Solving Multi-Area Environmental/Economic Dispatch by Pareto-Based Chemical-Reaction Optimization Algorithm

Junqing Li, *Member, IEEE,* Quanke Pan, Peiyong Duan, Hongyan Sang, and Kaizhou Gao

***Abstract*—In this study, we present a Pareto-based chemical-reaction optimization (PCRO) algorithm for solving the multi-area environmental/economic dispatch optimization problems. Two objectives are minimized simultaneously, i.e., total fuel cost and emission. In the proposed algorithm, each solution is represented by a chemical molecule. A novel encoding mechanism for solving the multi-area environmental/economic dispatch optimization problems is designed to dynamically enhance the performance of the proposed algorithm. Then, an ensemble of effective neighborhood approaches is developed, and a self-adaptive neighborhood structure selection mechanism is also embedded in PCRO to increase the search ability while maintaining population diversity. In addition, a grid-based crowding distance strategy is introduced, which can obviously enable the algorithm to easily converge near the Pareto front. Furthermore, a kinetic-energy-based search procedure is developed to enhance the global search ability. Finally, the proposed algorithm is tested on sets of the instances that are generated based on realistic production. Through the analysis of experimental results, the highly effective performance of the proposed PCRO algorithm is favorably compared with several algorithms, with regards to both solution quality and diversity.**

***Index Terms*—Chemical-reaction optimization algorithm, grid-based crowding distance, multi-area environmental/economic dispatch (MAEED) problem, multi-objective optimization.**

I. Introduction

Manuscript received October 25, 2015; revised March 1, 2016; accepted July 11, 2016; updated August 16, 2019. This work was partially supported by the National Natural Science Foundation of China (61773192, 61773246, 61603169, 61803192), Shandong Province Higher Educational Science and Technology Program (J17KZ005), Special Fund Plan for Local Science and Technology Development Lead by Central Authority, and Major Basic Research Projects in Shandong (ZR2018ZB0419). Recommended by Associate Editor Haibo He. *(Corresponding author: Peiyong Duan.)*

Citation: J. Q. Li, Q. K. Pan, P. Y. Duan, H. Y. Sang, and K. Z. Gao, "Solving multi-area environmental/economic dispatch by Pareto-based chemical-reaction optimization algorithm," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 5, pp. 1240−1250, Sept. 2019.

J. Q. Li and H. Y. Sang are with the School of Computer Science, Liaocheng University, Liaocheng 252059, China (e-mail: lijunqing@lcu-cs.com; sanghongyan@lcu-cs.com).

Q. K. Pan is with the School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200072, China (e-mail: panquanke@shu.edu.cn).

P. Y. Duan is with the School of Information Science and Engineering, Shandong Normal University, Jinan 250014, China (e-mail: duanpeiyong@sdnu.edu.cn).

K. Z Gao is with the Macau Institute of Systems Engineering, Macau University of Science and Technology, Macau 999078, China (e-mail: gaokaizhou@lcu-cs.com).

J. Q. Li is also with the School of Information Science and Engineering, Shandong Normal University, Jinan 250014, China.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JAS.2017.7510454

IN the power industry, economical dispatch (ED) has been studied for many years. With the objective to minimize fuel costs while fulfilling operational constraints, ED plays an important role in the power industry. During recent decades, considering the environmental dispatch constraints, the environmental/economic dispatch (EED) has gained more research attention. EED is a multi-objective problem with the following conflicting objectives: the minimization of fuel costs and pollution emission. Much research has been conducted to solve the EED problems [1]−[9]. Talaq et al. gave a summary of environmental/economic dispatch problems in electric power systems since 1970 [1]. During recent years, many heuristics and meta-heuristics have been applied to solve the EED problem, such as the linear programming techniques [2], genetic algorithm (GA) [3]−[6], and particle swarm optimization (PSO) [7], [8]. The most popular among these methods is genetic algorithm (GA), such as strength Pareto evolutionary algorithm (SPEA) [3], non-dominated sorting genetic algorithm (NSGA)-based approach [4], niched Pareto genetic algorithm (NPGA)-based approach [5], and NSGA-II-based approach [6]. Although many promising results have been obtained by applying the GA-based evolutionary method to solve the EED problem, many issues should be addressed for these algorithms, such as the convergence ability, solutions diversity, and the performing efficiency. Particle swarm optimization (PSO) is also a very popular intelligent optimization algorithm, which has also been widely applied to solve the EED problem, such as fuzzy clustering-based particle swarm (FCPSO) algorithm [7] and bare-bones multi-objective particle swarm optimization algorithm (BB-MOPSO) [8]. However, it is essential for the PSO-based algorithm that how to balance the ability of global search and local search. With the development of the cloud computing technology, a cluster of distributed devices has generally been built and utilized to provide on-demand computational resources or services for potential users across the internet [9]−[15]. Therefore, more and more studies have focused on the research in cloud or distributed production environments. Although there is much literature regarding the EED problems, most of them have not considered the multi-area EED (MAEED) problems, which is the common case in industrial applications. Several heuristics and meta-heuristics have been proposed for multi-area ED problems, such as the Dantzig Wolfe decomposition principle [16], evolutionary programming [17], expert systems [18], multi-objective particle swarm optimization (MOPSO) algorithm [19], and teaching-learning-based optimization algorithm (TLBO) [20]. Moreover, the extended version of the MAEED has also been studied in recent years. However, for the MAEED problem, two important issues should be solved: 1) the Pareto-based multi-objective method should be introduced to solve this type of problem, because most of the recently published approaches are based on the weighted sum approach; 2) the global and local search abilities should be balanced to increase the convergence ability while maintaining the solutions diversity. Very recently, by simulating the behavior of molecules in chemical reactions, an efficient chemical-reaction optimization (CRO) algorithm was proposed by Lam and Li to optimize combinatorial problems [21]. CRO has four elementary reactions, namely, on-wall ineffective collision, inter-molecular ineffective collision, decomposition, and synthesis. Experimental comparisons demonstrated that the performance of CRO is competitive with other swarm intelligent algorithms [21]−[24]. In this paper, we propose an effective Pareto-based CRO (PCRO) to solve the realistic multi-area environmental/economic dispatch problem. The main contributions of this study are given as follows: 1) we present a Pareto-based chemical-reaction optimization (PCRO) algorithm for solving the multi-area environmental/economic dispatch optimization problems; 2) a novel encoding mechanism for solving the multi-area environmental/economic dispatch optimization problems is developed to dynamically enhance the performance of the proposed algorithm; 3) an ensemble of effective neighborhood approaches is developed, and a self-adaptive neighborhood structure selection mechanism is also embedded in PCRO to increase the search ability while maintaining population diversity; 4) a grid-based crowding distance strategy is introduced, which can obviously enable the algorithm to easily converge near the Pareto front; and 5) a kinetic-energy-based search procedure is developed to enhance the global search ability. The rest of this paper is organized as follows: Section II describes the problem. The related algorithms are presented in Section III. Section IV gives the framework of the proposed algorithm. Section V illustrates the experimental results and compares them to the present performing algorithms from the literature to demonstrate the superiority of the proposed algorithm. The last section gives the concluding remarks and future research directions.

II. Problem Description

There are multiple areas in the MAEED, each containing several generators. The power can be transmitted from one area to another, while the transmission cost is routinely considered. Therefore, the objective of MAEED is to minimize the total production cost of supplying loads to all areas while satisfying the power balance constraints. Meanwhile, the pollution emission should also be minimized.

