Applied Soft Computing 48 (2016) 597–611

Contents lists available at ScienceDirect

Applied Soft Computing

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

Cultural quantum-behaved particle swarm optimization for environmental/economic dispatch☆

Tianyu Liu*, Licheng Jiao, Wenping Ma, Jingjing Ma, Ronghua Shang

Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, International Research Center for Intelligent Perception and Computation, Xidian University, Xi'an, Shaanxi Province 710071, China

A R T I C L E I N F O

Article history:
Received 9 July 2015
Received in revised form 6 April 2016
Accepted 18 April 2016
Available online 3 August 2016

Keywords:
Environmental/economic dispatch
Quantum-behaved particle swarm optimization
Cultural evolution mechanism
Local search
Multiobjective optimization

A B S T R A C T

In this paper, a novel CMOQPSO algorithm is proposed, in which cultural evolution mechanism is introduced into quantum-behaved particle swarm optimization (QPSO) to solve multiobjective environmental/economic dispatch (EED) problems. There are growing concerns about the ability of QPSO to handle multiobjective optimization problems. Two important issues in extending QPSO to multiobjective context are the construction of exemplar positions for each particle and the maintenance of population diversity. In the proposed CMOQPSO, one particle is measured for multiple times at each iteration in order to enhance its global searching ability. Belief space, which is based on cultural evolution mechanism and contains different types of knowledge extracted from the particle swarm, is adopted to generate global best positions for the multiple measurements of each particle. Moreover, to maintain population diversity and avoid premature, a novel local search operator, which is based on the knowledge in belief space, is proposed in this paper. CMOQPSO is compared with several state-of-art algorithms and tested on EED systems with 6 and 40 generators respectively. The comparative results demonstrate the effectiveness of the proposed algorithm.

1. Introduction

Economic dispatch (ED) can be formulated as a nonlinear constraint problem which aims to minimize the total fuel cost while satisfying several equal and unequal constraints by operating electric power systems. However, power generators using fossil fuel release some contaminants, which are the major contributors to air pollution. With the rising of public awareness of environmental protection, air pollution has become another important consideration in allocating optimal outputs of power generators. In this case, ED has been changed into an environmental/economic dispatch (EED) [1] problem. EED minimizes total fuel cost and pollution emission simultaneously and can be seen as a nonlinear multiobjective optimization problem with several constraints.

Various algorithms have been proposed to solve EED problems. These algorithms can be classified into two categories. For the first category, EED has been treated as a single objective problem by using different strategies. In [2], EED has been reduced to a single objective problem by considering pollution emission as a constraint. The algorithm is implemented without considering the tradeoff between fuel cost and emission. In addition, only a single solution can be obtained in an independent run by the algorithm. In [3], ϵ-constraint method has been proposed to solve multiobjective problems. In ϵ-constraint method, the most preferred objective is optimized and the other objectives are treated as constraints bounded by allowable levels. In [4], a new algorithm based on ϵ-constraint method is proposed to solve EED problems. The most obvious weakness of the algorithm is that it is time-consuming and tends to find weakly nondominated solutions. In [5–7], EED is treated as a single objective optimization problem by the linear combination of all the objectives. A set of nondominated solutions can be obtained by using different weight parameters. So in order to get a Pareto front, multiple runs are required. The algorithms belong to the second category deal with the two objectives in EED simultaneously. In [8], a fuzzy satisfaction-maximizing decision approach has been proposed to solve EED problems. In [9], a multiobjective stochastic search technique has been introduced for solving EED. The major drawback of the technique is that it is time-consuming and easily trapped into local optima. Because of the robustness and parallelism, evolutionary algorithms (EAs) have been applied to solve various kinds of optimization problems

☆ This document is a collaborative effort.

* Corresponding author.
E-mail addresses: tianyu_liu@163.com (T. Liu), lchjiao@mail.xidian.edu.cn (L. Jiao), wpma@mail.xidian.edu.cn (W. Ma), jjma@ieee.org (J. Ma), rhshang@mail.xidian.edu.cn (R. Shang).

http://dx.doi.org/10.1016/j.asoc.2016.04.021
1568-4946/

successfully [10–13]. Recently, using multiobjective EAs to solve EED problems has aroused general concern. Many evolutionary algorithms have been adopted to solve EED problems successfully, such as niched Pareto genetic algorithm (NPGA) [14], strength Pareto evolutionary algorithm (SPEA) [15], non-dominated sorting genetic algorithm (NSGA) [16], differential evolution algorithm (DE) [17–19], estimation of distribution algorithm (EDA) [20], and so on.

Particle swarm optimization (PSO), a population-based stochastic searching technology, was firstly proposed in 1995 [21]. Owing to its simplicity and facile realization, PSO has been widely used in dealing with many real-world problems [22–29]. However, some studies have demonstrated that global convergence cannot be guaranteed in a PSO system [30,31]. In order to overcome this disadvantage of PSO, quantum-behaved particle swarm optimization (QPSO) is proposed [32]. QPSO has been proved to be global convergent according to the analysis in [33]. Recently, quantum-behaved particle swarm optimization (QPSO) has attracted more and more attention. In QPSO, a significant parameter that can influence the measurement (update) of a particle is its attractor position, which is constructed by the particle's personal and global best positions. In multiobjective optimization, it is hard to find the best individual which could optimize all the objectives simultaneously, since the objectives are mutually conflictive. So, how to obtain particles' personal and global best positions is the major problem needs to be solved in extending QPSO to multiobjective context. In [34], a classic multiobjective QPSO, called MOQPSO, is proposed. In MOQPSO, a modified sigma method is adopted to generate the global best position for each particle. Another problem that can affect the performance of QPSO in optimizing multiobjective problems is the population diversity. With poor population diversity, it is easy for QPSO to fall into local optima and run into premature.

In this paper, a novel CMOQPSO algorithm is proposed, in which cultural evolution mechanism [35] is introduced into QPSO to solve the two above-mentioned problems. The main differences between CMOQPSO and traditional QPSO are listed below.

(a) Inspired by cultural evolution mechanism, belief space, which contains three types of knowledge extracted from particle swarm, is adopted in CMOQPSO.
(b) In CMOQPSO, each particle is measured for multiple times at a single iteration. If the tested problem has M objectives, then each particle will be measured for $M+1$ times. The first M measurements only consider one objective separately. For example, the 1st measurement focuses on the 1st objective, the 2nd measurement focuses on the 2nd objective, and so on. The last measurement ($(M+1)$th measurement) takes into account all the objectives. For each particle, the global best positions of the $M+1$ measurements can be obtained by using the knowledge in belief space. Multiple measurements can enhance the global searching ability of the algorithm.
(c) A novel local search operator, which is guided by the knowledge in belief space, is proposed to maintain population diversity and avoid premature convergence in this paper.

To summarize, in the proposed CMOQPSO, each particle is measured for multiple times. The global best position for each measurement can be obtained according to the knowledge in belief space. So, there exists a continuous cycle between particle swarm and belief space. Specifically, the knowledge in belief space is extracted from particle swarm and then the particles are measured (updated) according to the knowledge stored in belief space. Moreover, a local search operator, which is guided by the knowledge in belief space, is proposed to maintain population diversity and avoid premature convergence. In this paper, belief space contains three types of knowledge, namely situational knowledge, topographical knowledge and history knowledge. Situational and topographical knowledge is used to generate global best position for each measurement. Topographical knowledge is adopted in the proposed local search operator. The detailed description of CMOQPSO has been given in Section 3.3.

The proposed algorithm is adopted to solve EED problems and tested on two EED systems. The contents of this paper are organized as follows. Section 2 introduces the formulation of EED problems. Section 3 gives the detailed description of CMOQPSO for solving EED problems. Section 4 shows the comparative experiments and achieved results. Section 5 draws the concluding remarks.

2. Formulation of EED problems

EED can be formulated as a constrained multiobjective problem which minimizes two conflicting objectives, i.e. the total fuel cost and the emission of harmful pollutants of the tested power system.

2.1. Objective functions

2.1.1. Fuel cost function

The total fuel cost can be calculated as a quadratic function, which is shown in Eq. (1). Where, P_i and $F(P_i)$ are the power output and the fuel cost of the ith generator respectively. N_g is the number of generators in power system.

$$\min \sum_{i=1}^{N_g} F(P_i) = \sum_{i=1}^{N_g} (a_i + b_i P_i + c_i P_i^2) \tag{1}$$

Taking into account the practical operating conditions of power generators, the fuel cost function can be modified by adding a nonlinear sinusoid function as shown in Eq. (2) [36]. Where, a_i, b_i, c_i, d_i and e_i are the fuel cost coefficients of the ith generator. $P_i^{\min}$ is the minimized power output of the ith generator.

$$\min \sum_{i=1}^{N_g} F(P_i) = \sum_{i=1}^{N_g} (a_i + b_i P_i + c_i P_i^2 + |e_i \sin(d_i(P_i^{\min} - P_i))|) \tag{2}$$

2.1.2. Emission function

The total emission of harmful pollutants can be calculated by Eq. (3) [37]. Where, α_i, β_i, γ_i, ε_i and λ_i are the emission coefficients of the ith generator.

$$\min \sum_{i=1}^{N_g} E(P_i) = \sum_{i=1}^{N_g} (\alpha_i + \beta_i P_i + \gamma_i P_i^2 + \varepsilon_i e^{\lambda_i P_i}) \tag{3}$$

2.2. Constraints

2.2.1. Output constraint of each generator

The output of each generator must be within the given range. That is, the following constraint, as shown in Eq. (4), must be satisfied for each generator. Where, $P_i^{\min}$ and $P_i^{\max}$ are the boundary values. P_i is the output of the ith generator.

$$P_i^{\min} \le P_i \le P_i^{\max} \tag{4}$$

2.2.2. Power balance constraint of system

The power balance constraint can be formulated as Eq. (5). P_D is total demand and P_{Loss} is total transmission loss.

$$\sum_{i=1}^{N_g} P_i = P_D + P_{Loss} \tag{5}$$

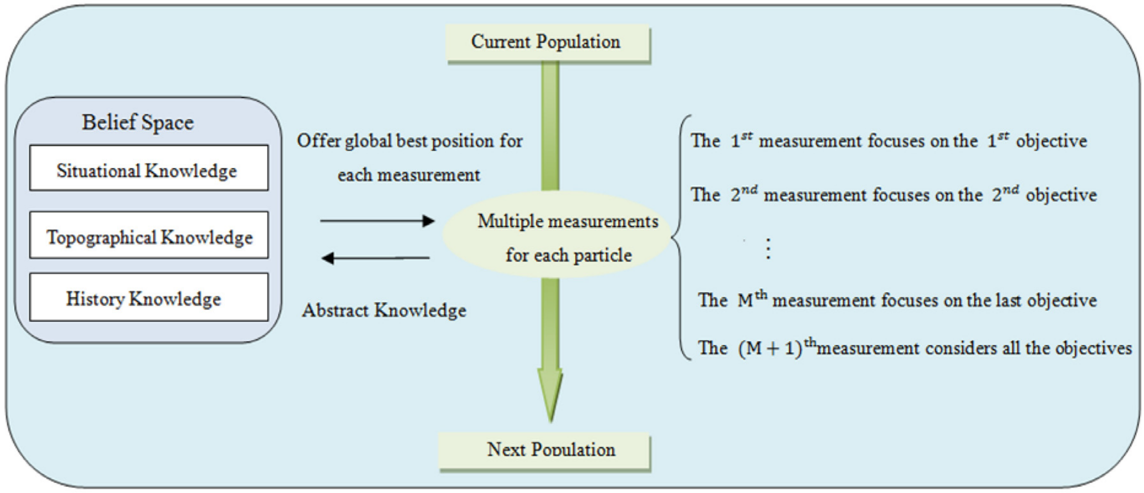

Fig. 1. Scheme of CMOQPSO.

