Quasi-oppositional Cuckoo Search Algorithm for Multi-objective Optimal Power Flow

Gonggui Chen, *Member, IAENG,* Siyuan Qiu, Zhizhong Zhang, Zhi Sun

***Abstract*—This paper introduces the multi-objective cuckoo search (MOCS) method for solving the multi-objective optimal power flow (MOOPF) which is a multi-variable, multi-constraint and nonlinear programming problem. In order to speed up convergence and enhance the quality of solution, the quasi-opposition based learning mechanism is adopted to propose the multi-objective quasi-oppositional cuckoo search (MOQOCS) algorithm in this paper. In MOCS and MOQOCS methods, the feasibility-prior domination principle (FDP) is presented to ensure the feasibility of simulation result which is based on the objective values and the sum of constraint violation values. The crowding-distance sorting is considered to enhance the diversity of Pareto-optimal solutions and obtain better distributed Pareto optimal front. IEEE 30-bus and IEEE 57-bus systems have been considered to check the performance of the proposed methods. The simulation results confirm that MOQOCS algorithm obtains superior compromise solution and produces better distributed Pareto optimal front compared to other methods.**

***Index Terms*—Multi-Objective Optimal Power Flow (MOOPF), Cuckoo search, Multi-Objective Quasi-Oppositional Cuckoo Search (MOQOCS), Feasibility-prior Domination Principle (FDP), Pareto optimal**

I. Introduction

THE primary purpose of optimal power flow (OPF) is to search the optimal load flow distribution which can satisfy the system constraints and minimize the selected objective, through optimal calculation to adjust the available control variables within the limits [1-3]. OPF has been a crucial problem for modern power system, which has attracted the attention of many scholars in recent decades. The selected objective is usually to minimization fuel cost of all generators for OPF problem. However, the voltage stability and power losses are also becoming more and more important due to the increasing demand for electricity and more complex power system. In this situation, the multi-objective OPF (MOOPF) is necessary to be considered which optimizes multiple objectives simultaneously.

The MOOPF is a multi-variable, multi-constraint and nonlinear optimization problem, which is difficult to accurately determine a best solution because of the conflict of different objectives [4, 5]. At present, numerous intelligent algorithms have been used to solve the MOOPF problem, which are usually divided into two types. The first approach is that many researchers transform the different objectives into a single objective function by adding weight values to each objective. Abaci and Yamacli [6] proposed the differential search algorithm (DSA) for solving MOOPF problem. The DSA method was implemented in three different test systems with single-objective and multi-objective optimization. The multi-objective adaptive immune algorithm (MOAIA) was presented by Xiong et al [7]. The power loss, voltage stability margin and voltage deviation were merged into an overall objective by weight coefficients and results validated the improved performance of MOAIA. Chaib et al. [8] presented backtracking search algorithm (BSA) and applied it for 16 different cases on three standard electric systems. In that paper, BSA approach has superior performance in most cases than other well-known approaches. However, these methods just obtain one optimal solution through running the program one time. If the demand of the decision maker is changed, these approaches require rerun the program and consume a lot of computational time.

Another approach is based on the non-domination principle and evolution algorithm which can obtain a set of Pareto-optimal solutions. Deb et al. [9] proposed the non-dominated sorting genetic algorithm II (NSGA-II) with the non-dominated sorting method and multi-criterion decision making. The results demonstrate that NSGA-II can obtain the solutions of better spread and superior convergence compared to PAES and SPEA methods in most cases. Sivasubramani et al. [10] presented the multi-objective harmony search (MOHS) algorithm for solving the MOOPF problem where cost, power losses and *L*-index were used to form the multi-objective optimization problems. The simulation results of MOHS have better distributed solutions compared to the NSGA-II approach. The multi-objective modified imperialist competitive algorithm (MOMICA) was presented and successfully applied to MOOPF problem by Ghasemi et al [11]. And the comparison of MOMICA algorithm with other methods indicated the superiority of the presented method. Recently, cuckoo search (CS) algorithm is

Manuscript received September 2, 2017; revised December 2, 2017. This work was supported in part by the Innovation Team Program of Chongqing Education Committee (CXTDX201601019), Chongqing University Innovation Team under Grant KJTD201312 and the National Natural Science Foundation of China (Nos. 51207064 and 61463014).