A. Design Objectives With Quadratic Cost Function

The first objective of the MAEED problem is to minimize the operational costs. In this study, the generator cost curves are represented by quadratic functions, similar to [8]. The total \$/h fuel cost FC ($\mathrm{P_G}$) can be represented as follows:

$$\mathrm{FC}\left(\mathrm{P_G}\right) = \sum_{i=1}^{N}\sum_{j=1}^{M_i}\left(a_{ij} + b_{ij}P_{G_{ij}} + c_{ij}P_{G_{ij}}^2\right) \tag{1}$$

where $P_{G_{ij}}$ is the power generated by the jth generator in area i, N is the area number of the system, M_i is the number of generators in the jth area, and a_{ij}, b_{ij}, and c_{ij} are the cost coefficients of the jth generator in area i.

The second objective of the MAEED problem is to minimize the pollution emission. The SO_2 and NO_x emissions can be approximated by using a quadratic function of the generator output, which is given as follows.

$$\mathrm{FE}\left(\mathrm{P_G}\right) = \sum_{i=1}^{N}\sum_{j=1}^{M_i}\left(a_{ij} + \beta_{ij}P_{G_{ij}} + \gamma_{ij}P_{G_{ij}}^2\right) \tag{2}$$

where α_{ij}, β_{ij}, and γ_{ij} are pollution emission coefficients of the jth generator in area i.

B. Design Objectives With Valve Point Loading

The total \$/h fuel cost FC ($\mathrm{P_G}$) considering valve-point loading of the generator can be represented as follows [19]:

$$\begin{aligned}\mathrm{FC}\left(\mathrm{P_G}\right) = &\sum_{i=1}^{N}\sum_{j=1}^{M_i}\left(a_{ij} + b_{ij}P_{G_{ij}} + c_{ij}P_{G_{ij}}^2\right)\\ &+\sum_{i=1}^{N}\sum_{j=1}^{M_i}\left|d_{ij}\times\sin\left\{e_{ij}\times\left(P_{G_{IJ}}^{\min} - P_{G_{ij}}\right)\right\}\right|\end{aligned} \tag{3}$$

where d_{ij} and e_{ij} are the cost coefficients of the jth generator in area i due to the valve-point effect.

The SO_2 and NO_x emissions can be computed as follows considering valve-point loading of the generator [18].

$$\begin{aligned}\mathrm{FC}\left(\mathrm{P_G}\right) = &\sum_{i=1}^{N}\sum_{j=1}^{M_i}\frac{a_{ij} + b_{ij}P_{G_{ij}} + c_{ij}P_{G_{ij}}^2}{100}\\ &+\sum_{i=1}^{N}\sum_{j=1}^{M_i}\xi_{ij}\times\exp\left(P_{G_{ij}}\lambda_{ij}\right)\end{aligned} \tag{4}$$

where ξ_{ij} and λ_{ij} are pollution emission coefficients of the jth generator in area i due to the valve-point effect.

C. Design Constraints

The constraints considered in this study include the generation capacity of each generator, area power balance and tie-line transfer limits.

Constraint 1: Generation capacity constraint. The real power output of each generator in each area is routinely restricted by lower and upper bounds as follows:

$$P_{G_{ij}}^{\min} \le P_{G_{ij}} \le P_{G_{ij}}^{\max} \tag{5}$$

where $P_{G_{ij}}^{\min}$ and $P_{G_{ij}}^{\max}$ are the minimum and maximum power produced by the jth generator in area i.

Constraint 2: Generation power balance constraint. The total real power generation of each area should satisfy its predefined power demand plus the total real power loss in the transmission lines in the area, which is computed as follows:

$$\sum_{j=1}^{M_i} P_{G_{ij}} - P_{D_i} - P_{L_i} - \sum_{k,k\neq i} T_{ik} = 0 \ \ \forall i \tag{6}$$

where P_{D_i} is the real power demand of area i. T_{ik} is the tie-line real power transfer from area i to area k, which is a positive value if the power is transferred from area i to area k and a negative value vice versa. The transmission loss P_{L_i} of area i may be expressed by using B-coefficients as follows:

$$P_{L_i} = \sum_{j=1}^{M_i}\sum_{l=1}^{M_i} P_{G_{ij}} B_{ilj} P_{G_{il}} + \sum_{j=1}^{M_i} B_{0ij} P_{G_{ij}} + B_{00i}. \tag{7}$$

Constraint 3: The tie-line real power transfer T_{ik} from area i to area k should not exceed the tie-line capacity, which is given as follows:

$$T_{ik}^{\min} \leq T_{ik} \leq T_{ik}^{\max}. \tag{8}$$

III. The Related Algorithms

A. The Canonical CRO

CRO was introduced by Lam and Li in 2010; it loosely mimics what happens to molecules in a chemical reaction system and tries to capture the energy in the reaction process [21]−[24]. The molecules represent the solutions for the considered problem, and possess two types of energies, i.e., potential energy (PE) and kinetic energy (KE). PE corresponds to the objective function of a molecule while the KE of a molecule symbolizes its ability to escape from a local minimum. CRO is a population-based intelligent algorithm with variable population size. In the canonical CRO, there are four elementary collisions: the on-wall ineffective collision, decomposition, inter-molecular ineffective collision, and synthesis. The main features of the four collisions are given as follows [21]−[24]:

1) The on-wall ineffective collision reaction occurs when a molecule hits the wall, bounces back, and then becomes a new molecule if the given condition is satisfied. During the collision, the molecule will lose some percent of KE to the buffer.

2) The decomposition reaction is used to mimic the process of hitting the wall and then decomposing into two or more pieces.

3) The process of two or more molecules sharing information with each other and then producing two or more other different molecules is called inter-molecular ineffective collision.

4) Synthesis is the process where more than one molecule collide and combine together.

B. The Concept of Pareto Optimal

To describe concisely, we list several basic Pareto definitions as follows [25]−[28]:

1) Pareto dominance: A solution x is said to (Pareto) dominate another solution u (denoted as $x \prec u$) if and only if

$$\begin{aligned}&(\forall i \in \{1,2,\cdots,w\} : f_i(x) \leq f_i(u)\\ &\quad \wedge (\exists j \in \{1,2,\cdots,w\} : f_j(x) < f_j(u))\end{aligned} \tag{9}$$

Pareto non-dominated

$$x \sim \mu \equiv \neg x \prec \mu \wedge \neg \mu \prec x. \tag{10}$$

2) Optimal Pareto solution: A solution x is said to be an optimal Pareto solution if and only if there is no solution v in the search space that satisfies $v \prec x$.

$$\neg \exists v \in \Theta : v \prec x. \tag{11}$$

3) Grid-based crowding distance: Increasing the selection pressure towards the Pareto front is crucial to the proposed algorithm. In this study, we introduce a grid-based crowding distance into our proposed algorithm, as in [28].

Given a set of non-dominated solutions, let $\min_k(P)$ and $\max_k(P)$ denote the minimum and maximum value of the kth objective, respectively. Let M be the number of objectives.

Let $v_k = (\max_k(P) - \min_k(P))/(2 \times div)$, $lb_k = \min_k(P) - v_k$, $ub_k = \max_k(P) + v_k$, and $d_k = (ub_k - lb_k)/div$.

Let $G_k(x) = \lfloor (F_k(x) - lb_k)/d_k \rfloor$, which is the grid location of an individual in the kth objective, where $\lfloor \cdot \rfloor$ denotes the floor function and $F_k(x)$ is the actual objective value in the kth objective.