P_{Loss} can be calculated by Eq. (6) [38]. Where, B, B_0 and B_{00} are the loss coefficients.

$$P_{Loss} = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} P_i B_{i,j} P_j + \sum_{i=1}^{N_g} B_{0i} P_i + B_{00} \tag{6}$$

2.3. Formulation of EED

According to the above description, EED can be formulated as a two-objective optimization problem with equal and unequal constraints. It can be modelled as shown in Eq. (7). N_g is the number of generators.

$$\begin{aligned} &\min M = [\sum_{i=1}^{N_g} F(p_i), \sum_{i=1}^{N_g} E(P_i)] \\ &s.t.\quad P_i^{\min} \le P_i \le p_i^{\max}, i = 1, 2, \ldots, N_g \\ &\sum_{i=1}^{N_g} P_i = P_D + P_{Loss} \end{aligned} \tag{7}$$

3. CMOQPSO for solving EED problems

3.1. Quantum-behaved particle swarm optimization (QPSO)

A brief introduction of QPSO [32] is given in this section. In QPSO, the update process of a particle can be seen as a measurement. For each particle, its personal and global best positions play an important role in the measurement. In QPSO, particle x_i is updated according to the following Eq. (8) [39].

$$\begin{aligned} &x_{i,j}(t+1) = attractor_{i,j} \pm (\beta \cdot |mbest_{i,j}(t) - x_{i,j}(t)|) \cdot \ln(\frac{1}{\mu}) \\ &attractor_{i,j}(t) = \varphi \cdot PBEST_{i,j}(t) + (1-\varphi) \cdot GBEST_{i,j}(t) \\ &mbest(t) = \frac{1}{N}\sum_{i=1}^{N} PBEST_i(t) \end{aligned} \tag{8}$$

where N is population size. $j \in \{1, 2, \ldots, D\}$ and D is the dimension of the searching space. $PBEST_i$ and $GBEST_i$ are the personal and global best positions of x_i respectively. β is the contraction-expansion coefficient [40], which usually decreases linearly with the running of QPSO. μ and φ are random numbers within [0, 1].

3.2. Cultural evolution mechanism

Cultural evolution mechanism, which is inspired by the cultural evolution process of human, has been formulated as a general form, i.e. cultural algorithm (CA) [35]. CA consists of two spaces: population space and belief space. Population space contains the evolving information of individual population while belief space consists of different types of knowledge extracted from population space. The knowledge stored in belief space has developed a universal mode. Five types of knowledge are usually used, namely normative knowledge, situational knowledge, domain knowledge, history knowledge and topographical knowledge. There exists a continuous cycle between population space and belief space until the termination condition is met. Specifically, individuals in the population are updated by using the knowledge in belief space. Then the knowledge in belief space is modified according to the updated population.

There are a lot of studies which concentrates on combing CA with evolutionary algorithms [41–46]. It is found that the performance of evolutionary algorithm is obviously improved by cooperating with CA.

3.3. Cultural quantum-behaved particle swarm optimization (CMOQPSO)

Inspired by the framework of CA, belief space is adopted in CMOQPSO. In CMOQPSO, belief space only contains three types of knowledge, i.e. situational knowledge, topographical knowledge and history knowledge. Fig. 1 shows the scheme of CMOQPSO and M is the number of objectives. In CMOQPSO, a particle will be measured for multiple times, then the next position of the particle is selected from these obtained individuals. As shown in Fig. 1, the measurements of each particle are implemented based on the knowledge in belief space. More specifically, the global best position used in each measurement is obtained by using the knowledge in belief space. Then the knowledge stored in belief space is modified according to the particles obtained by multiple measurements.

Fig. 2 shows the flowchart of CMOQPSO. The detailed procedures of the major operators in CMOQPSO are shown below. Suppose the population size is N, the dimension of searching space and objective space are D and M respectively.

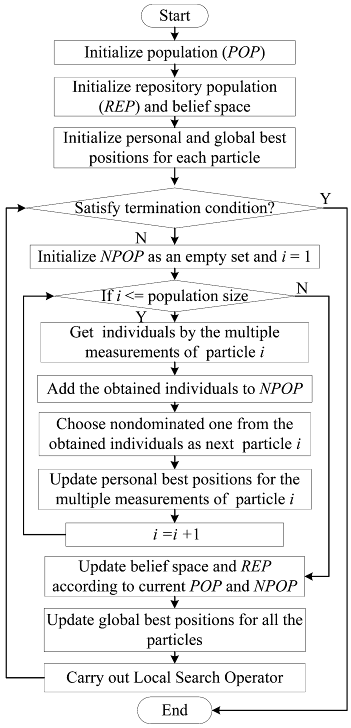

Fig. 2. Flowchart of CMOQPSO.

3.3.1. Initialize

(a) Population (*POP*)

POP is initialized by generating *N* particles and each particle contains *D* random numbers which are uniformly distributed in the given ranges.

(b) Repository population (*REP*)

REP is initialized as the nondominated solutions in *POP*.

(c) Belief space

Belief space consists of situational knowledge, topographical knowledge and history knowledge.

The representation of situational knowledge is shown in Fig. 3. Situational knowledge contains *M* parts. The *i*th part $PART_i = \{best_{i,1}, best_{i,2}, \ldots, best_{i,s}\}$ and $best_{i,j}$ $(j = 1, 2, \ldots, s)$ is the *j*th best solution for the *i*th objective in the past searching history. Situational knowledge is adopted to generate the global best positions for the first *M* measurements of each particle.

Topographical knowledge is constructed based on geographical mechanism [47]. It records the information of the best area in objective space. In other words, the best area in objective space is divided into several subcells uniformly and topographical knowledge records the lower and upper limit values of each subcell. Fig. 4 shows the representation of topographical

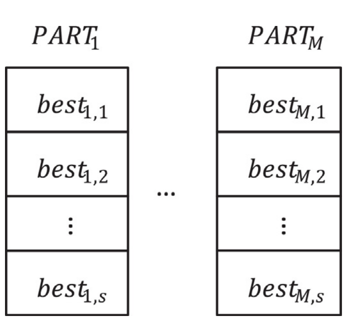

Fig. 3. Situational knowledge.

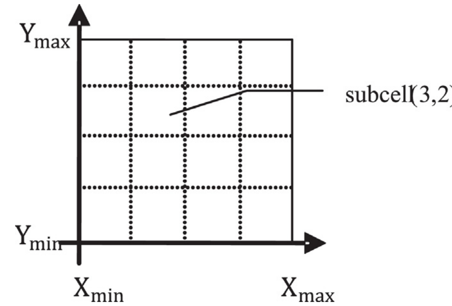

Fig. 4. Topographical knowledge.

knowledge. Suppose the dimension of the objective space is 2 and the best area in objective space is divided into 4 × 4 subcells. For *subcell*(*i*, *j*), its lower and upper limit values can be calculated by Eq. (9). Topographical knowledge is used to obtain global best position for the last measurement ((*M* + 1)th) of each particle.

$$L_x = \frac{X_{\max} - X_{\min}}{4} \cdot (i-1) + X_{\min}, \quad U_x = \frac{X_{\max} - X_{\min}}{4} \cdot i + X_{\min}$$
$$L_y = \frac{Y_{\max} - Y_{\min}}{4} \cdot (j-1) + Y_{\min}, \quad U_y = \frac{Y_{\max} - Y_{\min}}{4} \cdot j + Y_{\min} \tag{9}$$

History knowledge is used to guide the local search operator in this paper. It can be represented as shown in Fig. 5. H_i $(i = 1, 2, \ldots, M)$ is the number of iterations for which the best solution for the *i*th objective has been trapped. Which means, if the best solution for the *i*th objective has been hold for *t* iterations, then $H_i = t$. In this paper, each member in history knowledge is initialized as 0.

(d) Personal and global best positions

Suppose $PBEST_i^k$ and $GBEST_i^k$ are the personal and global best positions for the *k*th measurement of particle x_i, respectively. As shown in Eq. (10), the initialization of $\{PBEST_i^k | k = 1, 2, \ldots, M\}$ is same to that of $PBEST_i^{M+1}$.

$$PBEST_i^k = x_i, \quad k = 1, 2, \ldots, M+1 \tag{10}$$

However, the initialization of $\{GBEST_i^k | k = 1, 2, \ldots, M\}$ is different from that of $GBEST_i^{M+1}$. According to Eq. (11), for the *k*th objective $(1 \le k \le M)$, the global best position of the whole swarm are exactly the same. Where, $best_{k,1}$ is the element in situational knowledge and means the best individual found in the past searching history for the *k*th objective. So, $GBEST^k$ is used to represent the global best position for the *k*th measurement of the whole swarm. For particle x_i $(i \in 1, 2, \ldots, N)$, its global best position for the (*M* + 1)th measurement ($GBEST_i^{M+1}$) is randomly selected from *REP*. So, different particles may have different global best positions for the (*M* + 1)th measurement in the initialization step.

$$GBEST_i^k = best_{k,1}, \quad i = 1, 2, \ldots, N \tag{11}$$

3.3.2. Update

(a) Population (*POP*)

In this section, particle x_i is taken as an example to illustrate the update process in CMOQPSO. As shown in Fig. 6, x_i is measured for *M* + 1 times according to Eq. (8), and then

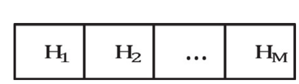

Fig. 5. History knowledge.