Gonggui Chen is with the Key Laboratory of Network control & Intelligent Instrument (Ministry of Education), Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: chenggpower@126.com).

Siyuan Qiu is with the Key Laboratory of Complex Systems and Bionic Control, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: qiusypower@163.com).

Zhizhong Zhang is with the Key Laboratory of Communication Network and Testing Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (corresponding author, e-mail: zhangzztx@163.com).

Zhi Sun is with the Guodian Enshi Hydropower Development, Enshi 445000, China (e-mail: sunzhi24@126.com).

presented and reported to outperform many well-known algorithms, which is inspired by the breeding parasitic characteristics of cuckoo and combined with the Lévy flights behavior [12]. For solving the MOOPF problem and obtain better Pareto optimal solutions, this paper presents multi-objective cuckoo search (MOCS) method. Most recently, CS method has been used to solve various realistic problems such as bin packing problem [13], large-scale antenna array for 5G beamforming [14], traffic signal controllers [15], hyperspectral image classification [16], planar graph coloring problem [17], segmenting satellite images [18], and multi-object optimization problems [19-21]. However, these improved CS algorithms have not been used for effectively solving MOOPF problem, which motivates us to modify the CS algorithm and apply the method to this proposed field. To speed up convergence speed and enhance searching ability of CS method, quasi-oppositional based learning is introduced to the MOCS method and propose multi-objective quasi-oppositional cuckoo search (MOQOCS) algorithm. In MOCS and MOQOCS methods, non-dominated sorting is considered to select the higher quality solutions, and crowding distance sorting is considered to enhance the diversity and obtain better distributed Pareto optimal front. To ensure the feasibility of simulation result, this paper presents the feasibility-prior domination principle (FDP) based on the sum of constraint violation values, which combines the constraint handing method with domination rule to determine the dominance relation of different solutions.

In this paper, the proposed MOCS and MOQOCS methods are examined in IEEE 30-bus and IEEE 57-bus systems with different object function. In order to evaluate the performance and effectiveness, the results of MOQOCS approach are compared with those solutions of MOCS and MOPSO approaches, which demonstrate that MOQOCS algorithm can obtain smaller best compromise solution and better distributed Pareto optimal front.

The rest of this paper is organized as follows: Section 2 presents the problem formulation of MOOPF problem. Section 3 describes the structure of CS method and the quasi-opposition-based learning mechanism. Next, the evolutionary mechanism of MOQOCS method is explained in Section 4, which presents the calculation process of MOQOCS algorithm for solving MOOPF problem. Section 5 tests the proposed MOCS and MOQOCS methods on IEEE 30-bus and 57-bus systems and describes the simulation results. Finally, Section 6 gives the conclusions.

II. MOOPF Problem

The mathematical model of OPF consists of objective function and various system constraints. The objective function can be fuel cost, voltage deviation and power losses, etc. The system constraints are composed of many equality and inequality constraints. Therefore, OPF is a complicated nonlinear problem and can be formulated as below [22]:

$$OF = \min f(x,u) \tag{1}$$

$$\text{Subject to: } g(x,u) = 0 \tag{2}$$

$$h(x,u) \le 0 \tag{3}$$

In the above formulation, $f(x, u)$ represents the chosen objective function; x denotes a vector composed of state variables; u denotes a vector composed of control variables; $g(x, u)$ and $h(x, u)$ represent those equal constraints and unequal constraints.