Let $GR_x = \sum_{k=1}^{M} G_k(x)$, which is a convergence estimator for ranking individual solutions based on their grid locations.

Let $GD(x,y) = \sum_{k=1}^{M} |G_k(x) - G_k(y)|$, which is the grid difference between solutions x and y.

A solution y is regarded as a neighbor of a solution x, if $GD(x,y) < M$.

Let $GCD(x) = \sum_{y \in N(x)} (M - GD(x,y))$, which is the density estimator of solution x, where $N(x)$ is the set of neighboring solutions of x.

Let

$$GCPD(x) = \sqrt{\sum_{k=1}^{M} \left(\frac{F_k(x) - (lb_k + G_k(x) \times d_k)}{d_k}\right)^2}$$

which is the normalized Euclidean distance between an individual and the best corner of its hyperbox.

IV. The Proposed PCRO

In this section, we present the detailed implementation of the proposed PCRO algorithm, which includes the encoding and decoding, the improved CRO reaction operators, and the update process of the Pareto archive set. The flowchart of the PCRO algorithm is given as follows.

Step 1: Initialize the system parameters.

Step 2: Initialize the population, and evaluate each solution in the population (Section IV-A).

Step 3: Apply the non-dominated sorting function to the current population, and update the Pareto archive set by using the solutions in the first level (Section IV-H).

Step 4: While maximum computational time is not reached do following steps.

Step 5: For each molecule in the current population, apply the on-wall ineffective collision function (Section IV-D).

Step 6: Randomly select two molecules in the current population and apply the Inter-molecular ineffective collision function (Section IV-E).

Step 7: Apply the kinetic-energy-based search procedure (Section IV-I).

Step 8: Apply the decomposition (Section IV-F) and synthesis (Section IV-G) functions to the current population.

Step 9: Generate the next population: 1) Record the neighboring solutions generated by the above four elementary collisions into a neighboring set; 2) Apply the non-dominated function on the neighboring set; 3) Update the Pareto archive set by using the solutions in the first level; 4) Generate the next population by using the grid-based crowding method (Section IV-H).

A. Encoding

In this study, we propose a novel encoding mechanism for solving the multi-area environmental/economic dispatch optimization problems. A detailed realization is given as follows.

First, the whole evolution stage is divided into five stages.

1) At the first stage, each power generator is represented by an integer. For example, given a problem with six power generators, Table I presents the solution representation at the first stage. As shown in Table I, at the first stage, the energy value for power generator PG_1 is 0.4, while the values for other generators are 0.5, 0.5, 0.7, 0.6, and 0.3, respectively.

TABLE I
REPRESENTATION FOR THE FIRST STAGE

PG_1	PG_2	PG_3	PG_4	PG_5	PG_6
4	5	5	7	6	3

2) At the second stage, each power generator is represented by two integers. For example, given a problem with six power generators, Table II presents the solution representation at the second stage. As shown in Table II, at the second stage, the energy value for power generator PG_1 is 0.45, while the values for other generators are 0.34, 0.65, 0.45, 0.57, and 0.63, respectively.

TABLE II
REPRESENTATION FOR THE SECOND STAGE

PG_1	PG_2	PG_3	PG_4	PG_5	PG_6
45	34	65	45	57	63

3) At the last stage, each power generator is represented by five integers. For example, given a problem with six power generators, one solution for the last stage is 43 576, 49 534, 30 520, 2756, 3450, 87 625. The above solution means that at the last stage, the energy value for power generator PG_1 is 0.43576, while the values for other generators are 0.49534, 0.30520, 0.02756, 0.03450, and 0.87625, respectively.

B. Neighborhood Structures

In this study, considering the problem structure and the balance of the exploration and exploitation ability, five neighborhood structures are proposed, as follows:

1) Single-swap structure, denoted by N_1. a) Randomly select two integer numbers in the solution representation; b) Swap the two selected integers in the representation string.

2) Insert structure, denoted by N_2. a) Randomly select two positions r_1 and r_2 in the solution representation, where $r_1 < r_2$; b) Insert the integer at the position r_2 before r_1 in the representation string.

3) Consecutive-swap structure, denoted by N_3, with an example given in Fig. 1. a) Randomly select two positions r_1 and r_2 in the solution representation, where $r_1 < r_2$; b) Swap each pair of elements between the positions r_1 and r_2 in the representation string. The detailed steps are as follows: i) let $p = r_1$, and $q = r_2$; ii) swap the two elements p and q in the solution representation; let $p = r_1 + 1$, and $q = r_2 - 1$; iii) Repeat step ii) until $p \geq q$.

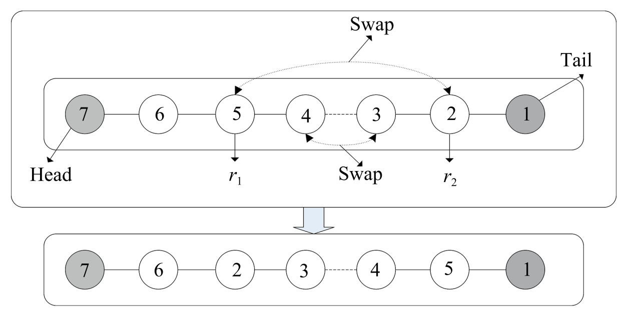


Fig. 1. N_3 neighborhood.

4) Multi-swap structure, denoted by N_4, similar to [4]. Perform several single-swap structures.

5) Multi-insert structure, denoted by N_5. a) Randomly generate a limit number h between h_1 and h_2, where h_1 and h_2 are the lower and upper bounds of the loop number, and are experimentally set to 5 and 10, respectively; b) Perform the following steps h times: randomly select one position r in the solution representation, insert the element at the positions $r + 1$ before the position $r - 1$. If $r = 0$, set $r - 1$ equal to n; if $r = n$, set $r + 1$ to 0.

C. A Self-adaptive Neighborhood Strategy

The neighboring approaches, introduced in Section IV-B, have different roles for the convergence capability or the population diversity. To balance the exploration and exploitation capability during the evolution, that is, to enhance the search ability while maintaining population diversity and to utilize different neighborhood structures in different stages, we introduce a self-adaptive strategy, that is similar to [29], [30]. The detailed steps of the proposed self-adaptive strategy are as follows.

Step 1: Set the function parameter, such as the length of the neighborhood vector: N_s, and the refill probability R_p.

Step 2: Initialize a neighborhood vector (NV), with length equal to N_s, by filling with a random neighborhood structure taken from the structures discussed in Section IV-B.

Step 3: Generate an empty winning neighborhood vector (WNV), with length equal to N_s.

Step 4: After receiving a call for selection of a neighborhood structure, perform Steps $5-7$.

Step 5: If NV is not empty, take the first neighboring structure from NV to generate a neighboring solution of the current one. If the current solution is dominated by or non-dominated with the new neighboring solution, insert the corresponding neighborhood structures into WNV.

Step 6: If NV is empty and WNV is not empty, fill NV with the elements of the current WNV. If the length of the new NV is less than N_s, the empty positions will be filled as follows: $75\,\%$ is refilled by WNV, and then the remaining $25\,\%$ is refilled by a random selection from the five neighborhood structures discussed in Section IV-B.

Step 7: If the WNV is empty, the new NV will be filled as follows: $50\,\%$ from the latest NV, with the remaining $50\,\%$ randomly selected from the neighborhood structures.