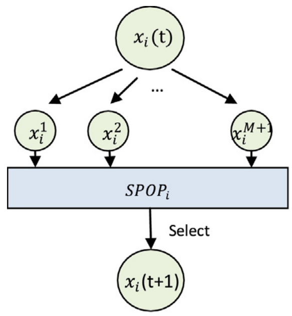

Fig. 6. Update of particle x_i.

$SPOP_i = \{x_i^k | k = 1, 2, \ldots, M+1\}$ is obtained. x_i will be updated by randomly selecting an individual from the nondominated solutions in $SPOP_i$.

(b) Repository population (*REP*)

REP contains the nondominated solutions found so far. As Fig. 2 shows, *NPOP* contains the individuals obtained by the measurements of all the particles at current iteration. *REP* can be updated by selecting nondominated solutions from *REP* and *NPOP*. If the size of *REP* exceeds the given maximum value, then the nondominated solutions in *REP* will be reduced by crowding distance sorting [48].

(c) Belief space

As mentioned in Section 3.3.1, situational knowledge can be divided into *M* parts. Before updating $PART_i = \{best_{i,1}, best_{i,2}, \ldots, best_{i,s}\}$ ($i = 1, 2, \ldots, M$), a hypermutation operator will be carried out. The hypermutation operator will be implemented by Eq. (12) [49]. Where, $best_{i,j}$ is the *j*th best solution for the *i*th objective in the past searching history, as shown in Fig. 3. $best_{i,j}(k)$ is the *k*th dimension of $best_{i,j}$. $F_{i,j}(k)$ is the *k*th dimension of the fitness value of $best_{i,j}$ and $MinF_i$ is the minimized fitness value of all the members in $PART_i$. $R(0, 1)$ is a random number within [0, 1]. $P^{\max}(k)$ and $P^{\min}(k)$ are the boundary values of the *k*th dimension of the searching space. ζ is self-adaptively controlled and its setting can be found in [49]. Then add current *POP* and the individuals after hypermutation to original $PART_i$. Since *s* is the given size of $PART_i$, the size of $PART_i$ will be limited to *s* by selecting *s* best individuals for the *i*th objective.

$$best_{i,j}(k) = best_{i,j}(k) + \zeta \cdot (F_{i,j}(k)/MinF_i(k)) \cdot R(0,1) \cdot (P^{\max}(k) - P^{\min}(k)) \quad (12)$$

Topographical knowledge is updated according to *REP*. The best area in objective space can be re-determined by the updated *REP*. Then topographical knowledge can be obtained by dividing the best area into several subcells.

As can be seen from Fig. 6, $SPOP_i$ is acquired by the measurements of x_i. Add up all $SPOP_i$ ($i = 1, 2, \ldots, N$) and get a new population *NPOP*. If the best fitness value for the *k*th objective in *NPOP* is better than $GBEST^k$, then $H_k = 0$, otherwise $H_k = H_k + 1$.

(d) Personal and global best positions

Suppose $PBEST_i^k$ and $GBEST_i^k$ are the personal and global best positions for the *k*th measurement of particle x_i, respectively. $SPOP_i = \{sp_1, sp_2, \ldots, sp_{M+1}\}$ contains the individuals which are obtained by the multiple measurements of particle x_i.

For particle x_i, the update of $\{PBEST_i^k | k = 1, 2, \ldots, M\}$ is different from that of $PBEST_i^{M+1}$. Suppose sp_j, which belongs to $SPOP_i$, has the best fitness value for the *k*th objective. If sp_j is better than $PBEST_i^k$ for the *k*th objective, then $PBEST_i^k$ will be replaced by sp_j. Otherwise, $PBEST_i^k$ will remain unchanged. For $PBEST_i^{M+1}$, if there exists a nondominated individual in $SPOP_i$ which can dominate $PBEST_i^{M+1}$, then $PBEST_i^{M+1}$ will be replaced by the nondominated individual. Otherwise, $PBEST_i^{M+1}$ will remain unchanged.

For particle x_i, the update of $\{GBEST_i^k | k = 1, 2, \ldots, M\}$ is different from that of $GBEST_i^{M+1}$. The update of $\{GBEST_i^k | k = 1, 2, \ldots, M\}$ is shown in Eq. (11). The strategy adopted to update $GBEST_i^{M+1}$ is borrowed from [48]. In this strategy, $GBEST_i^{M+1}$ is obtained by selecting an individual form *REP* according to topographical knowledge and roulette-wheel selection method.

3.3.3. Local search operator

To enhance the algorithm's ability of jumping out of local optima, a novel local search operator is carried out at each iteration. The local search operator is implemented according to history knowledge. In this operator, the global best position for each objective will be operated as follows. In order to illustrate, the *k*th objective will be taken as an example. Suppose $GBEST^k$ is the global best position for the *k*th objective. The members in $\{PBEST_i^k | i = 1, 2, \ldots, N\}$ are the personal best positions of the whole swarm for the *k*th objective and *N* is the population size. First, a local searching area needs to be determined. The limit values of the local searching area can be obtained according to the following Eq. (13).

$$\begin{aligned} L &= 0.5 \cdot \min\{PBEST_i^k | i = 1, 2, \ldots, N\} + 0.5 \cdot GBEST^k \\ U &= 0.5 \cdot \max\{PBEST_i^k | i = 1, 2, \ldots, N\} + 0.5 \cdot GBEST^k \end{aligned} \quad (13)$$

where *L* and *U* are the lower and upper limit values of the local searching area. Then, each dimension of the local searching area will be divided into *d* parts uniformly. So, for each dimension of the local searching area, *d* random numbers, which are within the ranges of the obtained *d* parts respectively, can be generated. The value of *d* is determined by the history knowledge, as shown in Eq. (14). For the *j*th ($j = 1, 2, \ldots, D$) dimension, try to replace $GBEST^k(j)$ with the corresponding *d* random numbers one after another. After that, *d* new individuals will be obtained. Select the best one for the *k*th objective among the *d* obtained individuals as the new $GBEST^k$. The detailed procedure of the proposed local search operator is shown in Algorithm 1.

$$d = \begin{cases} 5, & H_k \le 10 \\ 10, & H_k > 10 \end{cases}, \quad k = 1, 2, \ldots, M \quad (14)$$

Algorithm 1. Procedure of local search operator

```
k = 1;
While k ≤ M, do:
Determine limit values of local searching area (i.e. L and U);
Get the value of d according to Eq. (14);
t = 1;
While t ≤ D, do:
   a) Randomly select a dimension x;
   b) Divide [L(x), U(x)] into d parts uniformly, that is {[L(x), L1), [L1, L2), ...,
   [Ld−1, U(x))}
   c) For each part, get a random number which is within its corresponding
   range, and replace GBEST^k(x) with the obtained number. So, d new
   individual will be acquired in this step.
   d) Select the best individual for the kth objective as new GBEST^k;
   e) t = t + 1;
   End while.
k = k + 1;
End While.
```

3.4. CMOQPSO for solving EED problems

To solve EED problems by CMOQPSO, a constraints-handling operator should be cooperated with the basic framework of CMOQPSO.

3.4.1. Constraints-handling operator

The constraints-handling operator is adopted to limit individuals to feasible regions. In this paper, two different cases of EED problems are considered. The first case (Case1) is EED without transmission loss and the other one (Case2) is EED with transmission loss. The constraints-handling operator adopted in this paper is based on *Dif*, which represents the extent of the unfeasibility of individuals [50]. *Dif* can be obtained by Eq. (15). Where, x_i is the particle needs to be repaired. P_D and P_L are the total system demand and transmission loss respectively. Suppose the dimension of x_i is *D*. $P^{\min}(k)$ and $P^{\max}(k)$ are the boundary values of the *k*th dimension in the searching space. The detailed procedure of the constraints-handling operator is shown in Algorithm 2.

$$Dif = \begin{cases} P_D - sum(x_i) & if \quad Case1 \\ P_D + P_{Loss} - sum(x_i) & if \quad Case2 \end{cases} \tag{15}$$

Algorithm 2. Procedure of constraints-handling operator

```
1: Select a random integer k from 1 to D.
2: While x_i is unfeasible, do:
3:    Get Dif by Eq. (15);
4:    x_i(k) = x_i(k) + Dif;
5:    x_i(k) = min[x_i(k), P^max(k)] and x_i(k) = max[x_i(k), P^min(k)]
6:    k = mod(k, D) + 1;
7: End while.
```

3.4.2. Flowchart of CMOQPSO for solving EED problems

Fig. 7 is the detailed flowchart of CMOQPSO for solving EED problems. The difference between Fig. 7 and 2 is that only Fig. 7 has the 'constraints-handling operator'. In CMOQPSO, the constraint-handling operator is carried out in initializing and updating the particle swarm.

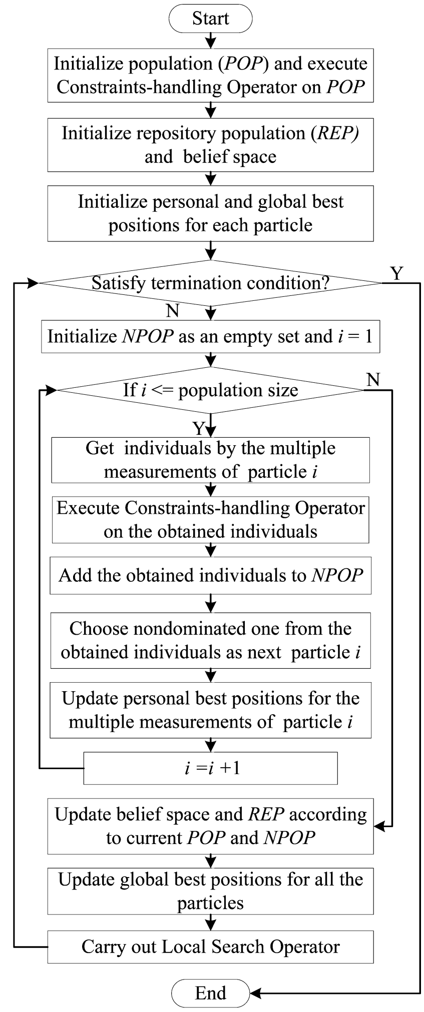

Fig. 7. Flowchart of CMOQPSO for solving EED problems.