In many actual situations, the OPF problem of considering a single objective of fuel cost can not meet the system demand. Therefore, the multi-objective OPF problem is formulated to satisfy multiple objectives simultaneously, which can be expressed as:

$$OF = \min\left[f_1(x,u), f_2(x,u), \cdots, f_M(x,u) \right] \tag{4}$$

where $f_i(x, u)$ indicates the ith objective function; M indicates the number of optimal objectives. The different objectives of MOOPF are often conflict, because the performance of an objective may decrease when the performance of another objective is improved. It is impossible to make multiple objectives optimal simultaneously and obtain the optimal solution. Thus a collection of compromise solutions is necessary, which are Pareto optimal solutions. Generally, solution U_1 dominates solution U_2 (denoted by $U_1 \prec U_2$) only if both of the below conditions are satisfied [1]:

$$\begin{aligned} & f_i(U_1) \le f_i(U_2), \quad \forall i \in \{1,2,\cdots,M\} \\ & f_j(U_1) < f_j(U_2), \quad \exists j \in \{1,2,\cdots,M\} \end{aligned} \tag{5}$$

The solution U is regarded as Pareto-optimal solution if there isn't another solution dominating U in the whole population. For solving MOOPF problem, the main purpose is to get a set of Pareto-optimal solutions. In addition, the detailed description for the relevant concepts of Pareto optimization method can refer to [23, 24].

A. Objective Functions

1) Minimization of Fuel Cost

For solving OPF problem, the general optimal objective is the minimization of fuel cost associated with generator active power that can be represented as:

$$f_{cost} = \sum_{i=1}^{N_G} \left(a_i + b_i P_{Gi} + c_i P_{Gi}^2 \right) \tag{6}$$

where P_{Gi} indicates the real power output of the ith generator; a_i, b_i and c_i indicate the cost coefficients of the ith generator; N_G indicates the number of total generators.

2) Minimization of Active Power Losses

In this case the active power losses are considered as the objective function, which can be formulated as [4]:

$$f_{loss} = \sum_{k=1}^{N_{TL}} g_k \left(|V_i|^2 + |V_j|^2 - 2|V_i||V_j|\cos\delta_{ij} \right) \tag{7}$$

where V_i and V_j respectively indicate the voltage value of bus i and bus j; g_k indicates conductance of line k connected between the ith and jth bus and $i \ne j$; δ_{ij} indicates the phase difference among bus i and bus j; N_{TL} represents the amount of transmission lines.

3) Minimization of Voltage Magnitude Deviation

For the OPF problem of power system, bus voltage is a very important safety indicator. The objective function f_{cost} can make the fuel cost optimal, but the bus voltage profile of the best solution may be undesirable. So the voltage magnitude deviation should be reduced as much as possible from the base 1.0 in p.u. to improve the voltage stability. The objective is to make the sum of voltage deviations minimum, which can be represented by [4]:

$$f_{VD} = \sum_{i=1}^{N_{PQ}} \left| V_i - 1.0 \right| \tag{8}$$

where N_{PQ} indicates the number of PQ buses.

B. Multi-objective Optimization

1) Minimization of f_{cost} and f_{loss}

For minimizing the fuel cost and active power losses simultaneously, the multi-objective function F_1 is considered to solve the MOOPF problem in this study, which can be shown as:

$$F_1 = \min\left[f_{cost}(x,u), f_{loss}(x,u) \right] \tag{9}$$

2) Minimization of f_{cost} and f_{VD}

The multi-objective function F_2 is adopted for optimizing the fuel cost and voltage magnitude deviation simultaneously, and the corresponding function can be formulated as:

$$F_2 = \min\left[f_{cost}(x,u), f_{VD}(x,u) \right] \tag{10}$$

C. System Constraints

MOOPF is a large-scale nonlinear problem which requires satisfying various equal constraints and unequal constraints as follows:

1) Equality Constraints

In MOOPF model, the equal constraints consist of the active and reactive power load flow equations, which can be given as [25]:

$$P_{Gi} - P_{Di} - V_i \sum_{j=1}^{N_B} V_j \left(G_{ij} \cos\delta_{ij} + B_{ij} \sin\delta_{ij} \right) = 0, \quad i \in N_B \tag{11}$$

$$Q_{Gi} - Q_{Di} - V_i \sum_{j=1}^{N_B} V_j \left(G_{ij} \sin\delta_{ij} - B_{ij} \cos\delta_{ij} \right) = 0, \quad i \in N_B \tag{12}$$

where N_B represents the amount of all system buses; P_{Gi} and P_{Di} indicate the injected active power and active load demand on bus i; Q_{Gi} and Q_{Di} indicate the injected reactive power and reactive load demand on bus i; G_{ij} and B_{ij} respectively indicate the real part and imaginary part of the ijth element of the node admittance matrix; δ_{ij} indicates the voltage phase difference between the ith and jth buses [26].

2) Inequality Constraints

In the MOOPF model, the unequal constraints are described as follows [27]:

i. Generator constraints: generator voltage, active power and reactive power of the generator are limited by their minimum and maximum limits:

$$V_{Gi}^{\min} \le V_{Gi} \le V_{Gi}^{\max}, \quad i = 1,2,\ldots,N_G \tag{13}$$

$$P_{Gi}^{\min} \le P_{Gi} \le P_{Gi}^{\max}, \quad i = 1,2,\ldots,N_G \tag{14}$$

$$Q_{Gi}^{\min} \le Q_{Gi} \le Q_{Gi}^{\max}, \quad i = 1,2,\ldots,N_G \tag{15}$$

ii. Transformer taps constraints:

$$T_i^{\min} \le T_i \le T_i^{\max}, \quad i = 1,2,\ldots,N_T \tag{16}$$

where N_T represents the number of transformer branches.

iii. Shunt VAR compensator constraints:

$$Q_{Ci}^{\min} \le Q_{Ci} \le Q_{Ci}^{\max}, \quad i = 1,2,\ldots,N_C \tag{17}$$

where N_C represents the number of reactive compensators.

iv. Security constraints:

$$V_{Li}^{\min} \le V_{Li} \le V_{Li}^{\max}, \quad i = 1,2,\ldots,N_{PQ} \tag{18}$$

$$S_{Li} \le S_{Li}^{\max}, \quad i = 1,2,\ldots,N_{TL} \tag{19}$$

where V_L represents the voltage at load bus; S_L represents the transmission line loading; N_{PQ} and N_{TL} indicate the number of PQ buses and transmission lines.

III. QOCS Algorithm

A. Overview of CS Algorithm

The CS is a novel heuristic algorithm which is inspired by the breeding parasitic characteristics of cuckoo and combined with the Lévy flights behavior. It is worth mentioning that the host may find that the egg is not its own with a probability $p_a \in [0, 1]$, and the host will abandon the invasive egg from the nest or form a new nest on this situation. For establishing the mathematic model of CS algorithm, three idealized assumptions were used: i) every cuckoo can only lay one egg in a randomly selected nest for one time; ii) the superior nests with better eggs will be retained to next generation; iii) the number of nests are invariant during the whole search process [28, 29].

In CS algorithm, an egg is regarded as a candidate solution. Let $U_i(k)$ denote the ith solution (for i = 1, 2,…, NP) at kth iteration. In the initial process of the cuckoo search algorithm, each solution is randomly generated within the range of the specified boundaries. When generating new solution $U_i(k+1)$ of the ith cuckoo at (k+1)th iteration, the Lévy flight is performed as follows:

$$U_i(k+1) = U_i(k) + \alpha \oplus L\acute{e}vy(\lambda) \tag{20}$$

where $\alpha > 0$ denotes the step size and usually considered to be 1; the special symbol $\oplus$ denotes the entry wise multiplication. The Lévy flight follows the random walk, which can be defined according to the Lévy distribution as bellow:

$$L\acute{e}vy(\lambda) \sim u = t^{-\lambda}, \quad (1 < \lambda \le 3) \tag{21}$$