D. On-wall Ineffective Collision

In the proposed PCRO, each molecule performs the on-wall ineffective collision procedure to make a deep search during the evolution. Therefore, the on-wall ineffective collision is crucial for the algorithm and should consider both the exploration and exploitation ability. In this study, we propose a flexible on-wall ineffective collision, which is given as follows:

Step 1: Given a molecule, select a neighborhood structure by using the self-adaptive neighborhood strategy.

Step 2: Generate a new neighboring solution around the current individual by using the selected neighborhood structure.

Step 3: If the new neighboring solution is non-dominated with the current solution, or the current one is dominated by the new generated solution, insert the new neighboring solution into a neighbouring set and insert the selected neighborhood structure into the winning vector WNL; otherwise, discard the neighboring solution and decrease the kinetic energy value of the current solution.

E. Inter-molecular Ineffective Collision

The inter-molecular ineffective collision occurs when two molecules collide and then produce two new molecules. Similar to the canonical CRO, in this study, the inter-molecular ineffective collision is realized by performing independently on-wall ineffective collision for the two selected molecules. That is, in parallel, the two molecules experience a slight change to their structures.

F. Decomposition

The decomposition reaction produces two or more molecules based on one molecule. Because the solution in the Pareto archives set usually has a nice performance feature, therefore, in the proposed algorithm, we select a non-dominated solution to generate two neighboring solutions to replace the two solutions, i.e., the worst solution in the population, and the solution with the minimum kinetic energy value. If there is more than one solution with the same minimum kinetic energy value, randomly select one. The worst solution is the one in the last Pareto level and with the least grid crowding distance value. The decomposition is realized as follows:

Step 1: Randomly select a non-dominated solution from the Pareto archive set, denoted as ω_r.

Step 2: Perform an on-wall collision N_c times for the selected molecule and insert the new neighboring solutions into a solution set S_{nb}.

Step 3: Apply the non-dominated sorting function to the solution set S_{nb} and randomly select two solutions in the first Pareto level, if there is only one solution in the first level, randomly select another solution in the second level; the two selected solutions are denoted as ω_1 and ω_2, respectively.

Step 4: Replace the worst solution in the current population and the solution with the minimum kinetic energy value with ω_1 and ω_2, respectively.

G. Synthesis

The synthesis in the canonical CRO algorithm generates one solution based on two or more solutions to increase the exploration ability to escape from local optima. In this study, we realize the synthesis procedure as follows:

Step 1: Sequence the solutions in the current population in increasing order according to their kinetic energy values. Select the two molecules ω_1 and ω_2 at the first two positions.

Step 2: If ω_1 dominates ω_2, denote ω_1 as ω_c and ω_2 as ω_n. Otherwise, let $\omega_c = \omega_2$ and $\omega_n = \omega_1$.

Step 3: Perform an on-wall collision N_c times for ω_c, and insert the new neighboring solutions into a solution set S_{nb}.

Step 4: Apply the non-dominated sorting function to the solution set S_{nb}, randomly select one solution ω_b in the first Pareto level, and replace ω_n with ω_b.

H. Crowding Distance Based on Grid Division

In this study, the non-dominated sorting algorithm [31] was first applied to the population dividing the solutions into several levels according to their front level number. For the solutions at the same level, the solution with larger grid-based crowding distance is considered better than the one with a smaller crowding distance value. The detailed steps are as follows:

Step 1: For each kth objective of individuals in the current Pareto level, compute the min_k and max_k values.

Step 2: For each individual x, compute the lb_k, d_k, and ub_k values for each kth objective.

Step 3: For each individual x, compute the grid coordinate $G_k(x)$ and sum of grid coordinate $GR(x)$ values.

Step 4: For each individual x, compute and record the set of individuals that are dominated by x, or non-dominated with x.

Step 5: For each individual x, compute the grid difference $GD(x, y)$ from other individuals and record the neighboring solutions of each individual.

Step 6: For each individual x, compute the density estimator $GCD(x)$ and the normalized Euclidean distance $GCPD(x)$.

Step 7: Sort each individual in the current Pareto level by using the grid-based tournament selection function introduced in [28].

Step 8: Update the values of $G_k(x)$, $GCD(x)$, and $GCPD(x)$, after the selection.

I. Kinetic-energy-based Search Procedure

To enhance the ability to escape from the local optima, we propose a kinetic-energy-based search procedure, which is given as follows.

Step 1: In the initial phase, set the kinetic energy value to $KE_{\max}$ for each solution in the current population.

Step 2: At each evolution stage, record all solutions whose kinetic energy values equal to 0 into a set S_0.

Step 3: Randomly select one solution from the set S_0 and replace it with a randomly generated molecule.

V. Experimental Evaluation

This section discusses the computational experiments used to evaluate the performance of the proposed algorithm. Our algorithm was implemented in C++ on an Intel Core i7 3.4 GHz PC with 16 GB memory. The proposed algorithms were independently tested by using 30 runs for each case, and the best results were collected for comparison. During recent years, multi-objective optimization methods have commonly been compared using performance metrics such as convergence ratio to the Pareto front and hyper-volume. However, in the field of power systems, many currently published studies use the best cost solution, the best emission solution and the best compromise solution for comparison. Therefore, in this study, we also use the above three comparison metrics to verify the efficiency of the proposed PCRO algorithm. The parameters for the proposed PCRO are as follows: the initial population size is 50; the initial kinetic energy value for each solution $KE_{\max} = 20$; and the local search strength $N_c = 20$.

A. Performance for Case 1

In this section, we tested the proposed algorithm on the IEEE 30-bus problem with six generators. Table III gives the coefficients of fuel cost, pollution emissions (p.u.) and generator capacities. Table IV provides the best solutions for cost with eight algorithms with respect to optimizing Case 1. Table V presents the best solutions for emission, while Table VI lists the compromise solutions obtained by the proposed algorithm. The compared algorithms include LP [2], NSGA [4], NPGA [5], SPEA [3], NSGA-II [6], FCPSO [7], and BB-MOPSO [8].

As shown in Table IV, except the PCRO, the BB-MOPSO obtained the best solution with a minimum fuel cost of 600.112. However, our PCRO algorithm obtained a solution with the two objective values 600.092 and 0.221813, which dominates the solution obtained by BB-MOPSO. Meanwhile, the solutions obtained by LP, NSGA, and NSGA-II are also dominated by the above solution (600.092, 0.221813). In addition, the solution obtained by SPEA is dominated by (600.108, 0.221366) found by PCRO, and the result by FCPSO is also dominated by (600.123, 0.22114) provided by PCRO. Therefore, all solutions obtained by the seven compared algorithms are dominated by one solution due to the proposed PCRO algorithm. From the above analysis, we can conclude that PCRO is superior than the other seven compared algorithms.

From Table V, we can see that, the solution obtained by our PCRO algorithm dominates all solutions obtained by the other seven compared algorithms. It can be concluded from Tables IV−VI that the proposed PCRO is efficient for solving the given problem.

To show the efficiency of the proposed grid-based crowding distance strategy discussed in Section IV-H, we made a detailed comparison of the two approaches, i.e., the algorithm with the grid-based crowding distance strategy (denoted by PCRO_{G}) and the algorithm without the grid-based crowding distance strategy (denoted by PCRO_{NG}). Fig. 2 gives the last Pareto front comparisons of the two algorithms for optimizing Case 1. It can be seen from the figure that PCRO_{G} shows better than the results by PCRO_{NG} both in solutions quality and solutions diversity.