4. Experiments and results

To test the performance of the proposed CMOQPSO algorithm, two EED systems with 6 generators and 40 generators are adopted respectively. In this section, CMOQPSO is compared with 8 algorithms, namely MOQPSO [34], MOPSO [51], MA θ-PSO [52], MO-DE/PSO [50], SMODE [53], BSA-NDA [54], SPEA [15] and NSGA-II [55]. For all the algorithms, the size of population $n_{POP} = 50$ and the maximum size of repository population $n_{REP} = 100$. The parameter settings of all the algorithms are shown in Table 1. To compare CMOQPSO with the other algorithms fairly, the parameter β in CMOQPSO is the same as that in MOQPSO and the number of subcells in CMOQPSO is the same as that in MOPSO. In CMOQPSO, parameter *s* is used to control the size of situational knowledge. The tuning of parameter *s* is shown in Section 4.2.1. For all the algorithms, the statistical results are obtained by 30 independent runs.

4.1. Quantitative metrics

4.1.1. Spread metric SP

To estimate how well the nondominated solutions are distributed, spread metric *SP* [56] are adopted. *SP* can be calculated as shown in Eq. (16). Where, *n* and *M* are the number of nondominated solutions and the number of objectives, respectively. $f_i(k)$ is

Table 1
Parameter settings in algorithms.

Algorithm	n_{POP}	n_{REP}	Other parameters	References
CMOQPSO	50	100	$\beta: 1.2 \sim 0.5$, $s = 10$, 30 subcells	
MOQPSO	50	100	$\beta: 1.2 \sim 0.5$, $k = 4$	[34]
MOPSO	50	100	$w = 0.4$, 30 subcells	[51]
MAθ-PSO	50	100	$c_{1,\max} = c_{2,\max} = 2$, $c_{1,\min} = c_{2,\min} = 0$, $radius = 0.02$	[52]
MO-DE/PSO	50	100	$c_{1f} = c_{2f} = 2.5$, $c_{1i} = c_{2i} = 0.5$	[50]
SMODE	50	100	$F = 0.5$, $CR = 0.1$	[53]
BSA-NDA	50	100	$mixrate = 1$	[54]
SPEA	50	100	$p_{crossover} = 0.9$, $p_{mutation} = 0.2$	[15]
NSGA-II	50	100	$p_{crossover} = 0.9$, $p_{mutation} = 0.2$	[55]

Table 2
Fuel and emission coefficients.

Generator	Fuel cost			Emission					p^{min}	p^{max}
	a	b	c	α	β	γ	ε	λ		
P_1	10	200	100	4.091	−5.554	6.490	2e−04	2.857	0.05	0.5
P_2	10	150	120	2.543	−6.047	5.638	5e−04	3.333	0.05	0.6
P_3	20	180	40	4.258	−5.094	4.586	1e−06	8.000	0.05	1.0
P_4	10	100	60	5.326	−3.550	3.380	2e−03	2.000	0.05	1.2
P_5	20	180	40	4.258	−5.094	4.586	1e−06	8.000	0.05	1.0
P_6	10	150	100	6.131	−5.555	5.151	1e−05	6.667	0.05	0.6

the kth dimension of the fitness value of x_i. If $SP = 0$, then all the nondominated solutions are equidistantly spaced.

$$SP = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(\bar{d} - d_i)^2} \qquad (16)$$

$$d_i = \min_{1 \le j \le n}\left(\sum_{k=1}^{M}|f_i(k) - f_j(k)|\right)$$

4.1.2. Binary quality metric I_C

As shown in Eq. (17), $I_C(A, B)$ [57] is used to compare the coverage of two nondominated solution sets, namely A and B. Where, $\succeq$ means weakly dominate. $I_C(A, B) = 1$ means all the solutions in B are dominated by A. $I_C(A, B) = 0$ means none in B is weakly dominated by A. If $I_C(A, B) = 1$ and $I_C(B, A) = 0$, then all the solutions in B is dominated by A and none in A is dominated by B. In this case, A is better than B. If $1 \ge I_C(A, B) \ge I_C(B, A) \ge 0$, then the proportion of solutions in B which are weakly dominated by A is larger than the proportion of solutions in A which are weakly dominated by B. So, in a loose sense, A is better than B.

$$I_C(A, B) = \frac{b \in B; \exists a \in A : a \succeq b}{|B|} \qquad (17)$$

4.2. EED system with 6 generators

In this section, IEEE 30-bus system with 6 generators is adopted to test the performance of the algorithms. The total demand of the system $P_D = 2.834$. Table 2 shows fuel cost and emission coefficients [50]. Loss coefficients (B, B_0, B_{00}) [50], which are used to calculate the system loss, are given in Eq. (18). To make this a fair comparison, the maximum number of function evaluations for all the algorithms is set to 30, 000 and the statistical results are obtained by 30 independent runs.

$$B = \begin{pmatrix} 0.1382 & -0.0299 & 0.0044 & -0.0022 & -0.0010 & -0.0008 \\ -0.0299 & 0.0487 & -0.0025 & 0.0004 & 0.0016 & 0.0041 \\ 0.0044 & -0.0025 & 0.0182 & -0.0070 & -0.0066 & -0.0066 \\ -0.0022 & 0.0004 & -0.0070 & 0.0137 & 0.0050 & 0.0033 \\ -0.0010 & 0.0016 & -0.0066 & 0.0050 & 0.0109 & 0.0005 \\ -0.0008 & 0.0041 & -0.0066 & 0.0033 & 0.0005 & 0.0244 \end{pmatrix} \qquad (18)$$

$$B_0 = \begin{pmatrix} -0.0107 & 0.0060 & -0.0017 & 0.0009 & 0.0002 & 0.0030 \end{pmatrix}$$

$$B_{00} = 9.8573e-04$$

To demonstrate the effectiveness of the proposed algorithm, two cases are considered in this section.

Case1: Consider output constraints and power balance constraint without system loss.

Case2: Consider output constraints and power balance constraint with system loss.

4.2.1. Analysis of parameter s

In the proposed CMOQPSO, s controls the size of situational knowledge. In order to investigate the influence of s on *CMOQPSO*, Case1 is adopted. According to the description in Section 3.3.2 (c), a hypermutation operator is performed on the individuals stored in situational knowledge. If s is too large, then the algorithm will spend too much time on constructing new solutions and become inefficient. However, if s is too small, then the algorithm will have a weak ability of jumping out of local optima. Fig. 8 shows the mean value of the best fuel cost obtained by CMOQPSO with s changing from 2 to 20 by 30 independent runs and Fig. 9 shows the mean value of best emission obtained by CMOQPSO with s changing from 2 to 20 by 30 independent runs. It can be seen that $s = 6$ is good for both fuel cost and emission. So, in this paper, s is set to 6.

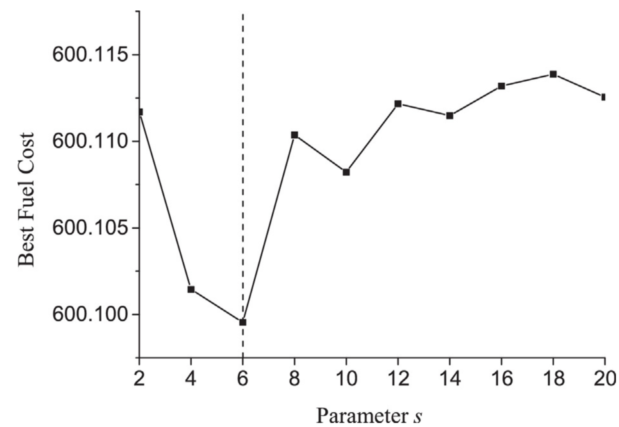

Fig. 8. Best fuel cost versus s.

4.2.2. Results comparison and analysis

Table 3 shows the best dispatch results obtained by CMOQPSO by 30 independent runs.

Table 4 and 5 give the statistical results, which include best solutions (Best), mean solutions (Mean) and standard deviations (Std), obtained by the proposed CMOQPSO and the other algorithms for Case1 and Case2 respectively. The numbers in boldface are the best values.

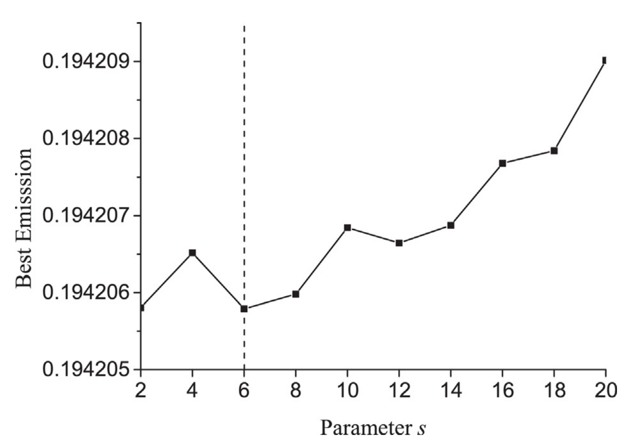

Fig. 9. Best emission versus s.

Table 3
Best dispatch result of CMOQPSO for EED system with 6 generators.

Generator	Case1		Case2	
	Best fuel cost	Best emission	Best fuel cost	Best emission
P_1	0.1102	0.4061	0.1196	0.4109
P_2	0.2990	0.4590	0.2874	0.4637
P_3	0.5267	0.5381	0.5830	0.5443
P_4	1.0152	0.3830	0.9903	0.3903
P_5	0.5237	0.5377	0.5266	0.5445
P_6	0.3592	0.5101	0.3526	0.5157
Fuel cost	600.089730	638.289451	605.977326	646.237913
Emission	0.194178	0.194203	0.220580	0.194178

It can be observed in Table 4 and 5 that the best solutions found by the proposed algorithm is better than those by the other algorithms for both fuel cost and emission. Furthermore, CMOQPSO also have obtained the best mean solutions for both Case1 and Case2. So, take into account best solution values and mean solution values, CMOQPSO has achieved the best performance for both Case1 and Case2. This may indicate the effectiveness of the strategies adopted in the proposed algorithm. More specifically, the multiple measurements in CMOQPSO enhance the global searching ability of the algorithm. The local search operator adopted in this paper make it easier for the algorithm to jump out of local optima. For Case1 and Case2, the standard deviations obtained by CMOQPSO are better than those obtained by most of the other algorithms. In other words, CMOQPSO is more stable than most of the comparative algorithms.

