Hindawi
Complexity
Volume 2018, Article ID 1027193, 20 pages
https://doi.org/10.1155/2018/1027193

Research Article

Multiobjective Collective Decision Optimization Algorithm for Economic Emission Dispatch Problem

Xinlin Xu,[1] Zhongbo Hu ,[1] Qinghua Su ,[1] and Zenggang Xiong [2]

[1] *School of Information and Mathematics, Yangtze University, Jingzhou, Hubei, China*
[2] *School of Computer and Information Science, Hubei Engineering University, Xiaogan, Hubei, China*

Correspondence should be addressed to Zhongbo Hu; huzbdd@126.com and Zenggang Xiong; jkxxzg2003@163.com

Received 18 May 2018; Revised 4 October 2018; Accepted 30 October 2018; Published 13 November 2018

Academic Editor: Carlos F. Aguilar-Ibáñez

The collective decision optimization algorithm (CDOA) is a new stochastic population-based evolutionary algorithm which simulates the decision behavior of human. In this paper, a multiobjective collective decision optimization algorithm (MOCDOA) is first proposed to solve the environmental/economic dispatch (EED) problem. MOCDOA uses three novel learning strategies, that is, a leader-updating strategy based on the maximum distance of each solution in an external archive, a wise random perturbation strategy based on the sparse mark around a leader, and a geometric center-updating strategy based on an extreme point. The proposed three learning strategies benefit the improvement of the uniformity and the diversity of Pareto optimal solutions. Several experiments have been carried out on the IEEE 30-bus 6-unit test system and 10-unit test system to investigate the performance of MOCDOA. In terms of extreme solutions, compromise solution, and three metrics (SP, HV, and CM), MOCDOA is compared with other existing multiobjective optimization algorithms. It is demonstrated that MOCDOA can generate the well-distributed and the high-quality Pareto optimal solutions for the EED problem and has the potential to solve the multiobjective optimization problems of other power systems.

1. Introduction

The classical economic dispatch (ED) of electric power generation operating at the absolute minimum cost is one of the mathematical optimization issues attracting many researchers' interests [1, 2]. However, coal, natural gas, and petroleum remain the primary feedstock for power plants around the world, although some countries are planning to build several nuclear power plants and are sourcing more of electricity from wind. The serious global environmental problems are caused by burning fossil fuel to release several contaminants, such as SO_2, CO_2, and NO_x, into the atmosphere. The ED problem therefore no longer is considered alone. In this case, the environmental/economic dispatch (EED), a short-term alternative to reduce the atmospheric emissions, has received much attention.

The EED problem is a highly constrained conflicting nonlinear multiobjective optimization problem which results in not only greater economical benefits, but also less pollutant emissions. Various approaches have been reported to solve the multiobjective EED problem. Initially, some conventional optimization methods such as a linear programming method were used as the optimizing tool to optimize the EED problem [3, 4]. However, they are not suitable for a complex multiobjective optimization EED problem. Hence, some heuristic search techniques especially evolutionary algorithms (EAs) [5] and swarm intelligence algorithms (SIs) [6] have gotten the attention of many researchers' interests. EAs and SIs have been successfully tried to solve the EED problem, which can be mainly divided into two categories:

(i) The first category regards the multiobjective EED problem as a single-objective optimization problem. The EED problem was handled as a single-objective problem by considering the emission as a constraint in [7, 8]. Another technique, using the linear weighted sum method, transforms a set of objective functions into a single objective [9–14]. This approach generally uses the following formula to transform two objectives into a single objective:

$$\min \quad wF(P_G) + (1-w)\sigma E(P_G) \tag{1}$$

where σ is the scaling factor which combines the total fuel cost $F(P_G)$ with the total emission $E(P_G)$ and w is the weight factor in the range of $[0,1]$. Each objective is multiplied by a weight related to a given value w, which usually embodies the importance of the objective. The approach should be operated for several times to get a set of Pareto optimal solutions by setting different w values. It cannot be applied to obtain the Pareto optimal solutions for the problem with a nonconvex Pareto optimal front.

(ii) The second category tackles the both objectives, i.e., fuel cost and emission simultaneously as two competing objectives. Some multiobjective evolutionary algorithms based on genetic algorithms have been applied to generate the Pareto optimal solutions of the EED problem, which include a niched Pareto genetic algorithm (NPGA) [15], a nondominated sorting genetic algorithm (NSGA) [16], a strength Pareto evolutionary algorithm (SPEA) [17], an improved genetic algorithm (IGA) [18], and a fast and elitist Multiobjective Genetic Algorithm (NSGA-II) [19, 20], etc. Some other multiobjective evolutionary algorithms based on particle swarm optimization, such as an external memory based Multiobjective Particle Swarm Optimization (MOPSO) [21], a comprehensive learning particle swarm optimizer (MOCLPSO) [22], a fuzzified multiobjective particle swarm optimization (FMOPSO) [23], a multiobjective chaotic particle swarm optimization (MOCPSO) [24], a fuzzy clustering-based particle swarm optimization (FCPSO) [25], a parameter-free bare-bones multiobjective particle swarm optimization algorithm (BB-MOPSO) [26], and a cultural quantum-behaved particle swarm optimization (CMOQPSO) algorithm [27], have also been presented to solve the EED problem. In addition, differential evolutions (DEs) have also been used to solve the EED problem [28]. These algorithms have achieved good results for the EED problem.

In this paper, a new heuristic evolutionary method called a multiobjective collective decision optimization algorithm (MOCDOA) is first proposed to solve the EED problem. The paper mainly has the following four contributions:

(i) **Proposing a MOCDOA:** we adopt the collective decision optimization algorithm (CDOA) [29] to solve the multiobjective EED problem for the first time.

(ii) **Designing a leader-updating strategy:** considering the uniformity performance of approximate solutions, a new technique based on the maximum distance of each solution in the external archive is proposed to update a global leader.

(iii) **Designing a wise random perturbation strategy:** a wise random perturbation strategy based on the sparse mark around a leader is used to enhance the uniformity of the obtained Pareto optimal solutions.

(iv) **Designing a geometric center-updating strategy:** a geometric center-updating strategy is presented to expand the diversity performance of the Pareto optimal set, which randomly selects an extreme point on the Pareto optimal solutions to replace a geometric center.

The rest of this paper is organized as follows. Section 2 formulates the environmental/economic dispatch problem. Section 3 gives a brief review of CDOA. Section 4 presents MOCDOA. Section 5 shows the application of MOCDOA to solve the EED problem. Section 6 gives the corresponding comparative results of several existing optimization methods, and Section 7 concludes.

2. Mathematical Model of the EED Problem

Satisfying several equality and inequality constraints, the EED problem requires minimizing simultaneously two competing objective functions, the fuel cost and the emission. The mathematical model for the EED problem including its objective functions and constraints is described in Section 2.1 and Section 2.2 in detail. The EED problem can be mathematically formulated as

$$\begin{aligned} \min \quad & (F(P_G), E(P_G)) \\ \text{st.} \quad & \\ & h(P_G) = 0 \\ & g(p_G) \le 0 \end{aligned} \tag{2}$$

where F and E denote total fuel cost and total emission, respectively. h and g are the equality and inequality constraints, respectively.

2.1. Objective Functions

2.1.1. Fuel Cost Minimization. The total fuel cost $F(P_G)$ (dollars per hour) is made up of n generator costs expressed by quadratic functions. The total fuel cost $F(P_G)$ can be represented as

$$F(P_G) = \sum_{i=1}^{n} \left(a_i + b_i P_{Gi} + c_i P_{Gi}^2\right) \tag{3}$$

where the vector $P_G = (P_{G1}, P_{G2}, \cdots, P_{Gn})$ is the real power outputs of generators. P_{Gi} is the real power output of the ith generator. a_i, b_i, and c_i are the cost function coefficients of the ith generator.

2.1.2. Emission Minimization. The emission function can be presented as the sum of all types of the emission considered, but only the emission of nitrogen oxides NO_x is considered in the present study. The total emission $E(P_G)(ton/h)$ of nitrogen oxides NO_x caused by the generators can be expressed as a function of generator output in (4), that is, the sum of some quadratic functions and exponential functions.

Table 1: Generator cost and emission coefficients in the IEEE 30-bus 6-unit system.

Unit	p^{min}	p^{max}	a_i	b_i	c_i	α_i	β_i	γ_i	ζ_i	λ_i
G_1	0.05	0.5	10	200	100	4.091	−5.554	6.490	2.0*E* − 4	2.857
G_2	0.05	0.6	10	150	120	2.543	−6.047	5.638	5.0*E* − 4	3.333
G_3	0.05	1	20	180	40	4.258	−5.094	4.586	1.0*E* − 6	8.000
G_4	0.05	1.2	10	100	60	5.326	−3.550	3.380	2.0*E* − 3	2.000
G_5	0.05	1	20	180	40	4.258	−5.094	4.586	1.0*E* − 6	8.000
G_6	0.05	0.6	10	150	100	6.131	−5.555	5.151	1.0*E* − 5	6.667

$$E(P_G) = \sum_{i=1}^{n} \left(10^{-2} \left(\alpha_i + \beta_i P_{Gi} + \gamma_i P_{Gi}^2 \right) + \zeta_i e^{\lambda_i P_{Gi}} \right) \quad (4)$$

where α_i, β_i, γ_i, ζ_i, and λ_i are the emission function coefficients of the ith generator. The parameters α_i, β_i, γ_i, ζ_i, and λ_i are shown in Tables 1 and 3.

2.2. Constraints. The power dispatch constraints include one equality constraint on the power balance and several inequality constraints on generation capacity.

2.2.1. Power Balance Constraint. The total power generation must cover the power demand P_D and the transmission network loss P_L, namely,

$$\sum_{i=1}^{n} P_{Gi} = P_D + P_L \quad (5)$$

Here, in general, P_L is determined by Krons loss formula [30] as

$$P_L = \sum_{i=1}^{n} \sum_{j=1}^{n} P_{Gi} B_{ij} P_{Gj} + \sum_{i=1}^{n} B_{0i} P_{Gi} + B_{00} \quad (6)$$

where B_{ij}, B_{0i}, and B_{00} are the transmission network power loss coefficients.

2.2.2. Generation Capacity Constraints. The power output of each generator is limited by its corresponding lower and upper bounds as shown in

$$P_{Gi}^{\min} \le P_{Gi} \le P_{Gi}^{\max}, \quad i = 1, 2, \cdots, n. \quad (7)$$

where $P_{Gi}^{\min}$, $P_{Gi}^{\max}$ are the minimum and maximum generation limits of the ith generator, respectively.

3. Collective Decision Optimization Algorithm

Collective decision optimization algorithm (CDOA) [29] is a new evolutionary method proposed by Qingyang Zhang in 2016 [29], inspired from the decision-making behavior of human such as holding a meeting. Each member of a group will express and exchange their own thoughts or plans in meeting. The best one among the resultant plans is selected as a global optimal solution. In CDOA, an individual X_i generates several candidate solutions on a multistep position-selected scheme guided by different individuals, such as a local leader, an others' individual, a geometric center, and a leader. The seven major steps of CDOA, described briefly next, are group generation, experience-based phase, others-based phase, group thinking-based phase, leader-based phase, innovation-based phase, and selection.

Step 1 (group generation). CDOA uses N individuals as the group or population in each generation. The group and each individual at the tth iteration are denoted by $pop(t) = \{X_1(t), \ldots, X_i(t), \ldots, X_N(t)\}$ and $X_i(t) = \{x_i^1(t), x_i^2(t), \cdots, x_i^D(t)\}$, respectively, where D denotes the dimension of each generation. The group is initialized as shown in (8), so the N individuals are uniformly distributed in the feasible solution space.

$$x_{ik} \sim U(lx_k, ux_k) \quad (8)$$

where U is the uniform distribution and lx_k and ux_k are the lower and upper limits of the kth dimension, respectively.

Step 2 (experience-based phase). In CDOA, personal experience represents the local leader, i.e., the personal best position (Xb) obtained by each individual itself so far. The new position of the individual $X_i(t)$ will be expressed as follows:

$$newX_{i0} = X_i(t) + \overrightarrow{\mu_0} \cdot step(t) \cdot d_0, \quad d_0 = Xb - X_i(t) \quad (9)$$

where $\overrightarrow{\mu_0}$ is a random vector with each number uniformly distributed in the interval (0, 1) and $step(t)$ denotes the step size of the tth iteration.

$$step(t) = 2 - 1.7\left(\frac{t-1}{T_{\max} - 1}\right) \quad (10)$$

where $T_{\max}$ is the maximum number of iteration.