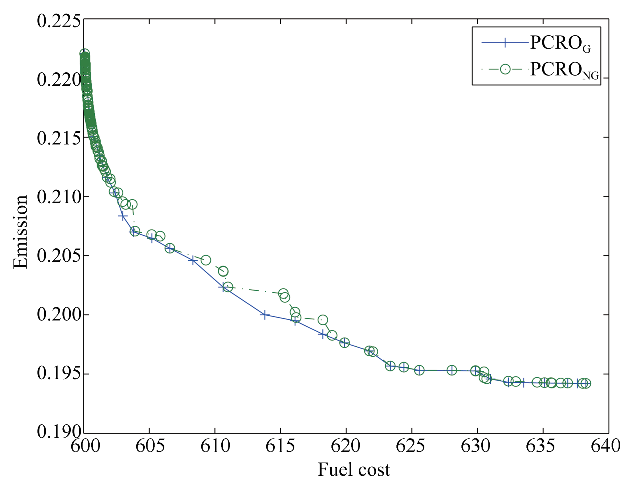


Fig. 2. Comparisons of the Pareto fronts for Case 1.

B. Performance for Case 2

In order to validate the constraint handling strategy and evaluate the performance of the proposed algorithm, in Case 2, the transmission losses are considered as in [8]. The B-coefficients related to the transmission losses are given as follows.

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

TABLE III
COEFFICIENTS OF FUEL COST, POLLUTION EMISSIONS (P.U.) AND GENERATOR CAPACITIES (FOR CASE 1, IEEE-30 BUS)

Generator	a_{ij}	b_{ij}	c_{ij}	α_{ij}	β_{ij}	γ_{ij}	ξ_{ij}	λ_{ij}	$P_{G_{ij}}^{\min}$	$P_{G_{ij}}^{\max}$
$G_{1,1}$	10	200	100	4.091	−5.554	6.490	2.0×10^{-4}	2.857	0.05	0.50
$G_{1,2}$	10	150	120	2.543	−6.047	5.638	5.0×10^{-4}	3.333	0.05	0.60
$G_{1,3}$	20	180	40	4.258	−5.094	4.586	1.0×10^{-6}	8.000	0.05	1.00
$G_{1,4}$	10	100	60	5.426	−3.550	3.380	2.0×10^{-3}	2.000	0.05	1.20
$G_{1,5}$	20	180	40	4.258	−5.094	4.586	1.0×10^{-6}	8.000	0.05	1.00
$G_{1,6}$	10	150	100	6.131	−5.555	5.151	1.0×10^{-5}	6.667	0.05	0.60

TABLE IV
BEST SOLUTIONS FOR COST WITH EIGHT ALGORITHMS WHEN OPTIMIZING CASE 1

	PG_1	PG_2	PG_3	PG_4	PG_5	PG_6	Fuel cost	Emission
BB-MOPSO	0.109	0.3005	0.5234	1.017	0.5238	0.3603	600.112	0.22220
LP	0.1500	0.3000	0.5500	1.0500	0.4600	0.3500	606.314	0.22330
NSGA	0.1567	0.2870	0.4671	1.0467	0.5037	0.3729	600.572	0.22282
NPGA	0.1080	0.3284	0.5386	1.0067	0.4949	0.3574	600.259	0.22116
SPEA	0.1062	0.2897	0.5289	1.0025	0.5402	0.3664	600.150	0.22151
NSGA-II	0.1059	0.3177	0.5216	1.0146	0.5159	0.3583	600.155	0.22188
FCPSO	0.1070	0.2897	0.525	1.015	0.5300	0.3673	600.132	0.22226
	0.112123	0.299888	0.523763	1.01258	0.525818	0.359739	**600.092**	**0.221813**
PCRO	0.113147	0.30149	0.525301	1.00719	0.525455	0.36137	**600.108**	**0.221366**
	0.113945	0.302306	0.527319	1.0044	0.524642	0.361385	**600.123**	**0.22114**

TABLE V
BEST SOLUTIONS FOR EMISSION WITH EIGHT ALGORITHMS WHEN OPTIMIZING CASE 1

	PG_1	PG_2	PG_3	PG_4	PG_5	PG_6	Fuel cost	Emission
BB-MOPSO	0.4071	0.4591	0.5374	0.3838	0.5369	0.5098	0.194203	638.262
LP	0.4000	0.4500	0.5500	0.4000	0.5500	0.5000	0.194227	639.600
NSGA	0.4394	0.4511	0.5105	0.3871	0.5553	0.4905	0.194356	639.209
NPGA	0.4002	0.4474	0.5166	0.3688	0.5751	0.5259	0.194327	639.180
SPEA	0.4116	0.4532	0.5329	0.3832	0.5383	0.5148	0.194210	638.507
NSGA-II	0.4074	0.4577	0.5389	0.3837	0.5352	0.5110	0.194204	638.249
FCPSO	0.4097	0.4550	0.5363	0.3842	0.5348	0.5140	0.194207	638.358
PCRO	**0.406447**	**0.457242**	**0.538921**	**0.384227**	**0.5381**	**0.509063**	**0.194203**	**638.102**

TABLE VI
BEST SOLUTIONS FOR COST AND EMISSION OPTIMIZED INDIVIDUALLY FOR CASE 1

	PG_1	PG_2	PG_3	PG_4	PG_5	PG_6	Fuel cost	Emission
Best cost	0.112123	0.299888	0.523763	1.01258	0.525818	0.359739	600.092	0.221813
Best emission	0.406447	0.457242	0.538921	0.384227	0.5381	0.509063	638.102	0.194203

$$B_0 = [-0.0107\ \ 0.0060\ \ -0.0017\ \ 0.0009\ \ 0.0002\ \ 0.0030]$$
$$B_{00} = 9.8573 \times 10^{-4}$$

The compared algorithms include SMOPSO [32], CMOPSO [33], TV-MOPSO [31], and BB-MOPSO [8]. The experimental results of SMOPSO, CMOPSO, TV-MOPSO are collected from [8]. Table VII provides the best solutions for cost with five algorithms with respect to optimizing Case 2, while Table VIII presents the best solutions for emission with five algorithms with respect to optimizing Case 2.

As shown in Table VII, except the PCRO, the BB-MOPSO obtained the best solution with the minimum fuel cost 605.9817. However, our PCRO algorithm obtained a solution with the objective values 603.108 and 0.217835, which dominates the solution obtained by BB-MOPSO. Meanwhile, the solutions obtained by SMOPSO, CMOPSO, and TV-MOPSO are also dominated by the above solution (603.108, 0.217835, 0.0115006). It should be noted that the total transmission loss

TABLE VII
BEST SOLUTIONS FOR COST WITH FIVE ALGORITHMS WHEN OPTIMIZING CASE 2

	PG_1	PG_2	PG_3	PG_4	PG_5	PG_6	Fuel cost	Emission	Loss
SMOPSO	0.1225	0.2899	0.5741	0.9932	0.5255	0.3547	606.0038	0.220522	0.02583
CMOPSO	0.1198	0.2928	0.5778	0.99	0.527	0.3524	606.0062	0.220414	0.02576
TV-MOPSO	0.1011	0.2883	0.5852	0.9832	0.5271	0.3749	606.1114	0.220503	0.02587
BB-MOPSO	0.1229	0.288	0.5792	0.9875	0.5255	0.3564	605.9817	0.220190	0.02562
PCRO	**0.166457**	**0.313815**	**0.509591**	**0.985789**	**0.499901**	**0.369948**	**603.108**	**0.217835**	**0.0115006**