Table 6 gives the mean values of *SP* of all the 9 algorithms by 30 independent runs. Figs. 10 and 11 show Pareto fronts found by all the algorithms for Case1 and Case2 respectively. It can be observed from Table 6 that the mean *SP* obtained by CMOQPSO is better than those obtained by the other algorithms except MOPSO for Case1. Which means, for Case1 the nondominated solutions found by CMOQPSO are spaced more equidistantly than those found by the other algorithms except MOPSO. For Case2, CMOQPSO has obtained the best mean value of *SP*, as shown in Table 6. In other words, the nondominated solutions found by CMOQPSO are spaced more equidistantly than those found by the other algorithms for Case2.

Table 6
Comparison of *SP* of different algorithms.

Algorithm	Mean values of *SP*	
	Case1	Case2
CMOQPSO	0.0164	0.0117
MOQPSO	0.1805	0.0818
MOPSO	0.0108	0.0162
MA θ-PSO	0.0236	0.0179
MO-DE/PSO	0.0253	0.0722
SMODE	0.0556	0.1703
BSA-NDA	0.2749	0.4858
SPEA	0.0252	0.0134
NSGA-II	0.0298	0.0221

Table 7
Comparison of $I_C(A, B)$ of different algorithms.

	Case1 A:CMOQPSO		Case2 A:CMOQPSO	
B	$I_C(A, B)$	$I_C(B, A)$	$I_C(A, B)$	$I_C(B, A)$
MOQPSO	0.31	0.12	0.28	0.09
MOPSO	0.21	0.10	0.23	0.14
MA θ-PSO	0.15	0.17	0.25	0.11
MO-DE/PSO	0.27	0.13	0.48	0.02
SMODE	0.27	0.09	0.29	0.06
BSA-NDA	0.39	0.02	0.42	0.02
SPEA	0.22	0.12	0.16	0.13
NSGA-II	0.52	0.05	0.63	0.05

Table 7 shows the comparison of the mean binary quality metric I_C between CMOQPSO and the other algorithms for Case1 and Case2 according to 30 independent runs. As observed from Table 7, in a loose sense, CMOQPSO has achieved the best performance for both Case1 and Case2.

To compare the computation efficiency and convergence speed, Figs. 12 and 13 present the convergence characteristics for Case1 and Case2, respectively. To avoid making the figures too

Table 4
Statistical results of fuel cost and emission for Case1.

Algorithm	Fuel cost			Emission		
	Best	Mean	Std	Best	Mean	Std
CMOQPSO	**600.0897**	**600.0958**	3.94e−03	**0.194203**	**0.194203**	**1.37e−06**
MOQPSO	600.1171	600.1464	1.80e−02	0.194250	1.194338	5.09e−05
MOPSO	600.1119	600.1160	5.95e−03	**0.194203**	0.194217	1.12e−05
MA θ-PSO	600.1047	600.1146	2.69e−03	**0.194203**	0.194237	2.01e−05
MO-DE/PSO	600.1238	600.2018	9.60e−02	0.194205	0.194220	1.32e−05
SMODE	600.1235	600.1712	3.58e−02	0.194273	0.194722	2.49e−04
BSA-NDA	600.3846	601.1612	6.18e−01	0.194252	0.194697	3.20e−04
SPEA	600.1116	600.1147	**1.30e−03**	**0.194203**	**0.194203**	3.98e−06
NSGA-II	600.0996	600.1140	3.64e−03	**0.194203**	**0.194203**	3.38e−06

Table 5
Statistical results of fuel cost and emission for Case2.

Algorithm	Fuel cost			Emission		
	Best	Mean	Std	Best	Mean	Std
CMOQPSO	**605.9773**	**605.9848**	6.30e−03	**0.194178**	**0.194179**	8.33e−08
MOQPSO	606.0070	606.0476	2.41e−02	0.194223	0.194230	6.98e−05
MOPSO	605.9986	606.0031	**5.35e−03**	0.194179	0.194188	9.43e−06
MA θ-PSO	605.9983	606.1599	3.42e−02	0.194179	**0.194179**	**4.68e−08**
MO-DE/PSO	606.0129	606.1049	7.96e−02	0.194181	0.194194	1.26e−05
SMODE	606.0091	606.0641	3.35e−02	0.194221	0.194622	2.18e−04
BSA-NDA	606.3409	607.1607	6.48e−01	0.194285	0.194743	4.78e−04
SPEA	606.0041	606.0166	5.66e−03	0.194179	**0.194179**	5.67e−08
NSGA-II	605.9808	605.9940	8.20e−03	0.194179	0.194181	1.59e−06

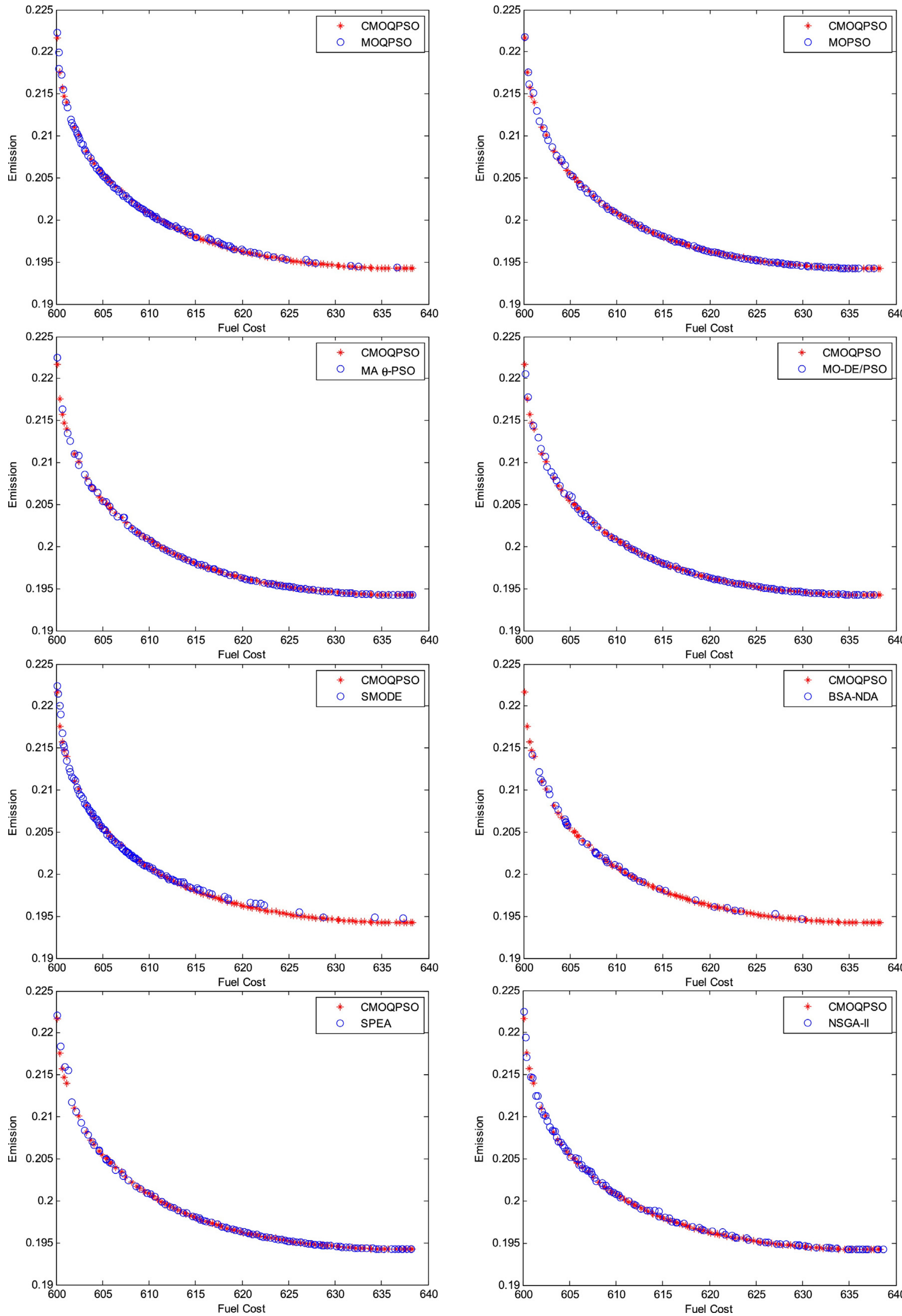

Fig. 10. Pareto fronts found by algorithms for Case1.