Step 3 (others'-based phase). In the meeting, all individuals will interact randomly. The individual $X_j(t)$ who is better than the current member $X_i(t)$ is randomly selected from the group. The calculation formula of updating the $newX_{i0}$ is defined as follows:

$$\begin{gathered} newX_{i1} = newX_{i0} + \overrightarrow{\mu_1} \cdot step(t) \cdot d_1, \\ d_1 = \alpha_1 \cdot d_0 + \beta_1 \cdot \left(X_j(t) - X_i(t)\right) \end{gathered} \quad (11)$$

where j denotes a random integer in [1, N], $\overrightarrow{\mu_1}$ is a random vector with each number uniformly distributed in (0, 1), and α_1 and β_1 are the random numbers in (-1, 1) and (0, 2), respectively.

Step 4 (group thinking-based phase). In the meeting, group thinking will influence the decision of each individual. In CDOA, the position of group thinking is represented by the geometric center (X_G) of all individuals. X_G can be described as follows:

$$\begin{aligned} X_G &= \frac{1}{N}\left(X_1(t), X_2(t), \cdots, X_N(t)\right) \\ &= \left(\frac{1}{N}\sum_{i=1}^{N} x_i^1(t), \frac{1}{N}\sum_{i=1}^{N} x_i^2(t), \cdots, \frac{1}{N}\sum_{i=1}^{N} x_i^N(t)\right) \end{aligned} \tag{12}$$

Here, updating the $newX_{i1}$ is calculated in the following formula:

$$\begin{aligned} newX_{i2} &= newX_{i1} + \overrightarrow{\mu_2} \cdot step(t) \cdot d_2, \\ d_2 &= \alpha_2 \cdot d_1 + \beta_2 \cdot \left(X_G(t) - X_i(t)\right) \end{aligned} \tag{13}$$

where $\overrightarrow{\mu_2}$ is a random vector with each number uniformly distributed in (0, 1) and α_2 and β_2 are the random numbers in (-1, 1) and (0, 2), respectively.

Step 5 (leader-based phase). In CDOA, the position of the best individual in the group is represented by a leader, i.e., a global best position (X_L). The next new position of ith individual will be designed as follows:

$$\begin{aligned} newX_{i3} &= newX_{i2} + \overrightarrow{\mu_3} \cdot step(t) \cdot d_3, \\ d_3 &= \alpha_3 \cdot d_2 + \beta_3 \cdot \left(X_L(t) - X_i(t)\right) \end{aligned} \tag{14}$$

where $\overrightarrow{\mu_3}$ is a random vector with each number uniformly distributed in (0, 1) and α_3 and β_3 are the random numbers in (-1, 1) and (0, 2), respectively.

The leader's mind can only be changed randomly by himself. The leader slightly changes its position by a random walk strategy in a local search space. In this phase, five neighbors are generated randomly around X_L, as shown in

$$newX_q = X_L + \overrightarrow{W_q}, \quad q = 1, 2, 3, 4, 5 \tag{15}$$

where $\overrightarrow{W_q}$ is a random vector with each number in $(0, 1)$. The next leader $newX_L$ is produced in

$$\begin{aligned} newX_L &= newX_k, \\ k &= min\_ObjFun\left(newX_q\right), \quad q = 1, 2, 3, 4, 5 \end{aligned} \tag{16}$$

where *min_ObjFun* is the index of a minimum objective function value.

Step 6 (innovation-based phase). To prevent a premature convergence, an innovation operator, which makes a small change among one of the dimension of each individual, is designed in CDOA. The innovation operator is similar to a mutation operator in evolutionary algorithms to improve the population diversity. The operator can be designed as follows:

$$newX_{i4} = \begin{cases} newX_{i3}, X_{i4}^p \sim U\left(lx_p, ux_p\right), & r0 < MR \\ newX_{i3}, & otherwise\,(t) \end{cases} \tag{17}$$

where $r0$ represents a random number in $(0, 1)$, p is a random integer in $[1, \mathrm{D}]$, and MR is an innovation (mutation) factor. The MR is set to 0.8 in the following algorithm.

Step 7 (selection). In the selection phase, the fitness values in X_i and X_{i4} are compared to update the population *pop* by using a greedy selection.

The procedure of CDOA again and again repeats from Step 2 to Step 7 until the terminating condition is satisfied.

4. Multiobjective Collective Decision Optimization Algorithm

CDOA is initially designed for single-objective optimization problems. In this paper, we extend CDOA to make it suitable for handling multiobjective optimization problems. Our main development of CDOA will be concerned in three novel learning strategies, which include a leader-updating strategy based on the maximum distance of each solution in nondominant solution set, a wise random perturbation strategy based on the sparse mark around the leader, and a geometric center-updating strategy based on an extreme point. The proposed three learning strategies are created for improving the uniformity and diversity of the Pareto optimal solutions. In addition, several existing techniques such as a local leader-updating strategy, a nondominated approach, an external elitist archive, and a circular crowded sorting are introduced into MOCDOA. Algorithm 1 presents the pseudocode of MOCDOA.

Similar to other evolutionary algorithms, MOCDOA can be divided into three processes, initialization, mutation, and selection. In the mutation phase of Algorithm 1, the main development of the three novel learning strategies is shown in the lines 11, 12, and 14, respectively, which are underlined and shown in bold. Whenever the archive goes beyond its capacity, the redundant crowded solutions will be removed from the archive in the lines 26-27 of Algorithm 1. The details of MOCDOA are described in this section.

4.1. Leader-Updating Strategy. In a multiobjective optimization problem, the conflicting multiple objectives make CDOA difficult to choose a global best position. To resolve this problem, MOCDOA maintains an external archive to store and update the nondominated solutions in each iteration. The leader X_L of each individual is selected from the external archive to improve the uniformity and diversity of the nondominated solutions. In MOCDOA, a new maximum distance (*md*) is designed to measure the sparsity of solutions. In Figure 1, the solid dots denote the solutions of the external archive. The maximum distance of the ith solution (md_i) is the maximum of two side-length sums of its two adjacent cuboids. The maximum distance of the ith solution (md_i) is calculated as follows:

$$md_i = \max\left(ud_i, ld_i\right) \quad i = 1, \ldots, Na \tag{18}$$

where Na is the maximum capacity of the archive. The distances of the ith solution to the upper point and the lower point are ud_i and ld_i shown in (19) and (20), respectively.

Input: input parameters $ObjFun, N, D, Na, T_{max}$
Output: Op
1: N: size of the population, D: dimension of the population, Na: maximum capacity of the archive, T_{max}: maximum number of iterations, $pop(t)$: population in tth iteration, $X_i(t)$: ith individual of population $pop(t)$ in tth iteration, Ar(t): external archive to store all the updated non-dominated solutions in tth iteration
2: **Initialization**
3: Generate an initial population $pop(0) = \{X_1(0), \cdots, X_N(0)\},\ where\ X_i(0) = \{x_i^1(0), \cdots, x_i^D(0)\}$
4: $X\mathrm{b} \longleftarrow pop(0)$ % Initialize the local leader Xb
5: $F(pop(0)) \longleftarrow ObjFun(pop(0))$ % Evaluate the fitness of each individual
6: $Ar(0) \longleftarrow Non\_dominated(pop(0))$ % Return the non-dominated solutions from pop^0
7: **while** $t \leq T_{max}$ **do**
8: **Mutation**
9: $Ar(t) \longleftarrow sort(Ar(t))$ % Sort Ar by objective functions value
10: **for** $i = 1\ to\ N$ **do**
11: $X_{Li}(t) \longleftarrow$ **Leader_updating**$(Ar(t))$ % Find the global best of each individual $X_{Li}(t)$
12: $X_{Gi}(t) \longleftarrow$ **Geometric_center_updating**$(Ar(t))$ % Find the geometric center $X_{Gi}(t)$
13: **if** $rand < 0.5$ **then**
14: $newpop_i(t) \longleftarrow$ **Leader_guiding**$(X_{Li}(t))$ % Directed search $X_{Li}(t)$ neighbor
15: **else**
16: $newpop_i(t) \longleftarrow Individual\_updating[newX_{i0}(t), newX_{i1}(t), newX_{i2}(t), newX_{i3}(t), newX_{i4}(t)]$ % Update the ith individual
17: **Selection**
18: $F(newpop(t)) = ObjFun(newpop(t))$ % Return the fitness value of each new population
19: **for** $i = 1$ to N **do**
20: **if** $newpop_i(t) \prec X_i(t)$ **then**
21: $X_i(t+1) = newpop_i(t)$
22: $F(X_i(t+1)) = F(newpop_i(t))$
23: **else if** $newpop_i(t) \nprec X_i(t)\ and\ X_i(t) \nprec newpop_i(t)$ **then**
24: Random access to the next generation
25: $Ar(t+1) \longleftarrow Non\_dominated(pop(t) \cup Ar(t))$ % Update the archive
26: **if** $|Ar(t+1)| > Na$ **then**
27: $Circular\_crowded\_sorting(Ar(t+1))$ % Maintain the archive, where $|Ar(t+1)|$ is the element number of $Ar(t+1)$
28: **for** $i = 1$ to N **do**
29: $Xb_i(t+1) = Local\_leader\_updating(Xb_i(t), X_i)$ % Update local leader
30: $t = t + 1$
31: $Op \longleftarrow Ar_t$ and stop the algorithm %Output the obtained Pareto optimal solutions

Algorithm 1: Multiobjective collective decision optimization algorithm.

$$ud_i = \sum_{j=1}^{M} \left| \frac{f_j(X_i) - f_j(X_{i-1})}{f_j^{\max} - f_j^{\min}} \right| \tag{19}$$

$$ld_i = \sum_{j=1}^{M} \left| \frac{f_j(X_i) - f_j(X_{i+1})}{f_j^{\max} - f_j^{\min}} \right| \tag{20}$$

where M is the number of objective functions. $f_j^{\min}$ and $f_j^{\max}$ are the maximum and minimum values of the jth objective function, respectively. $f_j(X_i)$ is the jth objective function value of the ith solution.

Since there is only one neighbor point for each extreme point in the archive, the maximum distance (md) of each extreme point is assigned with the distance between it and its neighbor point. In MOCDOA, the leader-updating strategy applies a roulette wheel selection method during each individual-mutating. The probability of each solution in the external archive to be selected as the leader is proportional to its maximum distance. We use Function Leader_updating to select X_L, which is shown in the line 11 of Algorithm 1. The leader-updating strategy is illustrated as Algorithm 2.

Here, we design a concept of sparse direction (l) for the wise random perturbation strategy in the next Section 4.3. The ud_i and ld_i of the ith solution of the archive are compared to record the sparse direction of the ith solution l_i, which will be used as a sparse mark of the solution to guide the perturbation in following strategy. If ld_i is greater than ud_i; that is, the gap of the ith solution for its upper neighbor solution is larger than that for its lower neighbor solution; we record $l_i = 1$. Otherwise, $l_i = -1$. For example, in Figure 1, ld_A is greater than ud_A for the Ath solution, so $l_A = 1$. For the Bth solution, ud_B is greater than ld_B, such that $l_B = -1$. The calculation formula of l_i is designed as

$$l_i = \begin{cases} 1, & ld_i > ud_i \\ -1, & ld_i \leq ud_i \end{cases} \tag{21}$$

```
Input: Ar(t)
Output: X_L
Nt = |Ar(t)|   % Nt is the element number of Ar(t)
for i = 1 to Nt do
    calculate md_i and l_i
for i = 1 to N do   % N is size of the population
    % X_Li(t) is selected according to the probability based on the percentage of the md_i as a whole
    index_i = Roulette_wheel_selection(md)
    X_Li(t) = Ar_index_i(t)
```

ALGORITHM 2: Leader-updating.

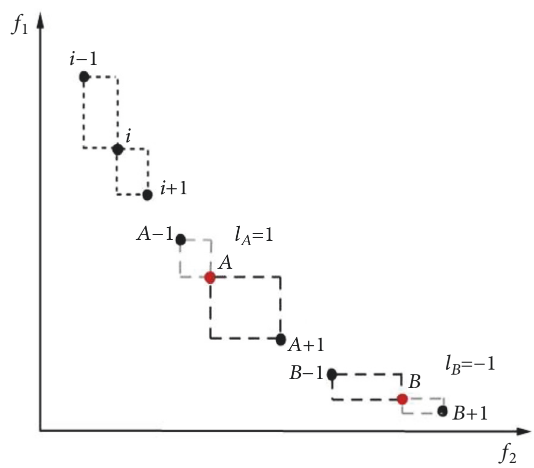


FIGURE 1: Calculation of maximum distance and sparse direction.