TABLE VIII
BEST SOLUTIONS FOR EMISSION WITH FIVE ALGORITHMS WHEN OPTIMIZING CASE 2

	PG_1	PG_2	PG_3	PG_4	PG_5	PG_6	Fuel cost	Emission	Loss
SMOPSO	0.4078	0.4824	0.5388	0.3977	0.5335	0.5093	646.0817	0.194216	0.03549
CMOPSO	0.4097	0.4648	0.5523	0.394	0.5361	0.5123	645.7762	0.194186	0.03515
TV-MOPSO	0.4188	0.4582	0.553	0.3803	0.5345	0.5251	647.665	0.194203	0.03583
BB-MOPSO	0.4103	0.4661	0.5432	0.3883	0.5447	0.5168	646.4847	0.194179	0.03537
PCRO	0.4074	0.4577	0.5389	0.3837	0.546409	0.511	**640.752**	0.194196	**0.0111091**

TABLE IX
STATISTICAL RESULTS OF THE METRIC SC FOR CASE 2

	PCRO	BB-MOPSO	SMOPSO	CMOPSO	TV-MOPSO
SC(PCRO, $*$)	–	0.3321	0.4752	0.4312	0.8231
SC(BB-MOPSO, $*$)	0.0912	–	0.4125	0.3148	0.6629
SC(SMOPSO, $*$)	0.0231	0.0471	–	0.2037	0.4802
SC(CMOPSO, $*$)	0.0378	0.0628	0.2243	–	0.5867
SC(TV-MOPSO, $*$)	0	0	0.1255	0.0926	–

experienced by the provided PCRO algorithm is also the minimum among the five compared algorithms. From the above analysis, we can conclude that PCRO is superior than the other four compared algorithms.

From Table VIII, we can see that the proposed PCRO algorithm obtains a solution with the fuel cost, emission, and loss values equal to 640.752, 0.194196, and 0.0111091, respectively. The results of CMOPSO and BB-MOPSO are slightly better than the solution by PCRO considering the emission value. However, our PCRO algorithm obtains a solution with minimum fuel cost and loss values, that is, the solution provided by PCRO is non-dominated by the results of other algorithms. It can be concluded from Tables VII and VIII that the proposed PCRO is efficient for solving the given problem.

To evaluate the closeness of the obtained Pareto front to the true Pareto front which is unknown in advance, the two-set-coverage (SC) [8] is also adopted. Table IX gives the comparison results among the five compared algorithms in terms of SC. It can be seen from this table that: 1) At the worst case, nearly $9\,\%$ solutions obtained by PCRO are dominated by BB-MOPSO. However, more than $30\,\%$ the solutions obtained by BB-MOPSO are dominated by the solutions of PCRO, which shows the superior performance of PCRO; 2) Comparing with SMOPSO, the solutions obtained by PCRO dominate by more than $47\,\%$ the solutions by SMOPSO; 3) In comparison with CMOPSO and TV-MOPSO, the dominant solution rates are about $43\,\%$ and $82\,\%$, respectively. Thus, the proposed PCRO algorithm is better than the four compared algorithms in terms of the convergence performance.

C. Performance for Case 3

To further verify the performance of PCRO, in this section, we take a four-area test system, which has four generators in each area with different fuel and emission characteristics. Table X gives the problem parameters. The transmission cost is not considered in simulations since it is normally small as compared with the total fuel costs. The compared algorithms include MOPSO [19], TLBO [20], TV-MOPSO [31], and BB-MOPSO [8]. All the compared algorithms are implemented in the same environment to solve the given four-area test system. The parameters for the compared algorithms are set as the same with their references.

The minimum fuel costs and minimum emissions obtained by the five compared algorithms are shown in Tables XI and XII, respectively. As shown in Table XI, our PCRO algorithm obtained a solution with the minimum fuel cost 1984.3, which is obviously better than the other four compared algorithms. Table XII also shows that our PCRO algorithm can obtain a better solution with minimum emission, which is also obviously better than the other compared algorithms. From the above analysis, we can conclude that PCRO is superior than the other four compared algorithms.

Table XIII gives the comparison results among the five compared algorithms in terms of SC for Case 3. It can be seen from this table that: 1) At the worst case, nearly $5\,\%$ solutions obtained by PCRO are dominated by BB-MOPSO. However, more than $40\,\%$ the solutions obtained by BB-MOPSO are

TABLE X
COEFFICIENTS OF FUEL COST, POLLUTION EMISSIONS (P.U.) AND GENERATOR CAPACITIES FOR CASE 3

Generator	a_{ij}	b_{ij}	c_{ij}	α_{ij}	β_{ij}	γ_{ij}	ξ_{ij}	λ_{ij}	$P_{G_{ij}}^{\min}$	$P_{G_{ij}}^{\max}$	P_D
$G_{1,1}$	150	189	0.50	0.016	−1.500	23.333	2.0×10^{-4}	2.122	0.0005	0.14	1.563
$G_{1,2}$	115	200	0.55	0.031	−1.820	21.022	5.0×10^{-4}	1.233	0.0005	0.10	
$G_{1,3}$	40	350	0.60	0.013	−1.249	22.050	1.0×10^{-6}	6.000	0.0005	0.13	
$G_{1,4}$	122	315	0.50	0.012	−1.355	22.983	1.0×10^{-3}	1.523	0.0005	0.12	
$G_{2,1}$	125	305	0.50	0.020	−1.900	21.313	1.0×10^{-6}	8.000	0.0005	0.25	
$G_{2,2}$	70	275	0.70	0.007	0.805	21.900	3.0×10^{-5}	5.167	0.0005	0.12	
$G_{2,3}$	70	345	0.70	0.015	−1.401	23.001	2.0×10^{-4}	3.857	0.0005	0.20	
$G_{2,4}$	70	345	0.70	0.018	−1.800	24.003	2.0×10^{-4}	3.333	0.0005	0.18	
$G_{3,1}$	130	245	0.50	0.019	−2.000	25.121	1.0×10^{-6}	7.000	0.0005	0.30	
$G_{3,2}$	130	245	0.50	0.012	−1.360	22.990	2.0×10^{-3}	3.000	0.0005	0.30	
$G_{3,3}$	135	235	0.55	0.033	−2.100	27.010	1.0×10^{-6}	6.000	0.0005	0.30	
$G_{3,4}$	200	130	0.45	0.018	−1.800	25.101	1.0×10^{-5}	1.667	0.0005	0.30	
$G_{4,1}$	70	345	0.70	0.018	−1.810	24.313	2.0×10^{-4}	3.857	0.0005	0.11	
$G_{4,2}$	45	389	0.60	0.030	−1.921	27.119	5.0×10^{-4}	5.233	0.0005	0.20	
$G_{4,3}$	75	355	0.60	0.020	−1.200	30.110	1.0×10^{-6}	4.000	0.0005	0.30	
$G_{4,4}$	100	370	0.80	0.040	−1.400	22.500	2.0×10^{-3}	3.000	0.0005	0.30	

