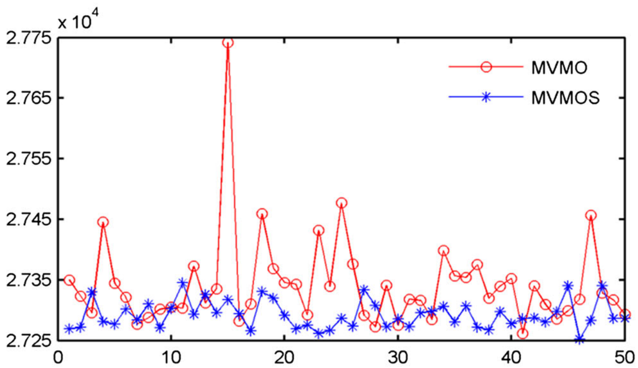


Fig. 11 Distribution of fuel cost of the MVMO and MVMO^S for case 2 (3 units with cubic fuel cost function)



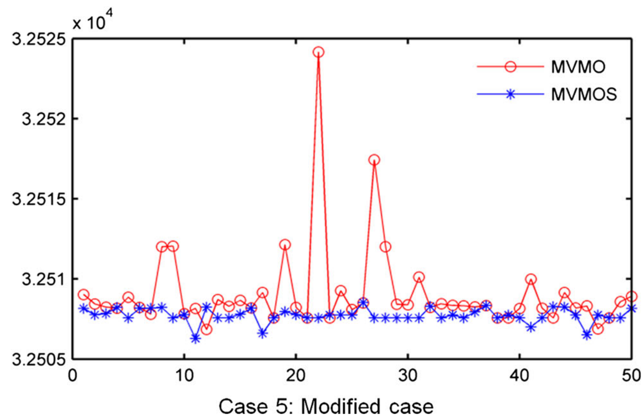


Fig. 12 Distribution of fuel cost of the MVMO and MVMO^S for case 3 (26 units with cubic fuel cost function)

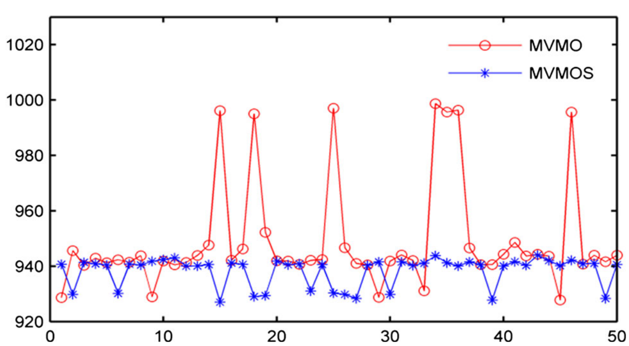


Fig. 13 Distribution of fuel cost of the MVMO and MVMO^S for case 4 (6 units with VPE)

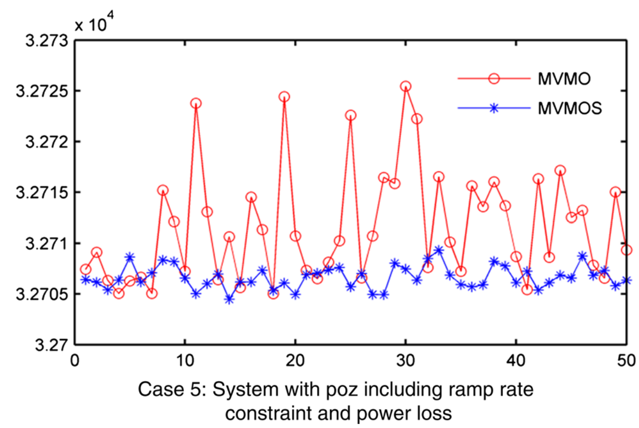


Fig. 14 Distribution of fuel cost of the MVMO and MVMO^S for case 5 (15 units with POZ). **a** Case 5: original case. **b** Case 5: modified case. **c** Case 5: system with poz including ramp rate constraint and power loss

fuel cost of the MVMO and MVMO^S of 50 trials for case 2, 3, 4 and case 5, respectively. All figures show that the MVMO^S is more robust than the the MVMO.