4.2. Wise Random Perturbation Strategy. To improve the uniformity performance of the external archive in MOCDOA, an individual that has the half probability to be selected is updated by using a wise random perturbation strategy around X_L. The perturbation method is shown in

$$newpop_i(t) = X_{Li}(t) + r1 \cdot \left(Ar_{index_i+l_i}(t) - X_{Li}(t)\right) \quad (22)$$

where $r1$ is a random number in $(0, 1)$. The strategy makes X_{Li} move to the direction of its sparse neighbor, thus generating a new individual $newpop_i$, which can reduce the sparsity of the external archive. We use Function Leader_guiding, shown in the line 14 of Algorithm 1, to update X_i.

4.3. Geometric Center-Updating Strategy. For improving the diversity performance of the external archive in MOCDOA, we use Function Geometric_center_updating to randomly select X_G in the extreme points of the archive. Function Geometric_center_updating is shown in the line 12 of Algorithm 1. The calculation formula of X_G is designed as

$$X_G(t) = \begin{cases} Ar_1(t), & r2 < 0.5 \\ Ar_{end}(t), & r2 \geq 0.5 \end{cases} \quad (23)$$

where $r2$ is a random number in $(0, 1)$.

4.4. Local Leader-Updating Strategy. The new personal best position, i.e., the new local leader, is updated according to the non_dominated relationship between the current individual X_i and the old local leader Xb_i in MOCDOA. If Xb dominates X_i, we keep Xb in memory. If Xb is dominated by X_i, X_i is selected as a new local leader to replace the old one. Otherwise, we randomly choice one of X_i and Xb as the new local leader. The local leader-updating shown in (24) is performed by Function Local_leader_updating in the line 29 of Algorithm 1.

$$Xb_i(t+1) = \begin{cases} Xb_i(t), & Xb_i(t) \prec X_i(t) \\ X_i(t), & X_i(t) \prec Xb_i(t) \\ Xb_i(t) \ \ or \ X_i(t), & Xb_i(t) \nprec X_i(t) \wedge X_i(t) \nprec Xb_i(t) \end{cases} \quad (24)$$

4.5. External Archive-Retaining Strategy. It is important to retain the nondominated solutions during the entire search process to obtain a good optimal solution set at the end of MOCDOA. Many scholars have used the external archive with a given maximal capacity to store and update the nondominated solutions in each iteration. At present, the maintenance strategy of an external archive is mostly adopted a more efficient nondominated sorting method called the fast nondominated sort in NSGA-II [31]. Furthermore, Deb et al. proposed an approach based on a crowding distance, which is usually the average distance of two neighbor points around each solution. Once the elitist archive has reached its maximal capacity, the crowding distance is adopted to remove the extra members and thereby keep the archive in its maximum capacity. Hence, the calculation method of the crowding distance greatly affects the distribution of the external archive. However, the crowding distance is only calculated once in each iteration of NSGA-II. Several adjacent solutions with small crowding distances will be removed, which may cause that the remaining solutions are too sparse. To overcome the above drawback, a cyclic crowded sorting algorithm [32], Function Circular_crowded_sorting in Algorithm 3, is adopted to improve the uniformity and the diversity of the Pareto optimal solutions.

5. Implementation of MOCDOA

In this section, the proposed MOCDOA is applied for solving the EED problem with one equality constraint on the power

Input: $Ar(t), Na$
Output: $Ar(t+1)$
$Nt = |Ar(t)|$ % Nt is the element number of $Ar(t)$
while $Nt > Na$ **do** % Na is maximum capacity of the archive
  for $i = 1$ *to* Nt **do**
    $Ar_i(t).distance = 0$ % Initialize the crowding distance of $Ar(t)$
  for $m = 1$ *to* M **do** % M is the number of objective functions
    $Ar(t) = sort(Ar(t), m)$ % Sort Ar by mth objective functions value
    $Ar_1(t).distance = Inf$ % Set the crowding distance of extreme solutions equal to infinite
    $Ar_{Nt}(t).distance = Inf$
    for i=2 to Nt-1 **do**
      $$Ar_i(t).distance = Ar_i(t).distance + \frac{Ar_{i+1}(t).distance - Ar_{i-1}(t).distance}{Ar_{Nt}(t).distance - Ar_1(t).distance}$$
  $k = min\_Ar(t).distance$ % Find the solution k with the smallest crowding distance
  $Ar_k(t) = [\,]$ % Delete the solution k from A
  $Nt = Nt - 1$ % Prepare for the next cycle
$Ar(t+1) \longleftarrow Ar(t)$

Algorithm 3: Cycled crowding sorting.

Input: X_i, P^{max}, P^{min}
Output: X_i''
X_i:ith individual,P^{max}: the upper limits of generation capacity, P^{min}: the lower limits of generation capacity
$e = P_L + P_D - sum(X_i)$ % Calculate and return the error between $P_L + P_D$ and the element sum of X_i
while $|e| > \sigma$ **do** % Check the feasibility of X_i
  $k = \text{rand}(1, N)$ % Pick a random integer between 1 to N
  $X'_{ik} = X_{ik}(P_D + P_L)/sum(X_{ik})$ % Adjust X_i to make it satisfy the constraint
  if $X'_{ik} > P_k^{max}$ *or* $X'_{ik} < P_k^{min}$ **then**
    $X'_{ik} = P_k^{min} + \text{rand} * \left(P_k^{max} - P_k^{min}\right)$
  $X_{ik} = X'_{ik}$
  $e = P_L + P_D - sum(X_i)$ % Calculate and return the error again

Algorithm 4: Constraint-handling.

balance. MOCDOA uses a constraint-handling mechanism to adjust an unfeasible solution in feasible search space and a fuzzy set theory to select a best compromise solution. Experiment design and parameter setting are introduced in final subsection.

5.1. Constraint-Handling. Since a resulting individual is not always guaranteed to satisfy the equality constraint, a constraint-handling strategy needs to be adopted to deal with the constrained EED problem. In order to guarantee the feasibility in all solutions, a straightforward constraint treatment method, the rejecting strategy, has been applied to handle the constraints of the EED problem in [23, 24, 33]. However, this approach produces the Pareto optimal solutions satisfying the equality constraints at the slowest pace.

In MOCDOA, Function Constrint_Handling is designed to handle the equality constraint of the EED problem. By applying Function Constrint_Handling, an unfeasible solution produced by MOCDOA can be modified into a feasible one. The Function Constrint_Handling is described in Algorithm 4. In this model, we set $\sigma = 1e - 12$.

5.2. Compromise Solution. After obtaining the Pareto optimal solutions, a fuzzy membership function [34] is proposed to simulate a decision-maker's preference and to extract a Pareto optimal solution as the best compromise solution. Usually, a membership function for each of the objective functions is defined by the experiences and intuitive knowledge of the decision-maker. In this work, a simple linear membership function is considered for each of the objective functions. The linear membership function is herein defined as

$$\mu_{ij} = \begin{cases} 1, & f_j(X_i) \le f_j^{\min} \\ \dfrac{f_j^{\max} - f_j(X_i)}{f_j^{\max} - f_j^{\min}}, & f_j^{\min} \le f_j(X_i) \le f_j^{\max} \\ 0, & f_j(X_i) \ge f_j^{\max} \end{cases} \quad (25)$$

The membership function value represents the degree of achievement of an objective function as a value between 0 and 1. $\mu_{ij} = 1$ is expressed as completely satisfactory, and $\mu_{ij} = 0$ is expressed as unsatisfactory. Figure 2 illustrates a fuzzy-based

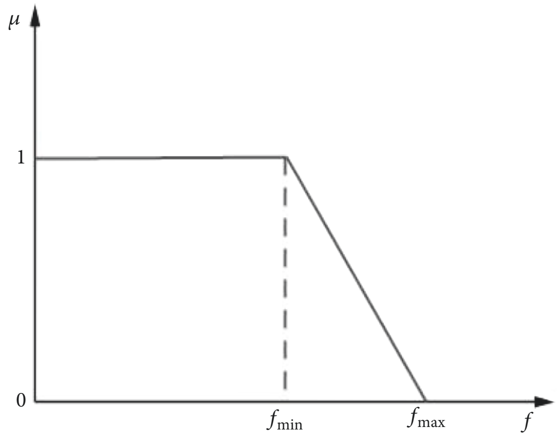


Figure 2: The fuzzy-based membership function.

membership function. The normalized membership function μ_k is calculated at the kth solution as follows:

$$\mu_k = \frac{\sum_{j=1}^{M} \mu_{kj}}{\sum_{i=1}^{Nt} \sum_{j=1}^{M} \mu_{ij}} \quad (26)$$

where M denotes the number of the objective functions and Nt is the number of the solutions in the final nondominated front. The best compromise solution is the one getting the highest value of μ_k.

5.3. Experiment Design and Parameter Setting. In this section, the standard IEEE 30-bus 6-unit system and 10-unit system are considered to validate the performance of the proposed MOCDOA for solving the EED problem. To evaluate the optimization performance of MOCDOA, two different cases (Case1 and Case2) of the EED problem are considered for the two test systems. In Case1, we do not consider the transmission loss of power balance constraint. On the contrary, we consider the transmission losses of power balance constraint for Case2. The data of the IEEE 30-bus 6-unit system are referenced in [26] and listed in Tables 1 and 2. The power demand of the IEEE 30-bus 6-unit system is set to 2.834 MW. The data of the IEEE 30-bus 10-unit system are referenced in [35] and listed in Tables 3 and 4. The power demand of the IEEE 30-bus 10-unit system is set to 2000 MW. We use Matlab software to run MOCDOA program on a personal computer with Pentium 2.60 GHz processor and 4.00GB RAM.

By using the orthogonal experimental method, the parameters N and Na are tuned and adjusted until the optimal settings are determined. In these two cases, the population size (N) and the maximum capacity of the archive (Na) are set as 100 and 50, respectively. The maximum iterations are restricted to 100 and 200 for Case1 and Case2, respectively. MOCDOA is set to conduct 30 runs to collect the statistical results for all the two test systems. For comparison, MOPSO [21], NSGA-II [19], and PESA-II [36] are applied to solve the EED problem and use the above same parameters. In case of MOPSO, the inertia weight, personal learning coefficient, global learning coefficient, and number of grids per dimension are selected as 0.7, 1.4, 1.4, and 7 for all the two test systems. In case of NSGA-II and PESA-II, the crossover percentage and mutation percentage are set to 0.7.

6. Experimental Results and Analysis

The first experimental results are to verify the effectiveness of the three learning strategies proposed in this paper. The second experimental results are obtained on the IEEE 30 bus 6-unit system. Further, the last experimental results are gained on the IEEE 30-bus 10-unit system.

6.1. Experimental Analysis of Three Learning Strategies. In order to verify that the uniformity and the diversity of the obtained Pareto optimal solutions are improved with three learning strategies, one type of multiobjective unconstrained test function is used. The function is as follows:

$$\begin{aligned} \min \quad & f_1 = 4x_1^2 + 4x_2^2 \\ \min \quad & f_2 = (x_1 - 5)^2 + 4(x_2 - 5)^2 \\ \text{st.} \quad & 0 < x_1 < 5, \\ & 0 < x_2 < 3 \end{aligned} \quad (27)$$

The parameters including $N = 40$, $D = 2$, $Na = 35$, and $T_{max} = 30$ are provided. The following four algorithms are compared for this test:

(1) The first one is a multiobjective CDOA (origin MOCDOA for short) obtained by transforming single-objective CDOA directly.

(2) The second one is an origin MOCDOA + strategy1, which is to add the leader-updating strategy into the origin MOCDOA.

(3) The third one is an origin MOCDOA + strategy1 + strategy2, which is to add the leader-updating strategy and the wise random perturbation strategy into the origin MOCDOA.

(4) The fourth one is an origin MOCDOA + strategy1 + strategy2 + strategy3, which is the algorithm proposed in this paper. It is to add the leader-updating strategy, the wise random perturbation strategy, and the geometric center-updating strategy into the origin MOCDOA.