TABLE XI
BEST SOLUTIONS FOR COST WITH FIVE ALGORITHMS WHEN OPTIMIZING CASE 3

Generator	MOPSO	TLBO	TV-MOPSO	BB-MOPSO	PCRO
$G_{1,1}$	0.139926	0.139982	0.13899	0.139905	0.139946
$G_{1,2}$	0.099853	0.099803	0.099833	0.099952	0.099991
$G_{1,3}$	0.002959	0.025094	0.030394	0.005454	0.0005
$G_{1,4}$	0.052366	0.051502	0.060031	0.063758	0.000618
$G_{2,1}$	0.03127	0.010464	0.055248	0.070681	0.000509
$G_{2,2}$	0.110544	0.10667	0.027646	0.033566	0.118505
$G_{2,3}$	0.000501	0.077181	0.005309	0.000971	0.0005
$G_{2,4}$	0.089786	0.023774	0.037031	0.025494	0.0005
$G_{3,1}$	0.299449	0.299951	0.299866	0.299866	0.299872
$G_{3,2}$	0.229105	0.248026	0.248978	0.214019	0.299997
$G_{3,3}$	0.159958	0.175195	0.259944	0.293495	0.299965
$G_{3,4}$	0.299342	0.299998	0.291186	0.296869	0.3
$G_{4,1}$	0.004985	0.001582	0.001011	0.011492	0.000501
$G_{4,2}$	0.000501	0.002745	0.004655	0.004655	0.0005
$G_{4,3}$	0.040074	0.0005	0.000587	0.000587	0.000503
$G_{4,4}$	0.002382	0.000532	0.002292	0.002238	0.0005
Fuel cost	2005.21	2002.35	1998.64	1995.8	**1984.3**
Emission	0.06352	0.066351	0.069858	0.071642	0.087263

dominated by the solutions of PCRO, which shows the superior performance of PCRO; 2) The solutions obtained by PCRO dominate by more than 50 % the solutions by MOPSO; 3) In comparison with TLBO and TV-MOPSO, the dominant solution rates are about 50 % and 87 %, respectively. Thus, the proposed PCRO algorithm is better than the four compared algorithms in terms of the convergence performance with respect to optimizing the four-area test problem.

VI. CONCLUSION

This paper presents an improved chemical-reaction optimization algorithm for solving the multi-area environmental/economic dispatch optimization problems. From the experimental comparison results, we can see that the proposed PCRO is efficient for solving the MAEED problems, the main reasons are as follows: 1) In the PCRO algorithm, the improved four elementary reactions, i.e., on-wall ineffective collision, inter-molecular ineffective collision, decomposition, and synthesis, can increase the local and global search abilities of the algorithm; 2) The encoding mechanism can dynamically enhance the performance of the proposed algorithm; 3) The five neighborhood structures and the self-adaptive neighborhood structure selection mechanism further improve the local search ability while maintaining population diversity; 4) The grid-based crowding distance strategy can obviously enable the algorithm to easily converge near the Pareto front; 5) The kinetic-energy-based search procedure further enhances the global search ability.

In future work, we will apply the proposed PCRO algorithm to solve other multi-objective problems in realistic energy conversion systems, and also to develop more efficient algorithm considering self-adaptive strategies similar as [34].

TABLE XII
BEST SOLUTIONS FOR EMISSION WITH FIVE ALGORITHMS WHEN OPTIMIZING CASE 3

Generator	MOPSO	TLBO	TV-MOPSO	BB-MOPSO	PCRO
$G_{1,1}$	0.102099	0.138962	0.137804	0.122113	0.10281
$G_{1,2}$	0.022984	0.019904	0.073998	0.019401	0.099945
$G_{1,3}$	0.042862	0.106588	0.083665	0.038608	0.108018
$G_{1,4}$	0.054979	0.054159	0.033498	0.027472	0.101981
$G_{2,1}$	0.107218	0.126632	0.080965	0.238099	0.125076
$G_{2,2}$	0.07819	0.0957	0.035857	0.011861	0.060398
$G_{2,3}$	0.184189	0.093039	0.030348	0.13827	0.098183
$G_{2,4}$	0.093428	0.025588	0.152	0.115749	0.108088
$G_{3,1}$	0.079913	0.186599	0.154764	0.196559	0.100646
$G_{3,2}$	0.171853	0.105365	0.246422	0.101207	0.08787
$G_{3,3}$	0.19304	0.133821	0.151412	0.170449	0.102841
$G_{3,4}$	0.049633	0.052267	0.060631	0.174155	0.102338
$G_{4,1}$	0.09496	0.042656	0.100196	0.067332	0.108082
$G_{4,2}$	0.134719	0.182205	0.06064	0.067166	0.090858
$G_{4,3}$	0.0675	0.101618	0.044723	0.059173	0.076017
$G_{4,4}$	0.085432	0.097897	0.116078	0.015387	0.089851
Fuel cost	2107.55	2105.25	2084	2071.47	**2098.88**
Emission	0.034481	0.034796	0.036498	0.038289	**0.023902**

TABLE XIII
STATISTICAL RESULTS OF THE METRIC SC FOR CASE 3

	PCRO	BB-MOPSO	MOPSO	TLBO	TV-MOPSO
SC(PCRO, $*$)	–	0.4151	0.5123	0.5012	0.8721
SC(BB-MOPSO, $*$)	0.0571	–	0.3795	0.3015	0.6915
SC(MOPSO, $*$)	0.0121	0.0378	–	0.1928	0.4357
SC(TLBO, $*$)	0.0232	0.0515	0.2315	–	0.5691
SC(TV-MOPSO, $*$)	0	0	0.1488	0.0815	–

REFERENCES

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

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

[3] M. A. Abido, "Multiobjective evolutionary algorithms for electric power dispatch problem," *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, pp. 315–329, Jun. 2006.

[4] M. A. Abido, "A novel multiobjective evolutionary algorithm for environmental/economic power dispatch," *Electr. Power Syst. Res.*, vol. 65, no. 1, pp. 71–81, Apr. 2003.

[5] M. A. Abido, "A niched Pareto genetic algorithm for multiobjective environmental/economic dispatch," *Int. J. Electr. Power Energy Syst.*, vol. 25, no. 2, pp. 97–105, Feb. 2003.

[6] R. T. F. A. King, H. C. S. Rughooputh, and K. Deb, "Evolutionary multi-objective environmental/economic dispatch: Stochastic versus deterministic approaches," in *Proc. 3rd Int. Conf. Evolutionary Multi-Criterion Optimization*, Berlin Heidelberg, Germany, 2005, pp. 677–691.

[7] S. Agrawal, B. K. Panigrahi, and M. K. Tiwari, "Multiobjective particle swarm algorithm with fuzzy clustering for electrical power dispatch," *IEEE Trans. Evol. Comput.*, vol. 12, no. 5, pp. 529–41, Oct. 2008.

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

[9] Z. H. Xia, X. H. Wang, X. M. Sun, and Q. Wang, "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data," *IEEE Trans. Parall. Distrib. Syst.*, vol. 27, no. 2, pp. 340–352, Feb. 2015.

[10] Z. J. Fu, K. Ren, J. G. Shu, X. M. Sun, and F. X. Huang, "Enabling personalized search over encrypted outsourced data with efficiency improvement," *IEEE Trans. Parall. Distrib. Syst.*, vol. 27, no. 9, pp. 2546–2559, Sep. 2016.

[11] P. Guo, J. Wang, B. Li, and S. Lee, "A variable thresholdvalue authentication architecture for wireless mesh networks," *J. Internet Technol.*, vol. 15, no. 6, pp. 929–936, 2014.

[12] Z. J. Fu, X. M. Sun, Q. Liu, L. Zhou, and J. G. Shu, "Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing," *IEICE Trans. Commun.*, vol. E98.B, no. 1, pp. 190–200, Jan. 2015.