6 Conclusion

In this paper, the proposed MVMO^S has been effectively implemented for solving both convex and nonconvex ED problems. For the convex ED problem, the fuel cost function is in turn as the quadratic and cubic form. For the nonconvex ED problem, the nonconvex elements and nonlinearities in objective function and constraints are considered such as valve point effects, prohibited operating zones, ramp rate limits and spinning reserve. The proposed method keeps the concept of the conventional MVMO and starts search process with a set of particles to improve its global search ability and solution quality for optimization problems. The proposed MVMO^S has merits of easy implementation, good solutions, robust algorithm and applicable to large-scale systems. The numerical results have shown that the proposed MVMO^S has better performance than the other optimization techniques available in the literature in terms of global solution and robustness. Therefore, the proposed MVMO^S could be a favorable method for solving the ED problems in power systems.

Acknowledgements This research work is sponsored by GA scheme of Universiti Teknologi PETRONAS.

Appendix

Table 15 Power output of 140-unit system with both VPE and POZ by MVMO^S

Unit	*Pi* (MW)	Unit	*Pi* (MW)	Unit	*Pi* (MW)
1	119.0000	47	250.0000	94	175.0000
2	164.0000	48	250.0000	95	175.0000
3	190.0000	49	250.0000	96	188.1755
4	190.0000	50	250.0000	98	175.0000
5	186.7611	51	177.8151	99	575.4000
6	190.0000	52	183.4212	100	547.5000
7	490.0000	53	198.0368	101	836.8000
8	490.0000	54	165.0000	102	837.5000
9	496.0000	55	180.0000	103	682.0000
10	495.9964	56	180.0000	104	720.0000
12	496.0000	57	103.0000	105	718.0000
13	496.0000	58	198.0000	106	720.0000
14	506.0000	59	312.0000	107	964.0000
15	509.0000	60	309.0610	108	958.0000
16	505.9985	61	163.0000	109	947.9000
17	505.0000	62	95.0000	110	934.0000
18	506.0000	63	511.0000	111	935.0000
19	506.0000	64	503.7111	112	876.5000
20	505.0000	65	490.0000	113	880.9000
21	505.0000	66	260.1385	114	873.7000
22	505.0000	67	490.0000	115	877.4000
23	504.9605	68	490.0000	116	871.7000
24	505.0000	69	130.0000	117	864.8000
25	505.0000	70	282.5584	118	882.0000
26	537.0000	71	158.3615	119	94.0000
27	537.0000	72	358.8404	120	94.0000
28	549.0000	73	195.0000	121	94.0000

Table 15 continued

Unit	*Pi* (MW)	Unit	*Pi* (MW)	Unit	*Pi* (MW)
29	549.0000	74	189.2363	122	244.0000
30	501.0000	75	285.5849	123	244.0000
31	499.0000	76	260.9699	124	244.0000
32	506.0000	77	295.9341	125	95.0000
33	506.0000	78	332.3602	126	95.0000
34	505.9815	79	531.0000	127	116.0000
35	506.0000	80	531.0000	128	175.0000
36	500.0000	81	541.9981	129	2.0000
37	500.0000	82	56.0000	130	4.0000
38	241.0000	83	115.0000	131	15.0000
39	241.0000	84	115.0000	132	9.0000
40	774.0000	85	115.0000	133	12.0000
41	768.9993	86	207.0000	134	10.0000
42	3.0000	87	207.0000	135	112.0000
43	3.0000	88	250.0000	136	4.0000
44	250.0000	89	250.0000	137	5.0000
45	250.0000	90	250.0000	138	5.0000
46	250.0000	91	250.0000	139	50.0000
45	250.0000	92	177.8151	140	5.0000
46	119.0000	93	183.4212		