The Pareto optimal solutions obtained by the four algorithms are depicted on Figure 3. According to the picture, we can get two results:

(1) As can be seen from Figure 3, a set of nondominant solutions can be found in all four methods. The Pareto front of the origin MOCDOA + strategy1 is more uniform than that of the origin MOCDOA, and the Pareto front of the origin MOCDOA + strategy1 + strategy2 is more uniform than that of the origin MOCDOA + strategy1. Furthermore, the origin MOCDOA + strategy1 and the origin MOCDOA + strategy1 + strategy2 have no multiple nondominant solutions converging in a small region. This shows that the strategy1 (the leader-updating strategy) and the strategy2 (the wise random perturbation strategy) can improve the uniformity of Pareto optimal solutions.

(2) As indicated by the first graph, the third graph, and the fourth graph in Figure 3, the coverage of the extreme solutions of the Pareto front for the origin MOCDOA +

Table 2: Transmission loss coefficients in the IEEE 30-bus 6-unit system.

B=	0.1382	−0.0299	0.0044	−0.0022	−0.0010	−0.0008	B_{00}=0.00098573
	−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	
B_0 =	0.0107	0.0060	−0.0017	0.0009	0.0002	0.0030	

Table 3: Generator cost and emission coefficients in the IEEE 30-bus 10-unit system.

Unit	p^{min}	p^{max}	a_i	b_i	c_i	d_i	e_i	α_i	β_i	γ_i	ζ_i	λ_i
G_1	10	55	1000.403	40.5407	0.12951	33	0.0174	360.0012	−3.9864	0.04702	0.25475	0.01234
G_2	20	80	950.606	39.5804	0.10908	25	0.0178	350.0056	−3.9524	0.04652	0.25475	0.01234
G_3	47	120	900.705	36.5104	0.12511	32	0.0162	330.0056	−3.9023	0.04652	0.25163	0.01215
G_4	20	130	800.705	39.5104	0.12111	30	0.0168	330.0056	−3.9023	0.04652	0.25163	0.01215
G_5	50	160	756.799	38.5390	0.15247	30	0.0148	13.8593	0.3277	0.00420	0.24970	0.01200
G_6	70	240	451.325	46.1592	0.10587	20	0.0163	13.8593	0.3277	0.00420	0.24970	0.01200
G_7	60	300	1243.531	38.3055	0.03546	20	0.0152	40.2669	−0.5455	0.00680	0.24800	0.01290
G_8	70	340	1049.998	40.3965	0.02803	30	0.0128	40.2669	−0.5455	0.00680	0.24990	0.01203
G_9	135	470	1658.569	36.3278	0.02111	60	0.0136	42.8955	−0.5112	0.00460	0.25470	0.01234
G_{10}	150	470	1356.659	38.2704	0.01799	40	0.0141	42.8955	−0.5112	0.00460	0.25470	0.01234

Table 4: Generator cost and emission coefficients in the IEEE 30-bus 10-unit system.

B=	0.000049	0.000014	0.000015	0.000015	0.000016	0.000017	0.000017	0.000018	0.000019	0.000020
	0.000014	0.000045	0.000016	0.000016	0.000017	0.000015	0.000015	0.000016	0.000018	0.000018
	0.000015	0.000016	0.000039	0.000010	0.000012	0.000012	0.000014	0.000014	0.000016	0.000016
	0.000015	0.000016	0.000010	0.000040	0.000014	0.000010	0.000011	0.000012	0.000014	0.000015
	0.000016	0.000017	0.000012	0.000014	0.000035	0.000011	0.000013	0.000013	0.000015	0.000016
	0.000017	0.000015	0.000012	0.000010	0.000011	0.000036	0.000012	0.000012	0.000014	0.000015
	0.000017	0.000015	0.000014	0.000011	0.000013	0.000012	0.000038	0.000016	0.000016	0.000018
	0.000018	0.000016	0.000014	0.000012	0.000013	0.000012	0.000016	0.000040	0.000015	0.000016
	0.000019	0.000018	0.000016	0.000014	0.000015	0.000014	0.000016	0.000015	0.000042	0.000019
	0.000020	0.000018	0.000016	0.000015	0.000016	0.000015	0.000018	0.000016	0.000019	0.000044
B_0 =	0		B_{00} =	0						

strategy1 + strategy2 marked by two green diamonds is more widespread than that of the origin MOCDOA marked by two blue squares. The coverage of the extreme solutions of the Pareto front for the origin MOCDOA + strategy1 + strategy2 + strategy3 marked by two red circles is more widespread than that of the origin MOCDOA + strategy1 + strategy2. Based on the above, one can draw a conclusion that the strategy3 (the geometric center-updating strategy) can increase the diversity of the Pareto optimal solutions, and combining the strategy3 with the other two strategies does not destroy the diversity.

6.2. IEEE 30-Bus 6-Unit System. All the experiments in this section are carried out for IEEE 30-bus 6-unit system. The first experiment is performed to evaluate the MOCDOA's performance by comparing the results of the extreme solutions and the compromise solutions in all cases for MOCDOA and other algorithms. The second experiment is carried out to evaluate the solution quality by comparing three metrics, i.e., SP, HV, and CM for four algorithms. The third experiment is implemented to analyze the robustness of MOCDOA by comparing the statistical results for the solutions of the minimal fuel cost, the minimal emission, and the ASD of compromise solution.

6.2.1. Comparison of Extreme Solutions and Compromise Solutions. Initially, the basic CDOA is implemented to optimize the fuel cost and the emission individually in order to explore the extreme points of the trade-off surface in all cases. The obtained best results are given in Table 5. The convergence of the fuel cost and the emission objectives for Case1 and Case2 are shown in Figure 4. CDOA with the cost as only objective function obtains the optimal values 600.111408 $/h and 605.998370 $/h for Case1 and Case2, respectively. The optimal values of CDOA with the emission as only objective function are 0.194203 ton/h and 0.194179

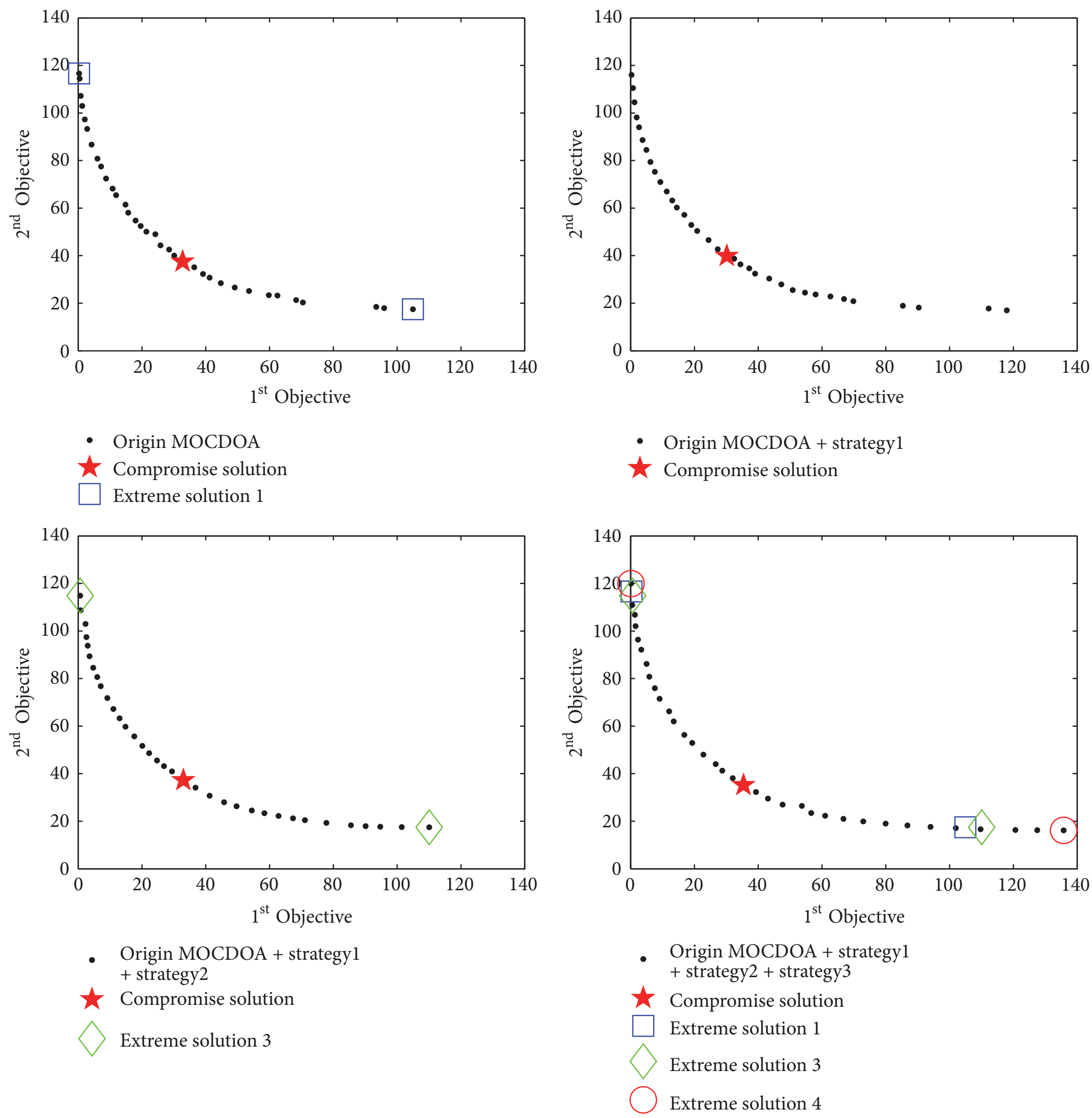


Figure 3: Comparisons of original MOCDOA and adding strategies MOCDOA for test1.

Table 5: IEEE 30-bus 6-unit system best solutions for cost and emission optimized individually.

	Case1		Case2	
	Best cost	Best emission	Best cost	Best emission
G_1	0.109712	0.406081	0.120952	0.410946
G_2	0.299772	0.459067	0.286307	0.463662
G_3	0.524300	0.537945	0.583597	0.544409
G_4	1.016191	0.382950	0.992842	0.390386
G_5	0.524308	0.537930	0.523967	0.544445
G_6	0.359717	0.510024	0.351894	0.515483
Fuel cost ($/h)	**600.111408**	638.273933	**605.998370**	646.207369
Emission (ton/h)	0.222145	**0.194203**	0.220730	**0.194179**

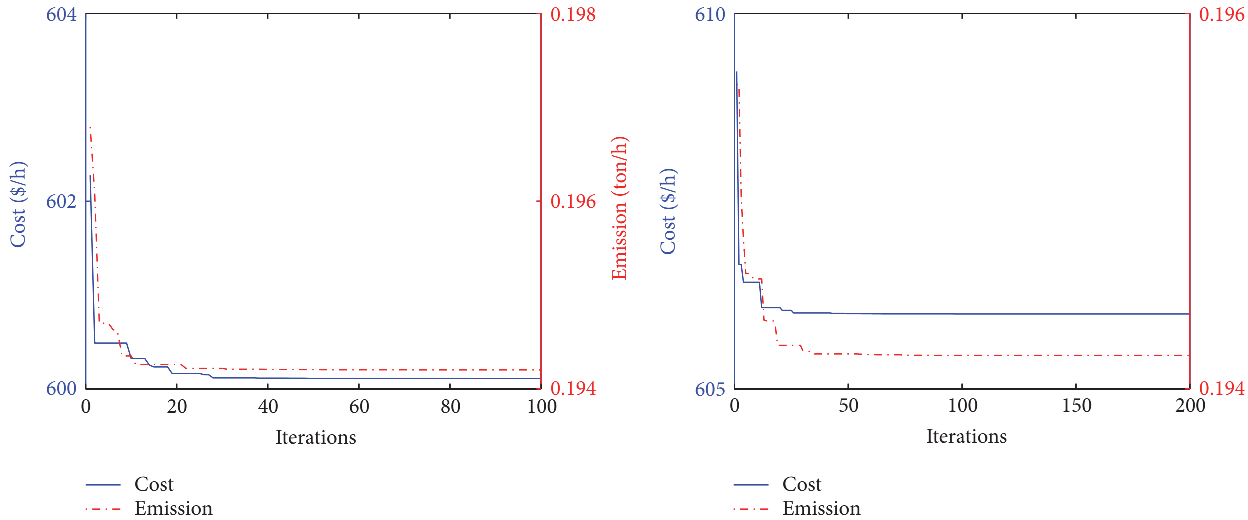


Figure 4: IEEE 30-bus 6-unit system convergence of cost and emission objective functions on Case1 and Case2.