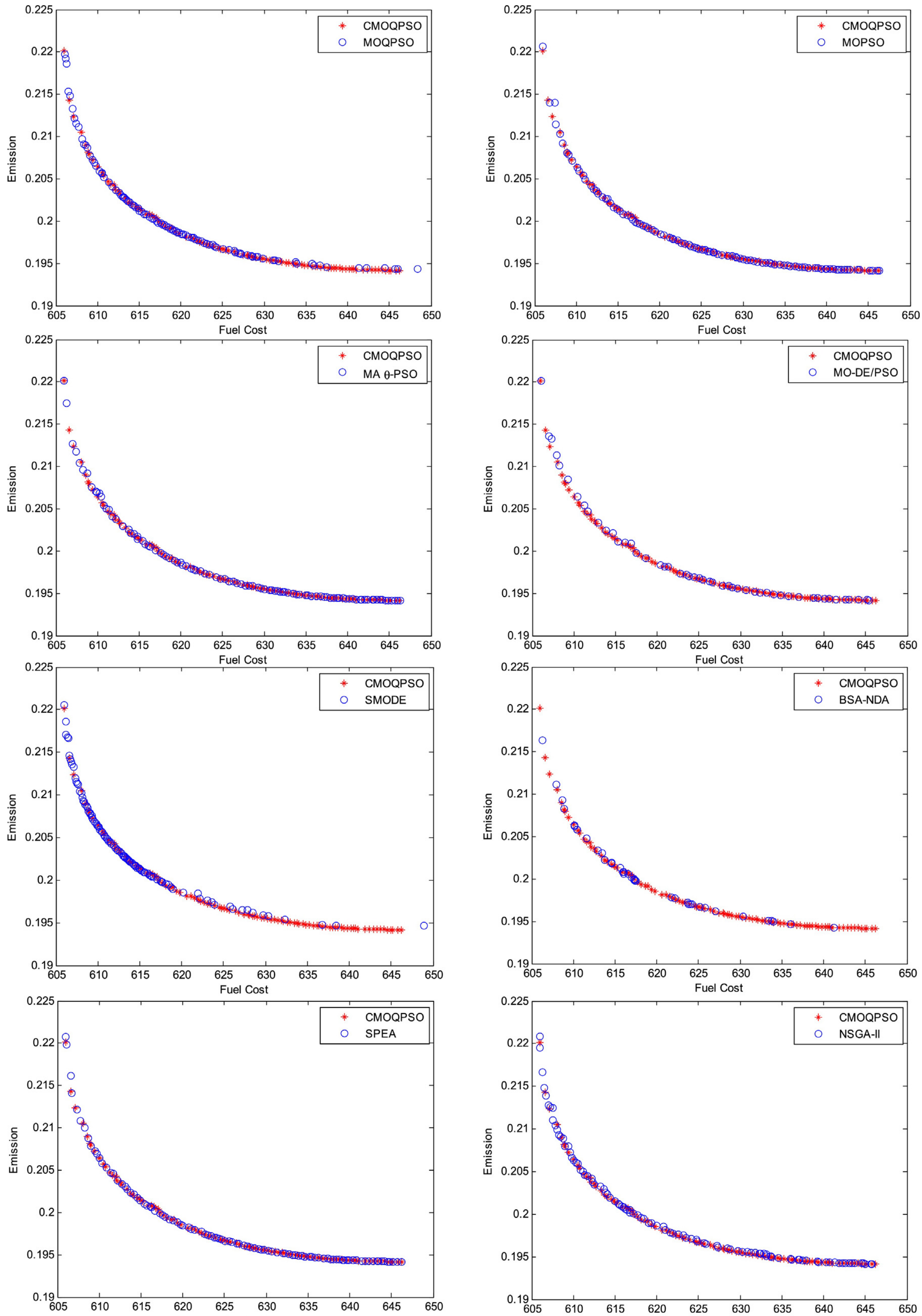

Fig. 11. Pareto fronts found by algorithms for Case2.

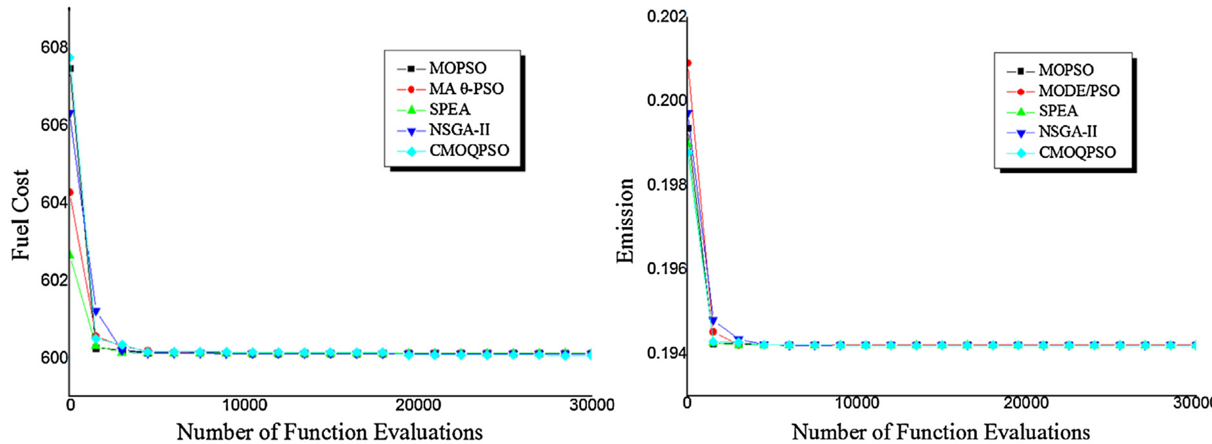

Fig. 12. Convergence characteristics of economic dispatch and emission dispatch for Case1.

complicated, the top 5 algorithms in terms of mean solutions are used for comparison. For Case1, the top 5 algorithm for fuel cost are MOPSO, MA θ-PSO, SPEA, NSGA-II and CMOQPSO. The top 5 algorithm for emission are MOPSO, MODE/PSO, SPEA, NSGA-II and CMOQPSO. It can be observed from Fig. 12 that all the algorithms have very close convergence speed and converged before 10,000 function evaluations. For Case2, the top 5 algorithm for fuel cost are NSGA-II, MOPSO, SPEA, MOQPSO, and CMOQPSO and the top 5 algorithm for emission are NSGA-II, MOPSO, SPEA, MA θ-PSO and CMOQPSO. Similarly, all the algorithms shown in Fig. 13 also have very close convergence speed and converged before 10,000 function evaluations. According to the statistical results in Table 4 and 5, it can be found that the mean solutions obtained by the proposed CMOQPSO are the best for both Case1 and Case2 and the convergence speed of CMOQPSO is comparable to those of the others algorithms shown in Fig. 12 and 13. This may indicate the computation efficiency of the proposed CMOQPSO.

4.3. EED system with 40 generators.

In this section, EED system with 40 generators is adopted. The output constraints and the power balance constraint without system loss are considered. The maximum number of function evaluations for all the algorithms is set to 50,000 and the statistic results are obtained by 30 independent runs. The fuel cost and emission coefficients are given in [58] and [52] respectively. The total load demand of the system P_D = 10,500.

4.3.1. Results comparison and analysis

Table 8 shows the best dispatch results obtained by CMOQPSO over 30 independent runs.

The comparison results of the proposed CMOQPSO with other algorithms are shown in Table 9. For fuel cost, the best solution found by CMOQPSO is worse than that found by SMODE. However, the mean solution obtained by CMOQPSO is the best among all the 9 algorithms. For Emission, the best solution found by CMOQPSO is slightly worse than that found by MA θ-PSO and the mean solution obtained by CMOQPSO is better than those obtained by the other algorithms. So, on average CMOQPSO has obtained the best performance for both fuel cost and emission by 30 independent runs. It can be observed from Table 9 that the standard deviations obtained by CMOQPSO are the best for both fuel cost and emission. Which means, CMOQPSO is more stable than the other algorithms in this section.

Fig. 14 shows the comparison of Pareto fronts found by CMOQPSO and other algorithms and Table 10 gives the mean values of *SP* for EED problem with 40 generators. As Table 10 shows, CMOQPSO has the second best mean value of *SP*. In other words, the nondominated solutions found by CMOQPSO are spaced more equidistantly than those found by the other algorithms except MA θ-PSO, as shown in Fig. 14. Table 11 shows the comparison of the mean binary quality metric I_C between CMOQPSO and the other algorithms according to 30 independent runs. As mentioned in Section 4.1.2, if $I_C(A, B) = 1$ and $I_C(B, A) = 0$, then the nondominated solution set A is better than B. As can be seen from Table 11, the nondominated solutions found by CMOQPSO is obviously better than

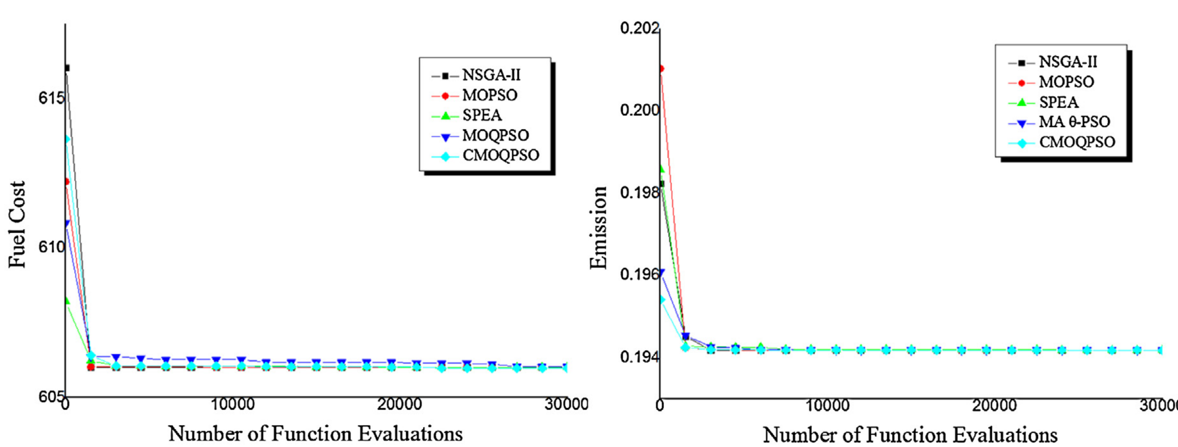

Fig. 13. Convergence characteristics of economic dispatch and emission dispatch for Case2.