References

1. Wollenberg B, Wood A (1996) Power generation, operation and control. Wiley, New York
2. Dieu VN, Schegner P, Ongsakul W (2013) Pseudo-gradient based particle swarm optimization method for nonconvex economic dispatch. In: Power, control and optimization. Springer, New York, pp 1–27
3. Liang Z-X, Glover JD (1992) A zoom feature for a dynamic programming solution to economic dispatch including transmission losses. IEEE Trans Power Syst 7:544–550
4. Jiang A, Ertem S, Subir S, Kothari D (1995) Economic dispatch with non-monotonically increasing incremental cost units and transmission system losses. Discussion. IEEE Trans Power Syst 10:891–897
5. Kumaran G, Mouly VSRK (2001) Using evolutionary computation to solve the economic load dispatch problem. In: Proceedings of the 2001 congress on evolutionary computation, vol 1, pp 296–301
6. Adhinarayanan T, Sydulu M (2006) Particle swarm optimisation for economic dispatch with cubic fuel cost function. In: TENCON 2006, IEEE region 10 conference, pp 1–4
7. Adhinarayanan T, Sydulu M (2006) Fast and effective algorithm for economic dispatch of cubic fuel cost based thermal units. In: First international conference on industrial and information systems, pp 156–160
8. Theerthamalai A, Maheswarapu S (2010) An effective non-iterative "λ-logic based" algorithm for economic dispatch of generators with cubic fuel cost function. Int J Electr Power Energy Syst 32:539–542
9. Jalid S, Amoli NA, Shayanfar HA, Barzinpour F (2012) Solving economic dispatch problem with cubic fuel cost function by firefly algorithm. In: 8th international conference on technical and physical problems of power engineering, pp 5–7