Table 6: IEEE 30-bus 6-unit system best solutions for cost with nine algorithms on Case1.

	MOCDOA	LP	MOSST	NSGA	NPGA	SPEA	NSGA-II	FCPSO	BB-MOPSO
G_1	0.110009	0.1500	0.1125	0.1567	0.1080	0.1062	0.1059	0.1070	0.1090
G_2	0.299387	0.3000	0.3020	0.2870	0.3284	0.2897	0.3177	0.2897	0.3005
G_3	0.521287	0.5500	0.5311	0.4671	0.5386	0.5289	0.5216	0.525	0.5234
G_4	1.019763	1.0500	1.0208	1.0467	1.0067	1.0025	1.0146	1.015	1.0170
G_5	0.523732	0.4600	0.5311	0.5037	0.4949	0.5402	0.5159	0.5300	0.5238
G_6	0.359821	0.3500	0.3625	0.3729	0.3574	0.3664	0.3583	0.3673	0.3603
Emission	0.222376	0.22330	0.22220	0.22282	0.22116	0.2215	0.22188	0.2223	0.22220
Fuel cost	**600.112351**	606.314	605.889	600.572	600.259	600.15	600.155	600.1315	**600.112**
$\left\|\sum_{i=1}^{N} P_{Gi} - P_D\right\|$	**0**	0.026	0.2026	$1e-04$	**0**	$1e-04$	**0**	**0**	**0**
FEs	**10,000**	-	-	100,000	100,000	100,000	**10,000**	20,000	**10,000**

ton/h for Case1 and Case2, respectively. Next, the results of the multiobjective extreme solutions and the compromise solutions are discussed. The best results in all the tables are highlighted in bold for each case.

For Case 1, an experiment is performed to search for the extreme solutions and the compromise solution on the Pareto optimal set. The distribution of the nondominated solutions in Pareto front is displayed in Figure 5, which indicates clearly that these found solutions are almost well distributed on the entire Pareto front of Case1. The best results for the fuel cost and the emission (the extreme points on the Pareto front) obtained by MOCDOA are compared with those obtained by linear programming(LP)[4], multiobjective stochastic search technique(MOSST) [37], NSGA [16], NPGA [16], SPEA [38], NSGA-II [20], FCPSO [25], and BB-MOPSO [26] in Tables 6 and 7. The average satisfactory degree (ASD) [26] of the decision-maker for the compromise solution of MOCDOA is then calculated. The results of the compromise solutions and ASDs of MOCDOA, BB-MOPSO, NSGA, NPGA, SPEA, and FCPSO are shown in Table 8. Indeed, we use ASDs to compare the compromise solutions of the different algorithms.

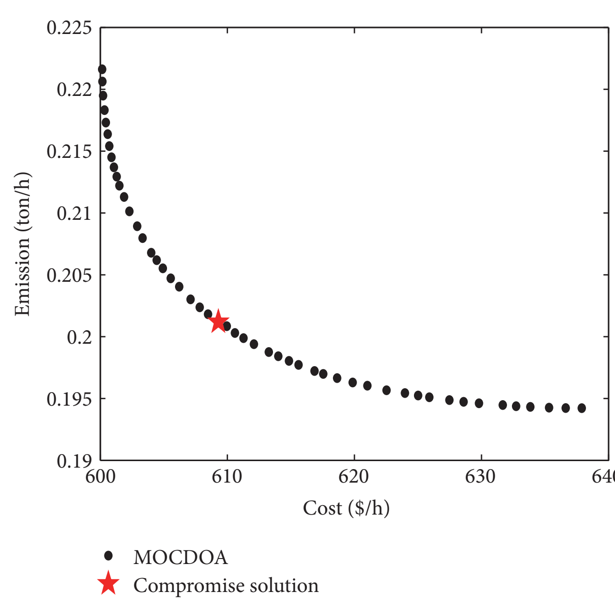


Figure 5: IEEE 30-bus 6-unit system Pareto front using MOCDOA on Case1.

Table 7: IEEE 30-bus 6-unit system best solutions for emission with nine algorithms on Case1.

	MOCDOA	LP	MOSST	NSGA	NPGA	SPEA	NSGA-II	FCPSO	BB-MOPSO
G_1	0.404380	0.4000	0.4095	0.4394	0.4002	0.4116	0.4074	0.4097	0.4071
G_2	0.461118	0.5500	0.4626	0.4511	0.4474	0.4532	0.4577	0.4550	0.4591
G_3	0.537554	0.4500	0.5426	0.5105	0.5166	0.5329	0.5389	0.5363	0.5374
G_4	0.382199	0.4000	0.3884	0.3871	0.3688	0.3832	0.3837	0.3842	0.3838
G_5	0.538118	0.5500	0.5427	0.5553	0.5751	0.5383	0.5352	0.5348	0.5369
G_6	0.510631	0.5000	0.5152	0.4905	0.5259	0.5148	0.5110	0.5140	0.5098
Fuel cost	638.327501	639.600	644.112	639.231	639.182	638.51	638.269	638.3577	638.262
Emission	0.194203	0.19424	**0.19418**	0.19436	0.19433	0.1942	0.19420	0.1942	0.194203
$\left\lvert\sum_{i=1}^{N} P_{Gi} - P_D\right\rvert$	**0**	0.016	0.027	$1e-04$	**0**	**0**	$1e-04$	**0**	**0**
FEs	**10,000**	-	-	100,000	100,000	100,000	**10,000**	20,000	**10,000**

Table 8: IEEE 30-bus 6-unit system best compromise solutions with six algorithms on Case1.

	G_1	G_2	G_3	G_4	G_5	G_6	Fuel cost	Emission	ASD
MOCDOA	0.269923	0.372134	0.529157	0.699728	0.546824	0.416234	609.656267	0.200942	**0.759457**
BB-MOPSO	0.2595	0.3698	0.5351	0.6919	0.5500	0.4277	609.747	0.20083	0.7555
NSGA	0.2571	0.3774	0.5381	0.6872	0.5404	0.4337	610.067	0.20060	0.7551
NPGA	0.2696	0.3673	0.5594	0.6496	0.5396	0.4486	612.127	0.19941	0.7491
SPEA	0.2785	0.3764	0.5300	0.6931	0.5406	0.4153	610.254	0.20055	0.7527
FCPSO	0.3193	0.3934	0.5359	0.5921	0.5457	0.447	619.998	0.19715	0.7267

Table 9: IEEE 30-bus 6-unit system best solutions for cost with six algorithms on Case2.

	G_1	G_2	G_3	G_4	G_5	G_6	Fuel cost	Emission	FEs
MOCDOA	0.120436	0.287096	0.580268	0.992202	0.525923	0.353730	**605.999549**	0.220643	20,000
NSGA	0.1168	0.3165	0.5441	0.9447	0.5498	0.3964	608.245	0.21664	100,000
NPGA	0.1245	0.2792	0.6284	1.0264	0.4693	0.3993	608.147	0.22364	100,000
SPEA	0.1086	0.3056	0.5818	0.9846	0.5288	0.3584	607.807	0.22015	100,000
NSGA-II	0.1182	0.3148	0.5910	0.9710	0.5172	0.3548	607.801	0.21891	10,000
FCPSO	0.1130	0.3145	0.5826	0.9860	0.5264	0.3450	607.7862	0.2201	20,000

From Table 6, it is quite evident that the proposed MOCDOA performs better than LP, MOSST, NSGA, NPGA, SPEA, NSGA-II, and FCPSO and almost as the same as BB-MOPSO in terms of the minimum fuel cost. Moreover, MOCDOA outperforms NSGA, NPGA, SPEA, and FCPSO and as the same as NSGA-II and BB-MOPSO in terms of the lowest fitness function evaluations (FEs). MOCDOA performs better than LP, MOSST, NSGA, and SPEA and almost as the same as NPGA, NSGA-II, FCPSO, and BB-MOPSO in terms of the error of equality constraint to obtain a Pareto front equal to 0. In Table 7, the minimum emission and the error of equality constraint in MOCDOA are equal to 0.194203 ton/h and 0, respectively, by using 10,000 FEs. MOCDOA has the lowest error of equality constraint and the minimal FEs for all the compared algorithms. Although MOSST has the minimum emission equal to 0.19418 ton/h, its error of equality constraint is the largest one equal to 0.027. MOCDOA's emission is close to the minimum emission equal to 0.1942 ton/h while the condition of zero error of equality constraint is satisfied. In addition, MOCDOA provides a higher ASD of compromise solution than those of the other five algorithms as shown in Table 8. As seen from the above discussions, MOCDOA is more efficient than almost all the other compared algorithms.

For Case2, this problem has been solved by using NSGA and NPGA in [16], SPEA in [38], NSGA-II in [20], and FCPSO in [25]. The minimum fuel cost and the minimum emission for MOCDOA and these five algorithms are presented in Tables 9 and 10.

From Tables 9 and 10, MOCDOA obtains the minimum fuel cost and the minimum emission equal to 605.999549 \$/h and 0.194179 ton/h, respectively, by using 20,000 FEs, which is lower than those of other five algorithms. Except NSGA-II with the lowest FEs 10,000, MOCDOA outperforms the other 4 algorithms in terms of FEs. Thus, the above results illustrate the stronger competitiveness of MOCDOA than other algorithms for the best solutions and the less computational time.

6.2.2. Comparison of Solution Quality. Unlike single-objective optimization problems, the evaluation of solution quality of a multiobjective optimization problem is substantially more complex. The following criteria are generally considered

Table 10: IEEE 30-bus 6-unit system best solutions for emission with six algorithms on Case2.

	G_1	G_2	G_3	G_4	G_5	G_6	Fuel cost	Emission	FEs
MOCDOA	0.410372	0.466870	0.543728	0.390095	0.542564	0.515742	646.334746	**0.194179**	20,000
NSGA	0.4113	0.4591	0.5117	0.3724	0.5810	0.5304	647.251	0.19432	100,000
NPGA	0.3923	0.4700	0.5565	0.3695	0.5599	0.5163	645.984	0.19424	100,000
SPEA	0.4043	0.4525	0.5525	0.4079	0.5468	0.5005	642.603	0.19422	100,000
NSGA-II	0.4141	0.4602	0.5429	0.4011	0.5422	0.5045	644.133	0.19419	10,000
FCPSO	0.4063	0.4286	0.5510	0.4084	0.5432	0.4974	642.8964	0.1942	20,000

Table 11: IEEE 30-bus 6-unit system statistical results of the SP on Case2.

	Best	Worst	Median	Average	Std
MOCDOA	**0.003844**	**0.007615**	**0.005605**	**0.005636**	**0.000850**
MOPSO	0.012307	0.023796	0.018037	0.018255	0.002701
NSGA-II	0.015961	0.023478	0.020096	0.020139	0.002054
PESA-II	0.014728	0.056674	0.023672	0.026567	0.008761

to evaluate the solution quality for multiobjective optimization problems [39].

(i) **Uniformity.** Maintaining uniform distribution.

(ii) **Diversity.** Maximum the distribution extent of the obtained nondominated set.

(iii) **Convergence.** Minimum the distance of the obtained Pareto optimal set and the true Pareto optimal front.

To evaluate MOCDOA's performance on the solution quality, three well-known algorithms including MOPSO [21], NSGA-II [19], and PESA-II [36] are selected and compared with MOCDOA. Here, we set all algorithms to have the same population size, archive size and maximum iteration. The constraint-handling strategy proposed in Section 5.2 is adopted into the four algorithms. The results of different algorithms are compared in terms of the above three criteria.

For comparing the uniformity of Pareto optimal solutions, the spacing metric (SP) [40] is adopted to measure the uniformity of the obtained nondominated solutions. The calculation of the SP is as follows:

$$SP = \sqrt{\frac{1}{|Ar| - 1} \sum_{i=1}^{|Ar|} \left(\overline{d} - d_i\right)^2},$$
$$d_i = \min_{q_j \in Ar \wedge q_j \neq q_i} \sum_{k=1}^{m} \left| f_k\left(q_i\right) - f_k\left(q_j\right) \right| \tag{28}$$

where d_i refers to the Euclidean distance of two consecutive solutions in the external archive and $\overline{d}$ is the mean of all d_i. The smaller the SP value is, the more uniform the distribution of solutions on the obtained Pareto front is. SP=0 represents that all solutions of the obtained Pareto front are equidistantly spaced. Table 11 illustrates the comparison results of SP for different algorithms. It can be seen from this table that the average performance of MOCDOA is far better than those of the other algorithms and the standard deviation of MOCDOA is smallest. In order to compare intuitively the uniformity of the solutions obtained by MOCDOA and the other algorithms, the Pareto fronts of MOCDOA, MOPSO, NSGA-II, and PESA-II are depicted together, as shown in Figure 6. It can be deduced from Figure 6 that the distribution of MOCDOA shows an advantage over the other three algorithms. The above discussions confirm that MOCDOA has a better uniformity performance than the other three algorithms.