[13] Y. J. Ren, J. Shen, J. Wang, J. Han, and S. Y. Lee, "Mutual verifiable provable data auditing in public cloud storage," *J. Internet Technol.*, vol. 16, no. 2, pp. 317–323, Mar. 2015.

[14] T. H. Ma, J. J. Zhou, M. L. Tang, Y. Tian, A. Al-Dhelaan, M. Al-Rodhaan, and S. Lee, "Social network and tag sources based augmenting collaborative recommender system," *IEICE Trans. Inf. Syst.*, vol. E98.D, no. 4, pp. 902–910, Apr. 2015.

[15] J. Y. Li, M. K. Qiu, Z. Ming, G. Quan, X. Qin, and Z. H. Gu, "Online optimization for scheduling preemptable tasks on IaaS cloud systems," *J. Parall. Distrib. Comp.*, vol. 72, no. 5, pp. 666–677, May 2012.

[16] V. H. Quintana, R. Lopez, R. Romano, and V. Valadez, "Constrained economic dispatch of multi-area systems using the Dantzig-Wolfe decomposition principle," *IEEE Trans. Power Appar. Syst.*, vol. PAS-100,

no. 4, pp. 2127−37, Apr. 1981.

[17] D. Streiffert, "Multi-area economic dispatch with tie line constraints," *IEEE Trans. Power Syst.*, vol. 10, no. 4, pp. 1946−51, Nov. 1995.

[18] C. Wang and S. M. Shahidehpour, "A decomposition approach to nonlinear multi-area generation scheduling with tie-line constraints using expert systems," *IEEE Trans. Power Syst.*, vol. 7, no. 4, pp. 1409−1418, Nov. 1992.

[19] L. F. Wang and C. Singh, "Reserve-constrained multiarea environmental/economic dispatch based on particle swarm optimization with local search," *Eng. Appl. Artif. Intell.*, vol. 22, no. 2, pp. 298−307, Mar. 2009.

[20] M. Basu, "Teaching-learning-based optimization algorithm for multi-area economic dispatch," *Energy*, vol. 68, pp. 21−28, Apr. 2014.

[21] A. Y. S. Lam, J. L. Xu, and V. O. K. Li, "Chemical reaction optimization for population transition in peer-to-peer live streaming," in *Proc. IEEE Congr. Evolutionary Computation (CEC)*, Barcelona, Spain, 2010, pp. 1−8.

[22] A. Y. S. Lam and V. O. K. Li, "Chemical-reaction-inspired metaheuristic for optimization," *IEEE Trans. Evol. Comput.*, vol. 14, no. 3, pp. 381−399, Jun. 2010.

[23] A. Lam, V. O. K. Li, and J. J. Q. Yu, "Real-coded chemical reaction optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 3, pp. 339−353, Jun. 2012.

[24] A. Y. S. Lam and V. O. K. Li, "Chemical reaction optimization: a tutorial," *Memetic Comp.*, vol. 4, no. 1, pp. 3−17, Mar. 2012.

[25] M. A. Abido, "Multiobjective particle swarm optimization for environmental/economic dispatch problem," *Electr. Power Syst. Res.*, vol. 79, no. 7, pp. 1105−1113, Jul. 2009.

[26] J. Li, Q. Pan, and S. Xie, "An effective shuffled frog-leaping algorithm for multi-objective flexible job shop scheduling problems," *Appl. Math. Comput.*, vol. 218, no. 18, pp. 9353−9371, 2012.

[27] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182−197, Apr. 2002.

[28] S. X. Yang, M. Q. Li, X. H. Liu, and J. H. Zheng, "A grid-based evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 5, pp. 721−736, Oct. 2013.

[29] Q. K. Pan, M. F. Tasgetiren, P. N. Suganthan, and T. J. Chua, "A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem," *Inf. Sci.*, vol. 181, no. 12, pp. 2455−2468, Jun. 2011.

[30] J. Q. Li, S. C. Bai, P. Y. Duan, H. Y. Sang, Y. Y. Han, and Z. X. Zheng, "An improved artificial bee colony algorithm for addressing distributed flow shop with distance coefficient in a prefabricated system," *Int. J. Prod. Res.*, 2018, doi: 10.1080/00207543.2019.1571687.

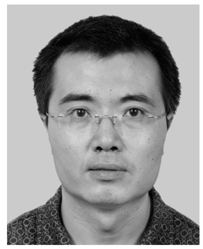
[31] P. K. Tripathi, S. Bandyopadhyay, and S. K. Pal, "Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients," *Inf. Sci.*, vol. 177, no. 22, pp. 5033−5049, Nov. 2007.

[32] S. Mostaghim and J. Teich, "Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO)," in *Proc. IEEE Swarm Intelligence Symp.*, Indianapolis, IN, USA, 2003, pp. 26−33.

[33] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 256−279, Jun. 2004.

[34] J. Zhao, S. X. Liu, M. C. Zhou, X. W. Guo, and Q. Liang, "Modified cuckoo search algorithm to solve economic power dispatch optimization problems," *IEEE/CAA J. Autom. Sinica*, vol. 5, no. 4, pp.794−806, Mar. 2018.

Junqing Li (M'08) received the master degree of computer science and technology in 2004 from Shandong Economic University, Shandong, China, and Ph.D. degree in 2016 from Northeastern University, Shenyang, China. Since 2004, he has been with the School of Computer Science Department, Liaocheng University, where he became an Associate Professor in 2008. He is also with the School of Information Science and Engineering, Shandong Normal University, where he became a Professor in 2017. His current research interests include intelligent optimization and scheduling. He has authored more than 60 refereed papers.



Quanke Pan received the B.Sc. degree and the Ph.D. degree from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 1993 and 2003, respectively. Since 2003, he has been with the School of Computer Science Department, Liaocheng University, where he became a Full Professor in 2006. He has been with the State Key Laboratory of Synthetical Automation for Process Industries (Northeastern University) since 2011. He is now with the School of Mechatronic Engineering and Automation, Shanghai University. His current research interests include intelligent optimization and scheduling. He has authored more than 200 refereed papers.

Peiyong Duan received the B.Sc. degree in 1996 from Shandong University of Technology (merged into Shandong University in 2000), Shandong, China, and Ph.D. degree in 1999 from Shanghai Jiao Tong University, Shanghai, China. From 1999 through 2014, he was with the School of Information and Electrical Engineering, Shandong Jianzhu University where he is appointed as an Associate Professor and a Full Professor in 1999 and 2002, respectively. Since 2014, he has been with the School of Computer Science, Liaocheng University, Shandong, China. Since 2017, he has been with the School of Information Science and Engineering, Shandong Normal University. His current research interests include discrete optimization and scheduling. He has authored more than 60 refereed papers.

Hongyan Sang received the master degree of computer science and technology in 2010 from Liaocheng University, Shandong, China, and Ph.D. degree in 2013 from Huazhong University of Science Technology, Wuhan China. Since 2003, she has been with the School of Computer Science Department, Liaocheng University, where she became an Associate Professor in 2016. Her current research interests include intelligent optimization and scheduling. She has authored more than 30 refereed papers.

Kaizhou Gao received the bachelor and master degrees from the Computer School of Liaocheng University and School of Information and Engineering of Yangzhou University, in 2005 and 2008, respectively. He received the Ph.D. degree from Nanyang Technological University, in 2016. He is now with the Macau Institute of Systems Engineering, Macau University of Science and Technology. His current research interests include intelligent optimization, heuristics and meta-heuristics, planning and scheduling, and intelligent traffic.