10. Ching-Tzong S, Chien-Tung L (2000) New approach with a Hopfield modeling framework to economic dispatch. IEEE Trans Power Syst 15:541–545
11. Yalcinoz T, Altun H, Hasan U (2000) Constrained economic dispatch with prohibited operating zones: a Hopfield neural network approach. In: Electrotechnical conference. MELECON 2000. 10th Mediterranean, pp 570–573
12. Su C-T, Chiou G-J (1997) An enhanced Hopfield model for economic dispatch considering prohibited zones. Electr Power Syst Res 42:72–76
13. Chiang C-L (2005) Improved genetic algorithm for power economic dispatch of units with valve-point effects and multiple fuels. IEEE Trans Power Syst 20:1690–1699
14. Orero S, Irving M (1996) Economic dispatch of generators with prohibited operating zones: a genetic algorithm approach. In: IEE proceedings generation, transmission and distribution, pp 529–534
15. Su C-T (2004) Nonconvex power economic dispatch by improved genetic algorithm with multiplier updating method. Electr Power Compon Syst 32:257–273
16. Jayabarathi T, Jayaprakash K, Jeyakumar D, Raghunathan T (2005) Evolutionary programming techniques for different kinds of economic dispatch problems. Electr Power Syst Res 73:169–176
17. Neto JXV, Bernert DLdA, Coelho LdS (2011) Improved quantum-inspired evolutionary algorithm with diversity information applied to economic dispatch problem with prohibited operating zones. Energy Convers Manag 52:8–14
18. Wong K, Fung C (1993) Simulated annealing based economic dispatch algorithm. In: IEE Proceedings C (Generation, Transmission and Distribution), pp 509–515
19. Hemamalini S, Simon SP (2010) Artificial bee colony algorithm for economic load dispatch problem with non-smooth cost functions. Electr Power Compon Syst 38:786–803
20. Panigrahi B, Yadav SR, Agrawal S, Tiwari M (2007) A clonal algorithm to solve economic load dispatch. Electr Power Syst Res 77:1381–1389
21. Bhattacharya A, Chattopadhyay PK (2010) Biogeography-based optimization for different economic load dispatch problems. IEEE Trans Power Syst 25:1064–1077
22. Li Y-L, Zhan Z-H, Gong Y-J, Chen W-N, Zhang J, Li Y (2015) Differential evolution with an evolution path: a deep evolutionary algorithm. IEEE Trans Cybern 45(9):1798–1810
23. Özyön S, Yaşar C, Temurtaş H (2011) Diferansiyel gelişim algoritmasının valf nokta etkili konveks olmayan ekonomik güç dağıtım problemlerine uygulanması. In: 6th international advanced technologies symposium (IATS'11), electrical & electronics technologies papers, pp 181–186
24. Gaing Z-L (2003) Particle swarm optimization to solving the economic dispatch considering the generator constraints. IEEE Trans Power Syst 18:1187–1195
25. Park J-B, Jeong Y-W, Shin J-R, Lee KY (2010) An improved particle swarm optimization for nonconvex economic dispatch problems. IEEE Trans Power Syst 25:156–166
26. Chaturvedi KT, Pandit M, Srivastava L (2008) Self-organizing hierarchical particle swarm optimization for nonconvex economic dispatch. IEEE Trans Power Syst 23:1079–1087
27. Kuo C-C (2008) A novel coding scheme for practical economic dispatch by modified particle swarm approach. IEEE Trans Power Syst 23:1825–1835
28. Selvakumar AI, Thanushkodi K (2007) A new particle swarm optimization solution to nonconvex economic dispatch problems. IEEE Trans Power Syst 22:42–51
29. Niknam T, Mojarrad HD, Meymand HZ (2011) A new particle swarm optimization for non-convex economic dispatch. Eur Trans Electr Power 21:656–679
30. Cai J, Ma X, Li L, Haipeng P (2007) Chaotic particle swarm optimization for economic dispatch considering the generator constraints. Energy Convers Manag 48:645–653
31. Li Y, Zhan Z-H, Lin S, Zhang J, Luo X (2015) Competitive and cooperative particle swarm optimization with information sharing mechanism for global optimization problems. Inf Sci 293:370–382
32. Lalwani S, Kumar R, Gupta N (2015) A novel two-level particle swarm optimization approach for efficient multiple sequence alignment. Memetic Comput 7:119–133
33. Nadeem Malik T, ul Asar A, Wyne MF, Akhtar S (2010) A new hybrid approach for the solution of nonconvex economic dispatch problem with valve-point effects. Electr Power Syst Res 80:1128–1136
34. Thitithamrongchai C, Eua-Arporn B (2007) Self-adaptive differential evolution based optimal power flow for units with non-smooth fuel cost functions. J Electr Syst 3:88–99
35. Lin W-M, Cheng F-S, Tsay M-T (2001) Nonconvex economic dispatch by integrated artificial intelligence. IEEE Trans Power Syst 16:307–311
36. Panigrahi BK, Ravikumar V (2008) Bacterial foraging optimisation: Nelder–Mead hybrid algorithm for economic load dispatch. IET Gener Transm Distrib 2:556
37. Erlich I, Venayagamoorthy GK, Worawat N (2010) A mean-variance optimization algorithm. In: Evolutionary computation (CEC). IEEE Congress, pp 1–6
38. Rueda JL, Erlich I (2013) Evaluation of the mean-variance mapping optimization for solving multimodal problems. In: IEEE symposium on swarm intelligence (SIS), pp 7–14
39. Khoa TH, Vasant PM, Singh BSM, Dieu VN (2015) Swarm-based mean-variance mapping optimization (MVMOS) for solving nonconvex economic dispatch problems. Handbook of research on swarm intelligence in engineering, p 211
40. Sajjadi S, Kazemzadeh R (2016) A new analytical maclaurin series based λ-logic algorithm to solve the non-convex economic dispatch problem considering valve-point effect. Fuel 10:6
41. Lee FN, Breipohl AM (1993) Reserve constrained economic dispatch with prohibited operating zones. IEEE Trans Power Syst 8:246–254
42. Somasundaram P, Kuppusamy K, Kumudini Devi RP (2004) Economic dispatch with prohibited operating zones using fast computation evolutionary programming algorithm. Electr Power Syst Res 70:245–252
43. Subbaraj P, Rengaraj R, Salivahanan S, Senthilkumar T (2010) Parallel particle swarm optimization with modified stochastic acceleration factors for solving large scale economic dispatch problem. Int J Electr Power Energy Syst 32:1014–1023