A performance metric of the convergence and the diversity of the Pareto optimal solutions, hypervolume (HV) [41], was proposed by Zitler and Thiele. The metric calculates the volume covered by all the solutions of a nondominated set and a given reference point. The HV is defined as follows:

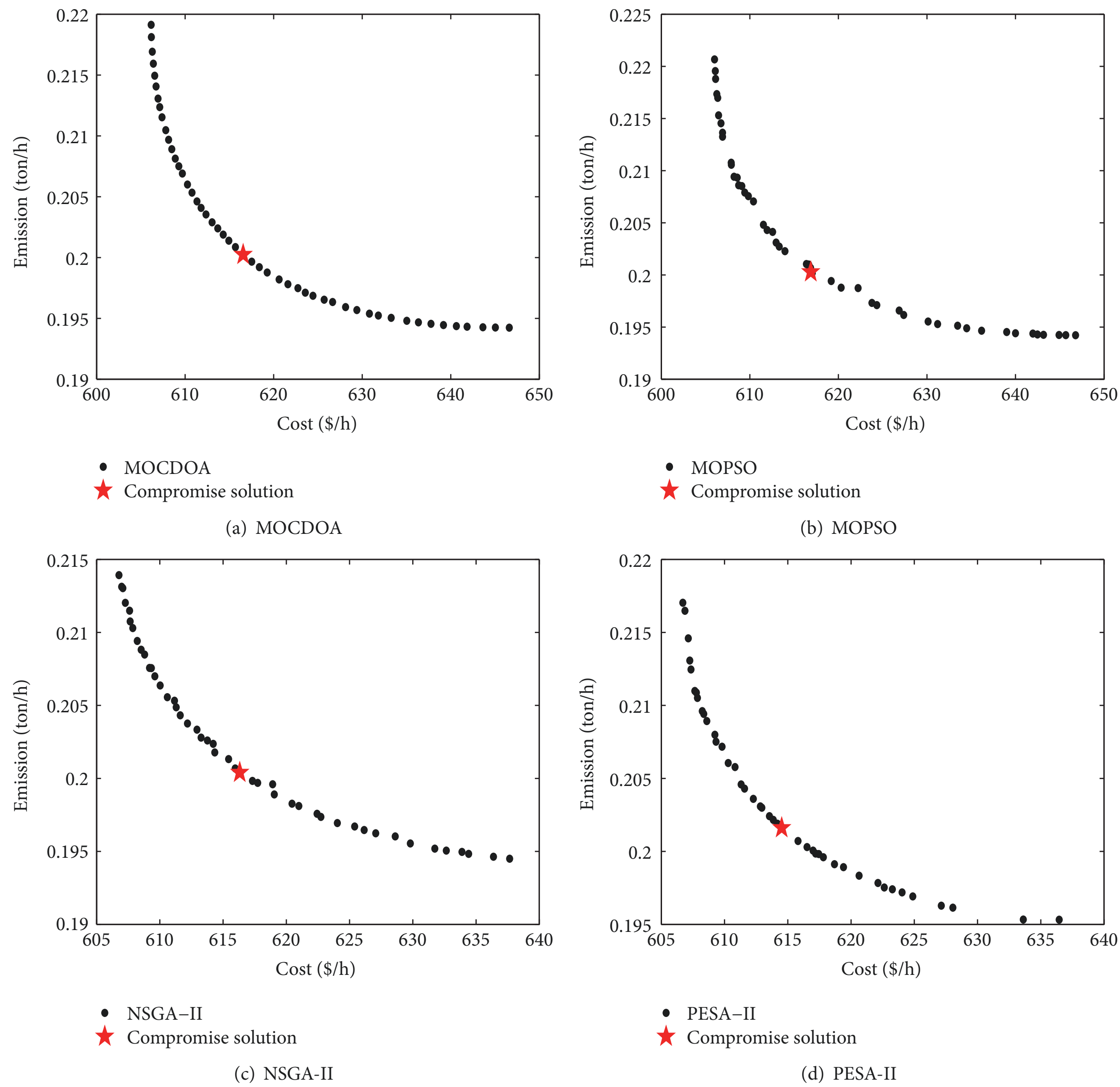
$$HV = \bigcup_{i=1}^{|Ar|} v_i \tag{29}$$

where a hypercube volume v_i is calculated with a reference point w_r and a solution $X_i \in Ar$ as the diagonal corners of a hypercube. For HV, a higher value is better. The reference point w_r will affect the calculation of HV. In our experiments, a same reference point is used for all the algorithms. The statistical results of HV are compared in Table 12 among four different algorithms. From Table 12, it can be seen that the proposed MOCDOA obtains the largest HV values, which means that MOCDOA has a better convergence and a diversity performance than these of MOPSO, NSGA-II, and PESA-II.

To evaluate the quality of the obtained Pareto optimal solutions of the optimized problem with the unknown true Pareto front, C-metric (CM) [42] is quite often used. It can be used to show the dominance relationship between two different algorithms. CM is defined as follows:

$$CM\left(Ar_1, Ar_2\right) = \frac{\left|\left\{x_2 \in Ar_2,\ \exists x_1 \in Ar_1 : x_1 \prec x_2\right\}\right|}{\left|Ar_2\right|} \tag{30}$$

where Ar_1 and Ar_2 are two solution sets of two different algorithms. $CM(Ar_1, Ar_2) = 1$ indicates that all solutions in Ar_2 are dominated by the solutions in Ar_1. This shows that Ar_1 is closer to the true Pareto optimal front than Ar_2.



(a) MOCDOA (b) MOPSO (c) NSGA-II (d) PESA-II

Figure 6: IEEE 30-bus 6-unit system Pareto fronts and compromise solution for the four algorithms on Case2.

Table 12: IEEE 30-bus 6-unit system statistical results of the HV on Case2.

	Best	Worst	Median	Average	Std
MOCDOA	**1.174740**	**1.172396**	**1.173860**	**1.173755**	**0.000630**
MOPSO	1.162428	1.155299	1.159138	1.158993	0.002064
NSGA-II	1.167940	1.146496	1.162377	1.160582	0.005742
PESA-II	1.158865	1.120439	1.151565	1.148653	0.009073

$CM(Ar_1, Ar_2) = 0$ represents that none solution in Ar_2 is covered by Ar_1. Table 13 shows the comparison results of CM produced by the best solutions of different algorithms. From this table, it can be seen that none solution in MOCDOA is dominated by that of MOPSO, and near 4% and 2% solutions of MOCDOA are dominated by those of NSGA-II and PESA-II, respectively. Moreover, MOPSO, NSGA-II, and PESA-II have 60%, 18%, and 16% solutions to be dominated by MOCDOA, respectively. Thus, MOCDOA has a better convergence performance than the three algorithms.

From the above analysis, it can be concluded that MOCDOA has a better performance of uniformity, diversity and convergence than those of the other algorithms in Case2. More specifically, the better uniformity performance of MOCDOA attributes to the common efforts of the leader-updating strategy in Section 4.1 and the wise random

Table 13: IEEE 30-bus 6-unit system statistical results of the CM on Case2.

	MOCDOA	MOPSO	NSGA-II	PESA-II
CM(MOCDOA,∗)	-	0.6	0.18	0.16
CM(MOPSO,∗)	**0**	-	0.6	0.1
CM(NSGA-II,∗)	**0.04**	0.38	-	0.16
CM(PESA-II,∗)	**0.02**	0.36	0.12	-

Table 14: IEEE 30-bus 6-unit system statistical results of the three objectives on Case1.

	Best	Worst	Median	Average	Std
Fuel cost	600.112572	600.277057	600.138081	600.159425	**0.041390**
Emission	0.194203	0.194325	0.194232	0.194243	$\mathbf{3.33}\boldsymbol{e}\mathbf{-05}$
ASD	0.759457	0.747940	0.754129	0.753532	**0.002742**

Table 15: IEEE 30-bus 6-unit system statistical results of the three objectives on Case2.

	Best	Worst	Median	Average	Std
Fuel cost	605.999549	606.1162352	606.015211	606.024775	**0.026281**
Emission	0.194179	0.194249	0.194188	0.194197	$\mathbf{1.98}\boldsymbol{e}\mathbf{-05}$
ASD	0.758042	0.744444	0.754744	0.753989	**0.003077**

perturbation strategy in Section 4.2. The better diversity performance of MOCDOA attributes to the effort of the geometric center-updating strategy in Section 4.3. The better convergence performance of MOCDOA attributes to the effort of a lot of random variables in the original CDOA, which results in the fact that the population of MOCDOA is able to escape the local Pareto front.

6.2.3. Robustness Analysis. In order to further investigate the robustness of MOCDOA for the EED problem, 30 trials for each case are performed to obtain the statistical results for the solutions of three objectives, that is, the minimal fuel cost, the minimal emission, and the ASD of compromise solution. Tables 14 and 15 list the statistical results of the solutions of the three objectives, respectively. Figure 7 depicts the box and whiskers plots of the solutions of the three objectives for two cases.

It can easily be seen from Figure 7 that the solutions of each trial remain close to the best obtained values for both cases. In Table 14, the standard deviations of the three objectives are 0.041390, $3.33e-05$, and 0.002742 for Case1. In Table 15, the standard deviations of the three objective are 0.026281, $1.98e-05$, and 0.003077 for Case2. From Tables 14 and 15, it is clear that the standard deviations of the solutions for the three objectives are small. This illustrates that the proposed MOCDOA provides the high-quality solutions and has a strong robustness for solving the constrained EED problem.

6.3. IEEE 30-Bus 10-Unit System. In this section, two experiments are carried out for IEEE 30-bus 10-unit system. The first experiment is to minimize the fuel cost and emission objectives by using basic CDOA individually. The second experiment is to compare the best solutions of the fuel cost, the best solutions of the emission, and the compromise solutions for the four algorithms on Case1 and Case2.

The fuel cost and the emission are minimized individually by the basic CDOA in all cases. Table 16 shows the best solutions. As in the above case, bold values in all tables represent the best results obtained for each case. The convergence of the fuel cost and the emission objectives for Case1 and Case2 is shown in Figure 8. The minimum values to consider the cost as only objective function are 106183.951158 \$/h and 111521.601406 \$/h on Case1 and Case2, respectively. The minimum values to consider the emission as only objective function are 3651.072701 ton/h and 3933.012596 ton/h on Case1 and Case2, respectively. Multiobjective results obtained by optimizing the cost and the emission simultaneously are discussed below.

In order to express how competitive the proposed algorithm is, it is compared with MOPSO, NSGA-II, and PESA-II. For fair comparison, 30 independent optimization runs have been carried out. The beat fuel cost, the best emission solutions, and the compromise solutions are given in Tables 17, 18, and 19, respectively. Figure 9 shows the Pareto fronts of MOCDOA, MOPSO, NSGA-II, and PESA-II.