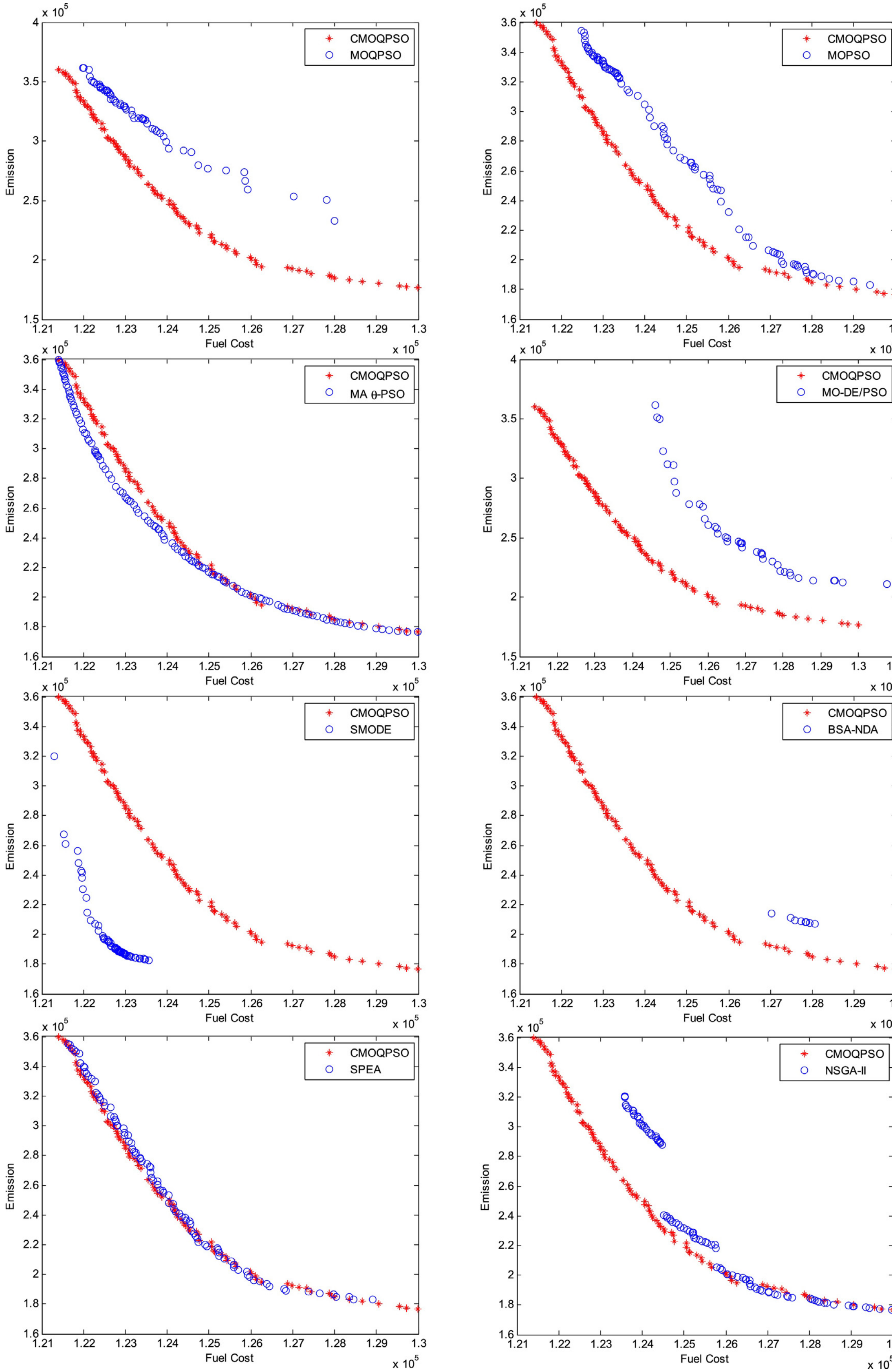

Fig. 14. Pareto Fronts found by algorithms.

Table 8
Best dispatch result of CMOQPSO for EED system with 40 generators.

	Best fuel cost				Best emission			
	P_1	110.81807	P_{21}	523.3834	P_1	114	P_{21}	439.2392
	P_2	110.8833	P_{22}	523.3563	P_2	114	P_{22}	439.9311
	P_3	97.4262	P_{23}	523.2405	P_3	120	P_{23}	439.7645
	P_4	179.7351	P_{24}	523.2291	P_4	169.3432	P_{24}	439.6137
	P_5	87.8221	P_{25}	523.2408	P_5	97	P_{25}	440.1445
	P_6	140	P_{26}	523.3673	P_6	124.2831	P_{26}	439.7974
	P_7	259.7139	P_{27}	10	P_7	299.8898	P_{27}	29.0320
	P_8	284.7269	P_{28}	10	P_8	298.0002	P_{28}	28.9942
	P_9	284.6685	P_{29}	10	P_9	297.2787	P_{29}	29.0036
	P_{10}	103.0302	P_{30}	87.7669	P_{10}	130	P_{30}	96.9911
	P_{11}	94	P_{31}	190	P_{11}	298.3530	P_{31}	172.2618
	P_{12}	94	P_{32}	189.9615	P_{12}	298.2319	P_{32}	172.2989
	P_{13}	394.3993	P_{33}	189.9743	P_{13}	433.5377	P_{33}	172.3391
	P_{14}	394.3993	P_{34}	164.8643	P_{14}	421.7557	P_{34}	200
	P_{15}	394.1226	P_{35}	194.3732	P_{15}	422.9475	P_{35}	200
	P_{16}	394.2335	P_{36}	200	P_{16}	422.7335	P_{36}	200
	P_{17}	489.2794	P_{37}	109.9812	P_{17}	439.4730	P_{37}	100.8536
	P_{18}	489.1483	P_{38}	109.9156	P_{18}	439.2154	P_{38}	100.9161
	P_{19}	511.2712	P_{39}	109.8678	P_{19}	439.4642	P_{39}	100.8023
	P_{20}	511.2389	P_{40}	511.2555	P_{20}	439.2905	P_{40}	439.2197
Fuel cost		121428.1853				129994.0844		
Emission		359876.1255				176683.7367		

Table 9
Statistical results of fuel cost and emission for EED system with 40 generators.

Algorithm	Fuel cost			Emission		
	Best	Mean	Std	Best	Mean	Std
CMOQPSO	121428.1853	**121487.1949**	**2.79e+01**	176683.7367	**176690.4569**	**1.27e+01**
MOQPSO	121989.7294	122155.6417	1.29e+02	228104.5694	240536.2612	7.36e+03
MOPSO	122490.6091	124020.4240	8.45e+02	177586.1960	186708.1733	4.3e+03
MA θ-PSO	121443.6804	121546.8803	8.05e+01	**176682.2634**	176733.4731	2.70e+02
MO-DE/PSO	124976.1084	125848.8533	8.00e+02	187367.8618	197624.1309	8.70e+03
SMODE	**121103.5240**	121493.8008	2.55e+02	176699.0174	192379.2038	1.84e+03
BSA-NDA	126609.7414	127376.3203	5.13e+02	203644.8492	205858.3131	1.64e+03
SPEA	121634.2033	121741.8602	5.38e+01	180910.3312	183552.0079	1.18e+03
NSGA-II	123574.1345	125146.1794	1.15e+03	176687.8707	177853.6674	4.83e+03

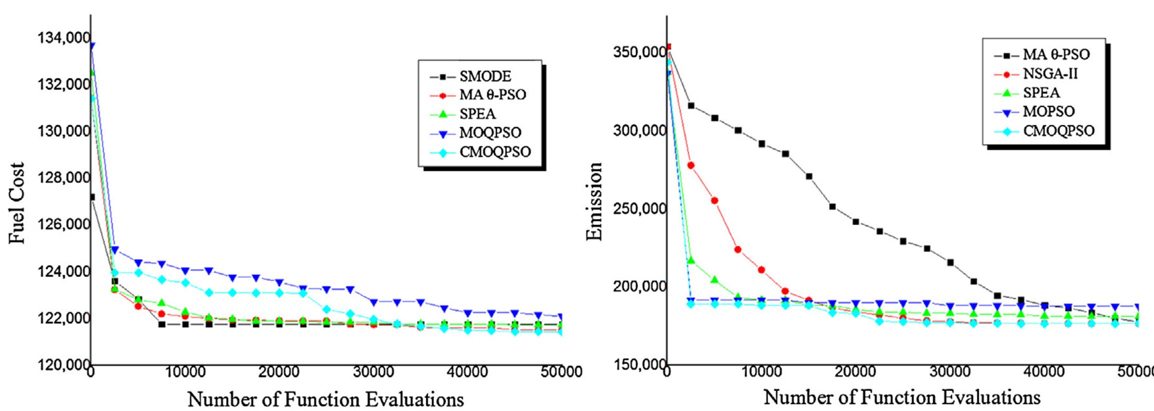

Fig. 15. Convergence characteristics of economic dispatch and emission dispatch for EED system with 40 generators.

Table 10
Comparison of *SP* of different algorithms.

Algorithm	Mean values of *SP* (1*e*+07)
CMOQPSO	0.047
MOQPSO	0.69
MOPSO	0.2
MA θ-PSO	0.038
MO-DE/PSO	0.82
SMODE	0.10
BSA-NDA	0.31
SPEA	0.067
NSGA-II	0.14

those found by MOQPSO, MOPSO, MO-DE/PSO and BSA-NDA. In a loose sense, the performance of CMOQPSO is better than SPEA and NSGA-II. However, the performance of CMOQPSO is worse than MA θ-PSO and SMODE.

The convergence characteristics for EED system with 40 generators are shown in Fig. 15. It can be observed that the top 5 algorithms for fuel cost in terms of mean values are SMODE, MA $\theta-$PSO, SPEA, MOQPSO and CMOQPSO. The top 5 algorithms for emission in terms of mean solutions are MA $\theta-$PSO, NSGA-II, SPEA, MOPSO and CMOQPSO. For both fuel cost and emission, CMOQPSO has not achieved the fastest convergence speed. This may because

Table 11
Comparison of $I_C(A, B)$ for different algorithms.

A:CMOQPSO		
B	$I_C(A, B)$	$I_C(B, A)$
MOQPSO	1	0
MOPSO	1	0
MA θ-PSO	0.067	0.88
MO-DE/PSO	1	0
SMODE	0.01	0.89
BSA-NDA	1	0
SPEA	0.67	0.21
NSGA-II	0.68	0.13

the multiple measurement strategy and the local search operator make the CMOQPSO has better searching ability, so that CMOQPSO can find good solutions in the searching space, which is also verified by the statistic results in Table 9.

5. Concluding remarks

In this paper, a multiobjective QPSO algorithm, called CMOQPSO, was adopted to solve EED problems. CMOQPSO has introduced cultural evolution mechanism into traditional QPSO to solve multiobjective optimization problems. In CMOQPSO, the evolution of the particles is guided by belief space which contains different types of knowledge extracted from the particle swarm. Another difference between CMOQPSO and the traditional QPSO is that each particle in CMOQPSO is measured for multiple times. For each particle, the global best positions of the multiple measurements are obtained according to the knowledge in belief space. Moreover, a local search operator is proposed to enhance the particles' ability of jumping out of local optima.

In order to show the effectiveness of the proposed algorithm, CMOQPSO is tested on EED systems with 6 and 40 generators respectively. The performance of CMOQPSO is compared with the other algorithms from different aspects. The comparative results have shown the effectiveness and superiority of CMOQPSO. However, in CMOQPSO each particle was measured for multiple times, this may lead to a relatively slower convergence speed in comparison with some other algorithms. This problem will be our future work of this paper.