From Table 17, the cost value of MOCDOA is smaller than those of the other three algorithms in Case1. In case2, the cost value of PESA-II is the smallest in those of the four algorithms, and the value of MOCDOA is only worse than that of PESA-II. From Table 18, the best emission value of MOPSO is the smallest in those of the four algorithms for Case1 and Case2. The best emission value of MOCDOA is worse than that of MOPSO and is better than those of the other two algorithms. As can be seen from Table 19, the ASD of MOCDOA is the best in those of the four algorithms; that is, the satisfaction of MOCDOA is the best in those of the four algorithms. Moreover, the EED problem is a multiobjective problem, and the results from Table 19 are the final results of the IEEE 30-bus 10-unit system in the two cases. It can be seen from Figure 9 that the Pareto front of MOCDOA is more homogeneous than those of the other three algorithms. It is

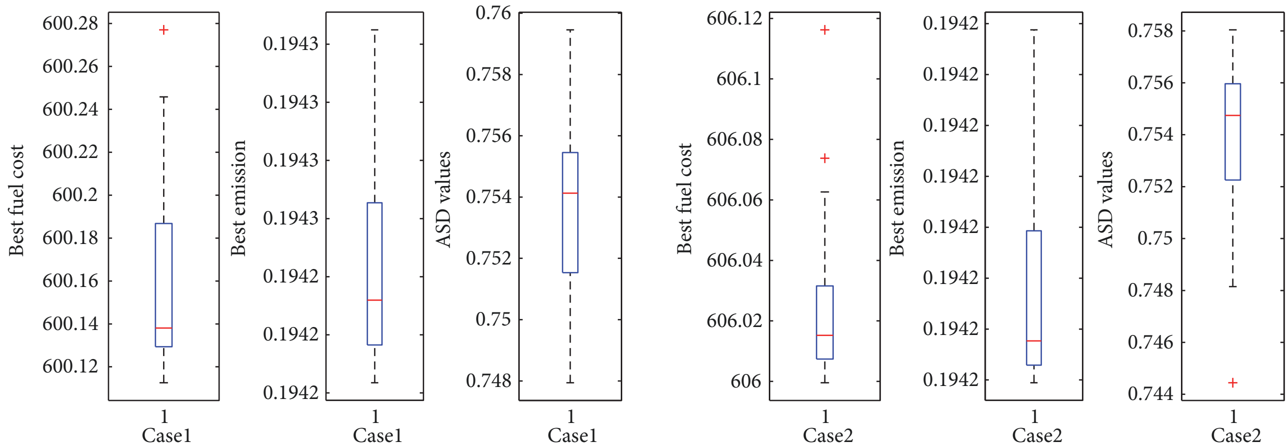


Figure 7: IEEE 30-bus 6-unit system statistical results obtained by MOCDOA on Case1 and Case2.

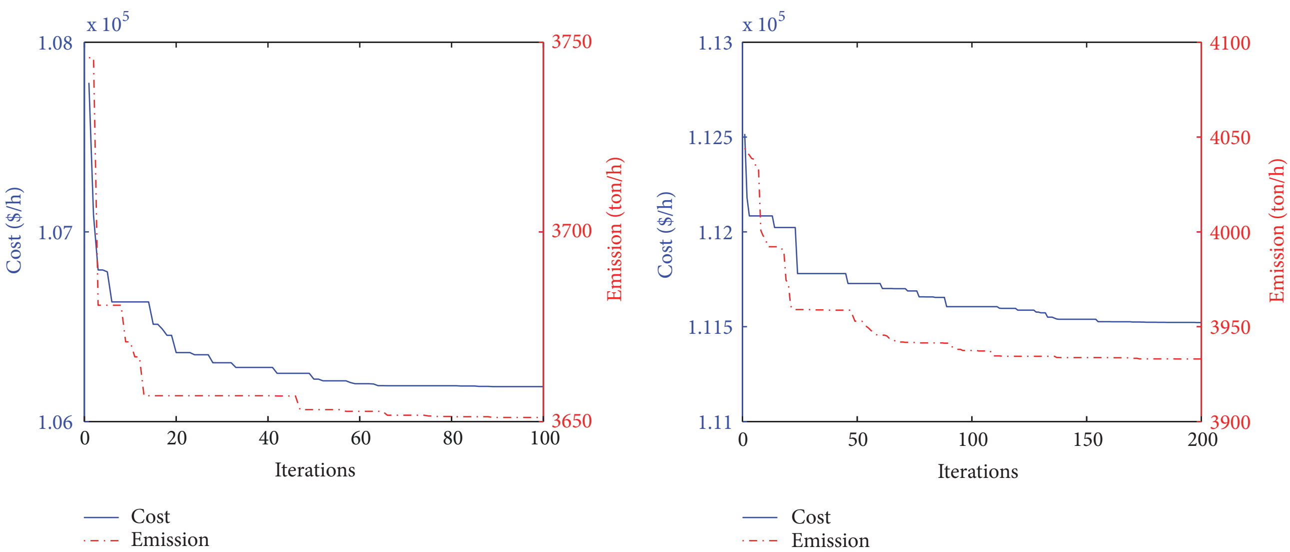


Figure 8: IEEE 30-bus 10-unit system convergence of cost and emission objective functions on Case1 and Case2.

Table 16: IEEE 30-bus 10-unit system best solutions for cost and emission optimized individually.

	Case1		Case2	
	Best cost	Best emission	Best cost	Best emission
G_1	54.354899	54.969212	54.237557	54.992582
G_2	77.475920	76.928036	79.941879	78.938898
G_3	88.276828	78.916625	104.852031	80.557478
G_4	81.509714	78.967157	99.757723	82.288308
G_5	66.071942	160	84.152371	159.974756
G_6	71.847842	240	87.902733	239.943905
G_7	287.674673	275.155451	298.512528	289.201904
G_8	332.788181	276.363682	338.327682	296.527420
G_9	470	379.528659	469.615571	400.653771
G_{10}	470	379.171177	469.619537	398.609472
Fuel cost ($/h)	**106183.951158**	111870.335739	**111521.601406**	116381.181212
Emission (ton/h)	4278.459561	**3651.072701**	4545.826580	**3933.012596**

Table 17: IEEE 30-bus 10-unit system best solutions for cost with four algorithms on Case1 and Case2.

	Case1				Case2			
	MOCDOA	MOPSO	NSGA-II	PESA-II	MOCDOA	MOPSO	NSGA-II	PESA-II
G_1	54.158870	54.134753	53.691308	54.803688	54.127113	52.863198	51.755735	54.997131
G_2	78.422533	78.139386	75.399985	77.335058	79.207632	78.825931	73.834838	79.995928
G_3	97.804234	110.593988	84.188022	89.476491	94.923188	104.094617	93.469134	112.435619
G_4	93.483470	92.354559	87.079197	62.613846	92.962807	102.294750	93.444728	85.764687
G_5	76.419930	80.279634	110.049490	67.144314	81.761262	97.001150	111.290861	91.517523
G_6	73.121641	70.016616	115.870843	86.785823	110.961706	95.871712	139.548089	82.535812
G_7	280.934498	279.361011	278.237057	289.140313	299.253109	294.030349	293.422964	299.966549
G_8	311.644912	321.711273	309.757782	335.411589	336.963929	325.450671	316.810101	339.967105
G_9	466.664170	446.519442	437.948030	468.245354	467.138652	467.865756	456.774987	469.961342
G_{10}	467.345738	466.889336	447.778284	469.043519	469.426001	468.141081	454.933764	469.969235
Emission	4252.190307	4260.635634	4001.839787	4249.562117	4463.735501	4487.984281	4294.006682	4545.689410
Fuel cost	**106264.834496**	106376.240273	107024.921918	106288.960020	111647.243250	111665.652452	112339.243763	**111543.388926**

Table 18: IEEE 30-bus 10-unit system best solutions for emission with four algorithms on Case1 and Case2.

	Case1				Case2			
	MOCDOA	MOPSO	NSGA-II	PESA-II	MOCDOA	MOPSO	NSGA-II	PESA-II
G_1	52.532519	52.852002	52.991134	54.222507	53.96756	54.236469	53.956584	53.908881
G_2	72.435520	69.597818	75.182766	75.066663	72.905941	78.981985	78.700245	77.049995
G_3	82.379093	71.767701	75.899072	84.396494	91.018244	76.975535	84.717066	84.158908
G_4	84.050261	81.122486	75.723913	84.227764	86.648978	77.977286	84.825929	80.928031
G_5	152.491324	158.672750	146.318487	151.316167	159.951634	159.519513	154.470253	151.279898
G_6	238.073874	238.166505	203.771856	230.429877	238.723255	237.781338	180.056057	235.222332
G_7	274.238437	264.847460	268.891862	271.546877	276.910571	276.088787	299.343585	290.055655
G_8	265.509554	286.431874	305.379732	277.395716	298.133654	300.031806	307.019564	304.265163
G_9	404.020356	391.188744	397.523330	384.398490	389.569294	405.729260	425.923887	403.498611
G_{10}	374.269061	385.352658	398.317844	386.999443	413.796996	414.886666	414.052021	401.556885
Fuel cost	111555.483102	111696.091418	110017.658015	111208.638485	116322.893631	116209.080846	114211.583909	115939.573531
Emission	3680.578627	**3667.508603**	3728.840966	3681.510679	3950.764391	**3945.408627**	4048.348353	3960.706739

Table 19: IEEE 30-bus 10-unit system best compromise solutions with four algorithms on Case1 and Case2.

	Case1				Case2			
	MOCDOA	MOPSO	NSGA-II	PESA-II	MOCDOA	MOPSO	NSGA-II	PESA-II
G_1	51.811253	54.146837	51.986438	54.500735	53.019793	54.564607	54.852490	54.399261
G_2	74.716733	79.720887	73.179620	76.228593	75.439733	73.581243	76.037006	77.493806
G_3	81.292214	83.454045	84.019834	82.014741	89.556375	88.832532	88.950481	87.999806
G_4	82.075395	75.651603	82.096182	81.748678	86.966333	87.819192	89.313517	87.373982
G_5	121.509751	144.184062	131.099548	121.142989	137.307583	128.411293	136.436222	120.078071
G_6	147.511266	148.565014	147.290760	139.813680	147.649522	144.095165	155.293274	142.652156
G_7	280.003081	284.922355	287.204027	288.236121	289.584082	298.618120	296.733338	297.184408
G_8	307.750415	293.335856	295.380918	299.231441	322.652284	321.940573	318.147618	330.804592
G_9	432.468854	407.299855	424.290718	421.127067	442.611979	443.421947	434.291352	445.836158
G_{10}	420.861036	428.719484	423.451949	435.955950	439.648299	443.427277	433.995612	441.036231
Fuel cost	107838.048975	108390.120075	108052.518200	107665.729718	112729.353799	113274.171282	113135.673565	112556.796831
Emission	3891.545095	3827.895782	3867.175411	3900.324575	4180.868813	4208.176152	4152.140421	4231.601876
ASD	**0.6810**	0.6279	0.5842	0.6581	**0.6802**	0.6566	0.5730	0.6471

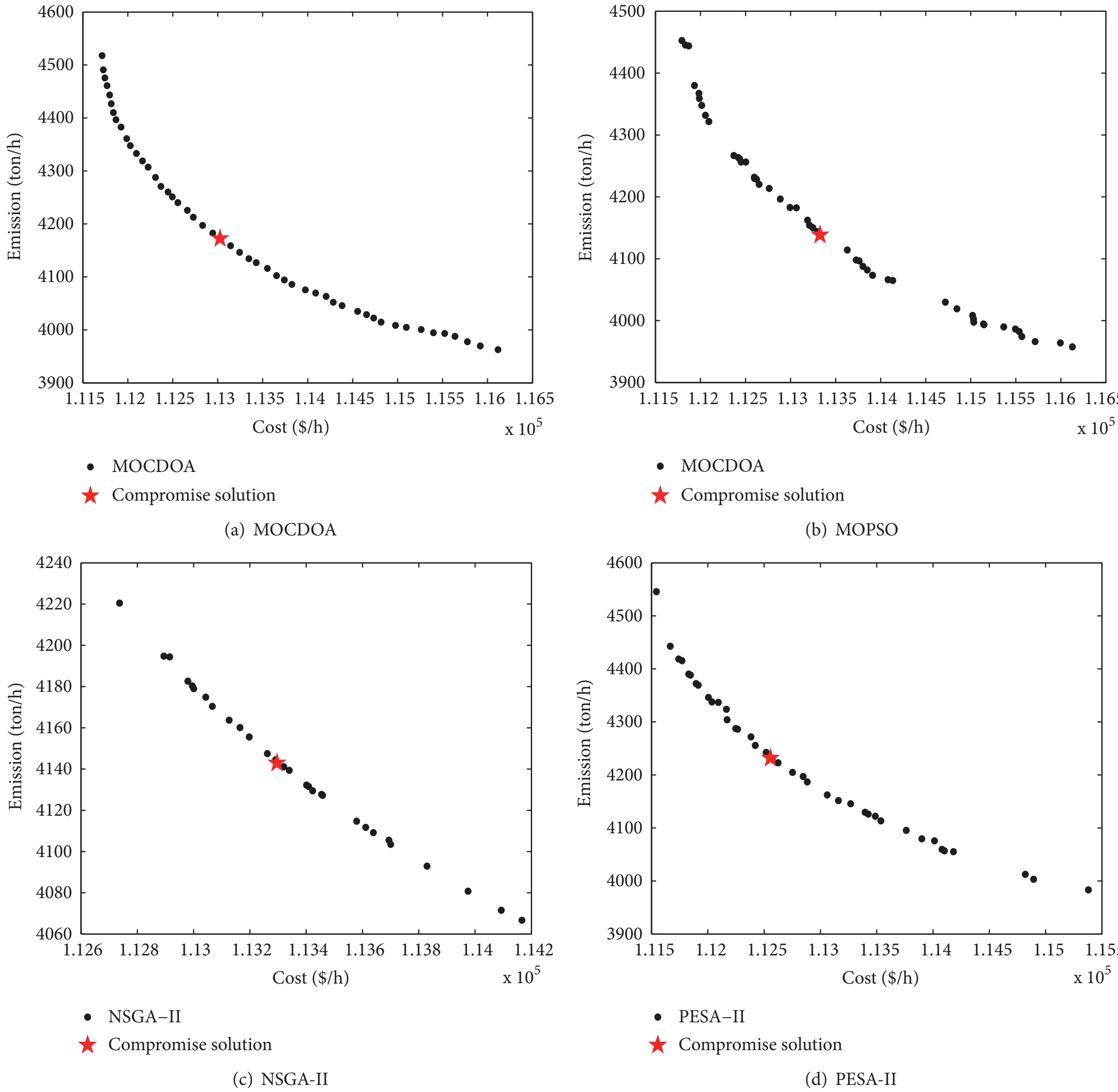


FIGURE 9: IEEE 30-bus 10-unit system Pareto fronts and compromise solution for the four algorithms on Case2.

quite evident that the proposed MOCDOA performs better than the other three algorithms. Combining the above results, we can see that MOCDOA is more efficient than almost the other three compared algorithms.

7. Conclusions

A novel algorithm MOCDOA presented in this paper is successfully employed to solve the environmental/economic power dispatch (EED) optimization problem with constrains. The algorithm extends the single-objective collective decision optimization algorithm to solve the multiobjective optimization problems for the first time. In order to enhance the uniformity of the obtained Pareto optimal solutions, two learning strategies are adopted, which are a leader-updating strategy based on maximum distance to measure the sparse degree of solutions and a wise random perturbation strategy based on the sparse mark to search the area around a leader. Furthermore, an extreme point of the nondominated solutions in MOCDOA is picked at random to replace a geometric center in CDOA, which can improve the diversity of Pareto optimal solutions. In addition, a new constraint-handling strategy to deal with the equality constraint of the EED problem and an established cyclic crowded sorting method to retain the external archive are designed in MOCDOA. The IEEE 30-bus 6-unit system and 10-unit system are used to investigate the effectiveness of the proposed MOCDOA. The results of MOCDOA's extreme solutions and compromise solutions are compared with those existing results obtained by other eight established algorithms. The compared results exhibit that MOCDOA has a good compromise solution and highly diverse Pareto optimal solutions in the lossless and loss-considered cases. Compared with other well-known algorithms for three metrics, SP, HV, and CM, MOCDOA reveals its superior characteristics and strong robust in the EED problem. It can be concluded that MOCDOA has the potential to be applied for solving some other multiobjective power system optimization problems.

Data Availability

The table data and modeling data used to support the finding of this study are included within the article. In addition, in order to better share the research results, the LaTeX codes relating the data are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China (no. 61370092), Hubei Provincial Department of Education Outstanding Youth Scientific Innovation Team Support Foundation (T201410), and State Key Laboratory of Silicate Materials for Architectures (Wuhan University of Technology, no. SYSJJ2018-21).

References

[1] J. Cai, X. Ma, L. Li et al., "Economic dispatch in power plant based on chaotic ant swarm optimization," in *Proceedings of the 7th IET International Conference on Advances in Power System Control, Operation and Management (APSCOM 2006)*, pp. 227-227, Hong Kong, China.

[2] J. Cai, X. Ma, L. Li, and P. Haipeng, "Chaotic particle swarm optimization for economic dispatch considering the generator constraints," *Energy Conversion and Management*, vol. 48, no. 2, pp. 645–653, 2007.

[3] J. H. Talaq, F. El-Hawary, and M. E. El-Hawary, "A summary of environmental/economic dispatch algorithms," *IEEE Transactions on Power Systems*, vol. 9, no. 3, pp. 1508–1516, 1994.

[4] A. Farag, S. Al-Baiyat, and T. C. Cheng, "Economic load dispatch multiobjective optimization procedures using linear programming techniques," *IEEE Transactions on Power Systems*, vol. 10, no. 2, pp. 731–738, 1995.

[5] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Adaptive Configuration of evolutionary algorithms for constrained optimization," *Applied Mathematics and Computation*, vol. 222, pp. 680–711, 2013.

[6] S. Das and A. Konar, "A swarm intelligence approach to the synthesis of two-dimensional IIR filters," *Engineering Applications of Artificial Intelligence*, vol. 20, no. 8, pp. 1086–1096, 2007.

[7] S. F. J. Brodsky and R. W. Hahn, "Assessing the influence of power pools on emission constrained economic dispatch," *IEEE Transactions on Power Systems*, vol. 1, no. 1, pp. 57–62, 1986.

[8] G. P. Granelli, M. Montagna, G. L. Pasini, and P. Marannino, "Emission constrained dynamic dispatch," *Electric Power Systems Research*, vol. 24, no. 1, pp. 55–64, 1992.

[9] C. S. Chang, K. P. Wong, and B. Fan, "Security-constrained multiobjective generation dispatch using bicriterion global optimisation," *IEE Proceedings Generation, Transmission and Distribution*, vol. 142, no. 4, pp. 406–414, 1995.

[10] R. Perez-Guerrero and J. Cedefio-Maldonado, "Differential evolution based economic environmental power dispatch," in *Proceedings of the 37th Annual North American Power Symposium*, pp. 191–197, Ames, IA, USA, 2005.

[11] A. De Lorena Stanzani, A. R. Balbo, L. Nepomuceno, and E. C. Baptista, "Solving the multiobjective environmental/economic dispatch problem using weighted sum and ε -constraint strategies and a predictor-corrector primal-dual interior point method," *Journal of Control, Automation and Electrical Systems*, vol. 25, no. 4, pp. 503–515, 2014.

[12] A. M. Jubril, O. A. Olaniyan, O. A. Komolafe, and P. O. Ogunbona, "Economic-emission dispatch problem: A semi-definite programming approach," *Applied Energy*, vol. 134, pp. 446–455, 2014.

[13] K. Bhattacharjee, A. Bhattacharya, and S. Halder Nee Dey, "Solution of Economic Emission Load Dispatch problems of power systems by Real Coded Chemical Reaction algorithm," *International Journal of Electrical Power & Energy Systems*, vol. 59, pp. 176–187, 2014.

[14] M. Modiri-Delshad and N. A. Rahim, "Multi-objective backtracking search algorithm for economic emission dispatch problem," *Applied Soft Computing*, vol. 40, pp. 479–494, 2016.

[15] M. A. Abido, "A niched Pareto genetic algorithm for multiobjective environmental/economic dispatch," *International Journal of Electrical Power & Energy Systems*, vol. 25, no. 2, pp. 97–105, 2003.

[16] M. A. Abido, "A novel multiobjective evolutionary algorithm for environmental/economic power dispatch," *Electric Power Systems Research*, vol. 65, no. 1, pp. 71–81, 2003.

[17] M. A. Abido, "Multiobjective evolutionary algorithms for electric power dispatch problem," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 315–329, 2006.

[18] C.-L. Chiang, "Optimal economic emission dispatch of hydrothermal power systems," *International Journal of Electrical Power & Energy Systems*, vol. 29, no. 6, pp. 462–469, 2007.

[19] M. Basu, "Dynamic economic emission dispatch using nondominated sorting genetic algorithm-II," *International Journal of Electrical Power & Energy Systems*, vol. 30, no. 2, pp. 140–149, 2008.

[20] R. T. F. A. King, H. C. S. Rughooputh, and K. Deb, "Evolutionary Multi-objective Environmental/Economic Dispatch: Stochastic Versus Deterministic Approaches," in *Proceedings of the Third International Conference, EMO 2005, Evolutionary Multi-Criterion Optimization*, pp. 677–691, Guanajuato, Mexico, March 2005.

[21] B. Zhao and Y.-J. Cao, "Multiple objective particle swarm optimization technique for economic load dispatch," *Journal of Zhejiang University SCIENCE A*, vol. 6, no. 5, pp. 420–427, 2005.

[22] T. A. A. Victoire and P. N. Suganthan, "Improved MOCLPSO algorithm for environmental/economic dispatch," in *Proceedings of the 2007 IEEE Congress on Evolutionary Computation, CEC 2007*, pp. 3072–3076, Singapore, September 2007.

[23] L. Wang and C. Singh, "Environmental/economic power dispatch using a fuzzified multi-objective particle swarm optimization algorithm," *Electric Power Systems Research*, vol. 77, no. 12, pp. 1654–1664, 2007.

[24] J. Cai, X. Ma, Q. Li, L. Li, and H. Peng, "A multi-objective chaotic particle swarm optimization for environmental/economic dispatch," *Energy Conversion and Management*, vol. 50, no. 5, pp. 1318–1325, 2009.

[25] S. Agrawal, B. K. Panigrahi, and M. K. Tiwari, "Multiobjective particle swarm algorithm with fuzzy clustering for electrical power dispatch," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 5, pp. 529–541, 2008.

[26] Y. Zhang, D.-W. Gong, and Z. Ding, "A bare-bones multi-objective particle swarm optimization algorithm for environmental/economic dispatch," *Information Sciences*, vol. 192, pp. 213–227, 2012.

[27] T. Liu, L. Jiao, W. Ma, J. Ma, and R. Shang, "Cultural quantum-behaved particle swarm optimization for environmental/economic dispatch," *Applied Soft Computing*, vol. 48, pp. 597–611, 2016.

[28] M. Varadarajan and K. S. Swarup, "Solving multi-objective optimal power flow using differential evolution," *IET Generation, Transmission & Distribution*, vol. 2, no. 5, pp. 720–730, 2008.

[29] Q. Zhang, R. Wang, J. Yang, K. Ding, Y. Li, and J. Hu, "Collective decision optimization algorithm: A new heuristic optimization method," *Neurocomputing*, vol. 221, pp. 123–137, 2017.

[30] H. Saadat, *Power system analysis*, McGraw-Hill, 1999.

[31] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[32] Y. Luo, M. Chen, and C. Zhang, "Improved NSGA-II algorithm with circular crowded sorting," *Control and Decision*, vol. 2, pp. 227–231, 2010.

[33] L. Wang and C. Singh, "Balancing risk and cost in fuzzy economic dispatch including wind power penetration based on particle swarm optimization," *Electric Power Systems Research*, vol. 78, no. 8, pp. 1361–1368, 2008.

[34] M. Sakawa, H. Yano, and T. Yumine, "An interactive fuzzy satisficing method for multiobjective linear-programming problems and its application," *The Institute of Electrical and Electronics Engineers Systems, Man, and Cybernetics Society*, vol. 17, no. 4, pp. 654–661, 1987.

[35] M. Basu, "Economic environmental dispatch using multi-objective differential evolution," *Applied Soft Computing*, vol. 11, no. 2, pp. 2845–2853, 2011.

[36] D. Corne W, N. Jerram R, J. Knowles D et al., "PESA-II: Region-based selection in evolutionary multiobjective optimization," in *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, pp. 283–290, Morgan Kaufmann Publishers Inc., 2001.

[37] D. Das B and C. Patvardhan, "New multi-objective stochastic search technique for economic load dispatch," *IEE Proceedings-Generation, Transmission and Distribution*, vol. 145, no. 6, pp. 747–752, 1998.

[38] M. A. Abido, "Environmental/economic power dispatch using multiobjective evolutionary algorithms," *IEEE Transactions on Power Systems*, vol. 18, no. 4, pp. 1529–1537, 2003.

[39] E. Zitzler, *Evolutionary algorithms for multiobjective optimization*, 2002.

[40] J. R. Schott, *Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization*, Air Force Institute of Technology Wright-Patterson AFB OH, 1995.

[41] L. H. Wu, Y. N. Wang, X. F. Yuan, and S. W. Zhou, "Environmental/economic power dispatch problem using multi-objective differential evolution algorithm," *Electric Power Systems Research*, vol. 80, no. 9, pp. 1171–1181, 2010.

[42] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: empirical results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.