References

[1] J. Talag, F. El-Hawary, M. El-Hawary, A summary of environmental/economic dispatch algorithms, IEEE Trans. Power Syst. 9 (3) (1994) 1508–1509.
[2] S. Brodesky, R. Hahn, Assessing the influence of power pools on emission constrained economic dispatch, IEEE Trans. Power Syst. PWRS 1 (1) (1986) 57–62.
[3] R. Yokoyama, S. Bae, T. Morita, H. Sasaki, Multiobjective generation dispatch based on probability security criteria, IEEE Trans. Power Syst. 3 (1) (1988) 317–324.
[4] V. Vahidinasab, S. Jadid, Joint economic and emission dispatch in energy markets: a multiobjective mathematical programming approach, Energy 35 (3) (2010) 1497–1504.
[5] J. Dhillon, S. Parti, D. Kothari, Stochastic economic emission load dispatch, Electr. Power Syst. Res. 26 (3) (1993) 186–197.
[6] C. Chang, K. Wong, B. Fan, Security-constrained multiobjective generation dispatch using bicriterion global optimization, IEE Proc. Gener. Transm. Distrib. 142 (4) (1995) 406–414.
[7] J. Xu, C. Chang, X. Wang, Constrained multiobjective global optimization of longitudinal interconnected power system by genetic algorithm, IEE Proc. Gener. Transm. Distrib. 143 (5) (1996) 435–446.
[8] C. Huang, H. Yang, C. Huang, Bi-objective power dispatch using fuzzy satisfaction-maximizing decision approach, IEEE Trans. Power Syst. 12 (4) (1997) 1715–1721.
[9] D. Das, C. Patvardhan, New multi-objective stochastic search technique for economic load dispatch, IEE Proc. Gener. Transm. Distrib. 145 (6) (1998) 747–752.
[10] L. Jiao, J. Liu, W. Zhong, An organizational coevolutionary algorithm for classification, IEEE Trans. Evol. Comput. 10 (1) (2006) 67–80.
[11] L. Jiao, Y. Li, M. Gong, X. Zhang, Quantum-inspired immune clonal algorithm for global optimization, IEEE Trans. Syst. Man Cybern. B Cybern. 38 (5) (2008) 1234–1253.
[12] R. Shang, L. Jiao, F. Liu, W. Ma, A novel immune clonal algorithm for MO problems, IEEE Trans. Evol. Comput. 16 (1) (2012) 35–50.
[13] J. Luo, L. Jiao, J. Lozano, A sparse spectral clustering framework via multi-objective evolutionary algorithm, IEEE Trans. Evol. Comput. 99 (2015) 1.
[14] M. Abido, A niched pareto genetic algorithm for multiobjective environmental/economic dispatch, Int. J. Electr. Power Energy Syst. 25 (2) (2003) 97–99.
[15] M. Abido, Multiobjective evolutionary algorithms for electric power dispatch problem, IEEE Trans. Evol. Comput. 10 (3) (2006) 315–329.
[16] M. Abido, A novel multiobjective evolutionary algorithm for environmental/economic power dispatch, Electr. Power Syst. Res. 65 (1) (2003) 71–91.
[17] M. Pandit, L. Srivastava, M. Sharma, Environmental economic dispatch in multi-area power system employing improved differential evolution with fuzzy selection, Appl. Soft Comput. 28 (2015) 498–510.
[18] H. Zhang, D. Yue, X. Xie, S. Hu, S. Weng, Multi-elite guide hybrid differential evolution with simulated annealing technique for dynamic economic emission dispatch, Appl. Soft Comput. 34 (2015) 312–323.
[19] M. Basu, Economic environmental dispatch using multi-objective differential evolution, Appl. Soft Comput. 11 (2) (2011) 2845–2853.
[20] Y. Li, H. He, Y. Wang, L. Jiao, An improved multiobjective estimation of distribution algorithm for environmental economic dispatch of hydrothermal power systems, Appl. Soft Comput. 28 (2015) 559–568.
[21] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of the International Conference on Neural Networks, 1995, pp. 1942–1948.
[22] Y. del Valle, G. Venayagamoorthy, S. Mohagheghi, J. Hernandez, R. Harley, Particle swarm optimization: basic concepts, variants and applications in power system, IEEE Trans. Evol. Comput. 12 (2) (2008) 171–195.
[23] N. Franken, A. Engelbrecht, Particle swarm optimization approaches to coevolve strategies for the iterated prisoner's dilemma, IEEE Trans. Evol. Comput. 9 (6) (2005) 562–579.
[24] J. Park, K. Lee, J. Shin, K. Lee, A particle swarm optimization for economic dispatch with non-smooth cost functions, IEEE Trans. Evol. Comput. 12 (2) (2008) 171–195.
[25] J. Chang, A robust adaptive array beamformer using particle swarm optimization for space time code division multiple access systems, Inf. Sci. 278 (2014) 174–186.
[26] A. Ouyang, K. Li, T.K. Truong, A. Sallam, E.H.-M. Sha, Hybrid particle swarm optimization for parameter estimation of Muskingum model, Neural Comput. & Applic. 25 (7–8) (2014) 1785–1799.
[27] A. Ouyang, Z. Tang, X. Zhou, Y. Xu, G. Pan, K. Li, Parallel hybrid PSO with CUDA for LD heat conduction equation, Comput. Fluids 110 (2015) 198–210.
[28] J. Zhao, M. Lin, D. Xu, L. Hao, W. Zhang, Vector control of a hybrid axial field flux-switching permanent magnet machine based on particle swarm optimization, IEEE Trans. Magn. 51 (2015).
[29] P. Regulski, D. Vilchis-Rodriguez, S. Djurovic, V. Terzija, Estimation of composite load model parameters using an improved particle swarm optimization method, IEEE Trans. Power Deliv. 30 (2) (2015) 553–560.
[30] F. Bergh, An Analysis of Particle Swarm Optimizers (Ph.D. thesis), University of Pretoria, 2001.
[31] F. Bergh, A. Engelbrecht, A new locally convergent particle swarm optimizer, IEEE Int. Conf. Syst. Man Cybern. 3 (2002).
[32] J. Sun, B. Feng, W. Xu, Particle swarm optimization with particles having quantum behavior, in: Proceedings of the 2004 Congress on Evolutionary Computation, CEC2004, 2004, pp. 326–331.
[33] J. Sun, X. Xu, V. Palade, W. Fang, C. Lai, W. Xu, Convergence analysis and improvements of quantum-behaved particle swarm optimization, Inf. Sci. 193 (15) (2012) 81–103.
[34] H. Al-Baity, S. Meshoul, A. Kaban, On extending quantum behaved particle swarm optimization to multiobjective context, in: 2012 IEEE Congress on Evolutionary Computation, CEC 2012, 2012, pp. 1–8.
[35] R. Reynolds, An introduction to cultural algorithms, in: Proceedings of the Third Annual Conference on Evolutionary Programming, 1994, pp. 131–139.
[36] T. Niknam, H. Doagou-Mojarrad, M. Nayeripour, A new fuzzy adaptive particle swarm optimization for non-smooth economic dispatch, Energy 35 (4) (2010) 1764–1778.
[37] M. Abido, Environmental/economic power dispatch using multiobjective evolutionary algorithms, IEEE Trans. Power Syst. 18 (4) (2003) 1529–1537.
[38] H. Saadat, Power System Analysis, The McGraw-Hill Companies, 1999.
[39] J. Sun, W. Xu, B. Feng, A global search strategy of quantum-behaved particle swarm optimization, in: 2004 IEEE Conference on Cybernetics and Intelligent Systems, 2004, pp. 111–116.
[40] J. Sun, W. Fang, X. Wu, V. Palade, W. Xu, Quantum-behaved particle swarm optimization: analysis of individual particle behavior and parameter selection, Evol. Comput. 20 (3) (2012) 349–393.
[41] M. Daneshyari, G. Yen, Cultural-based multiobjective particle swarm optimization, IEEE Trans. Syst. Man Cybern. B: Cybern. 41 (2) (2011) 553–567.
[42] N. Awad, M. Ali, R. Duwairi, Cultural algorithm with improved local search for optimization problems, in: 2013 IEEE Congress on Evolutionary Computation, CEC 2013, 2013, pp. 284–291.
[43] L. dos Santos Coelho, V. Mariani, An efficient particle swarm optimization approach based on cultural algorithm applied to mechanical design, in: 2006 IEEE Congress on Evolutionary Computation, CEC 2006, 2006, pp. 1099–1104.

[44] C. Coello, R. Becerra, Cultured differential evolution for constrained optimization, Comput. Methods Appl. Mech. Eng. 195 (33) (2006) 4303–4322.
[45] R. Reynolds, S. Zhu, Knowledge-based function optimization using fuzzy cultural algorithms with evolutionary programming, IEEE Trans. Syst. Man Cybern. B: Cybern. 31 (1) (2001) 1–18.
[46] C. Chen, S. Yang, Efficient de-based symbiotic cultural algorithm for neuro-fuzzy system design, Appl. Soft Comput. 34 (2015) 18–25.
[47] C. Coello, M. Coello, Lechuga, MOPSO: a proposal for multiple objective particle swarm optimization, in: 2002 IEEE Congress on Evolutionary Computation, CEC 2002, 2002, pp. 1051–1056.
[48] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.
[49] R. Goncalves, C. Almeida, M. Goldbarg, M. Delgado, Improved cultural immune systems to solve the economic load dispatch problems, in: 2013 IEEE Congress on Evolutionary Computation, CEC 2013, 2013, pp. 621–628.
[50] D. Gong, Y. Zhang, C. Qi, Environmental/economic power dispatch using a hybrid multi-objective optimization algorithm, Int. J. Electr. Power Energy Syst. 32 (6) (2010) 607–614.
[51] M. Abido, Multiobjective particle swarm optimization for environmental/economic dispatch problem, Electr. Power Syst. Res. 79 (7) (2009) 1105–1113.
[52] T. Niknam, H. Doagou-Mojarrad, Multiobjective economic/emission dispatch by multiobjective θ-particle swarm optimisation, IET Gener. Transm. Distrib. 6 (5) (2012) 363–377.
[53] B. Qu, J. Liang, Y. Zhu, Z. Wang, P. Suganthan, Economic emission dispatch problems with stochastic wind power using summation based multi-objective evolutionary algorithm, Inf. Sci. (2016).
[54] M.-D. Mostafa, R. Nasrudin Abd, Multi-objective backtracking search algorithm for economic emission dispatch problem, Appl. Soft Comput. 40 (2016) 479–494.
[55] R. King, H. Rughooputh, K. Deb, Evolutionary multiobjective environmental/economic dispatch: stochastic vs. deterministic approaches, in: Third International Conference on Evolutionary Multi-Criterion Optimization, EMO 2005. Proceedings (Lecture Notes in Computer Science), 2005, pp. 677–691.
[56] J. Schott, Fault tolerant design using single and multicriteria genetic algorithm optimization, No. AFIT/CI/CIA-95-039. AIR Force Inst of Tech Wright-Patterson AFB OH, 1995.
[57] E. Zitzler, L. Thiele, Multiobjective optimization using evolutionary algorithms – a comparative case study, in: 5th International Conference on Parallel Problem Solving from Nature – PPSN V, 1998, pp. 292–301.
[58] N. Sinha, R. Chakrabarti, P. Chattopadhyay, Evolutionary programming techniques for economic load dispatch, IEEE Trans. Evol. Comput. 7 (1) (2003) 83–94.