This is a stochastic equation of heavy tailed probability distribution with an infinite variance. In this form of walking, it may be short distance step and occasionally a long step. In the process of exploring a space with wide range, Lévy flight is greatly efficient to global search. And the $L\acute{e}vy(\lambda)$ can be specifically calculated as follow [30, 31]:

$$Levy(\lambda) \sim \frac{\mu}{|v|^{1/\beta}} \tag{22}$$

$$\sigma_\mu = \left\{ \frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma\left[(1+\beta)/2\right]\beta 2^{(\beta-1)/2}} \right\}^{1/\beta}, \sigma_v = 1 \tag{23}$$

where μ and v are random values and obey the normal distribution; Γ is the standard Gamma function and β is a parameter usually taken as 1.5. Therefore, the update formula of CS method can be calculated as:

$$V_i(k) = U_i(k) + \alpha_0 \cdot \frac{\mu}{|v|^{1/\beta}} \left(U_i(k) - U_{best} \right) \tag{24}$$

where α_0 is the step size scaling factor; U_{best} indicates the current best solution.

After producing the new solution $V_i(k)$, the CS will use the greedy strategy to select the better solution recorded as $V_i(k)$ according to their objective function values. The last operation in CS method can be seen as the replacement strategy by discovering a new solution, which is formulated as:

$$U_i(k+1) = \begin{cases} V_i(k) + rand \cdot (U_{r1} - U_{r2}), & rand < p_a \\ V_i(k), & otherwise \end{cases} \quad (25)$$

where U_{r1} and U_{r2} are two randomly selected solutions. If the objective function of V_i is smaller than $U_i(k+1)$, V_i is regarded as the next generation solution, otherwise $U_i(k+1)$ would remain unchanged.

B. Quasi-opposition-based Learning

Opposition-based learning (OBL) mechanism can accelerate the convergence and improve the quality of solutions through considering the current solutions and opposite solutions synchronously, which was first presented by Tizhoosh [32]. On the basis of probability theory, the random solution is 50% better than its opposite solution and vice versa. Thus, the superior solution between the two inverse solutions is chosen as the candidate solution which can enhance search efficiency of evolutionary algorithms. The OBL method has been effectively applied to a variety of problems. In order to explain clearly the principle of opposition-based learning, the concepts of opposite number and opposite point are defined in this paper, which are given as follows [33]:

Opposite number: If x is a random number in the search zone $[a, b]$, its opposite number can be expressed as:

$$x^o = a + b - x \quad (26)$$

Opposite point: If $P(x_1, x_2, \ldots, x_i, \ldots, x_d)$ is a point in d-dimensional space where $x_i \in [a_i, b_i]$, its opposite point $OP(x_1^o, x_2^o, \ldots, x_i^o, \ldots, x_d^o)$ may be defined as follows:

$$x_i^o = a_i + b_i - x_i; \quad i = 1, 2, \ldots, d \quad (27)$$

However, it should be pointed out that the OBL has some improvement mechanisms, in which quasi-opposition-based learning (QOBL) has been applied by many researchers and proved to be more effective than OBL [34]. Moreover, we can define the quasi-opposite number and quasi-opposite point as follows:

Quasi-opposite number: The quasi-opposite number x^{qo} of a random number x in the search zone $[a, b]$ can be expressed as:

$$x^{qo} = rand\left[\left(\frac{a+b}{2}\right), (a+b-x)\right] \quad (28)$$

Quasi-opposite point: The quasi-opposite point $QOP(x_1^{qo}, x_2^{qo}, \ldots, x_i^{qo}, \ldots, x_d^{qo})$ in d-dimensional space is calculated as follows:

$$x_i^{qo} = rand\left[\left(\frac{a_i+b_i}{2}\right), (a_i+b_i-x_i)\right]; \quad i = 1, 2, \ldots, d \quad (29)$$

The QOBL can be applied not only to the initialization process, but also to the evolutionary process of CS algorithm for updating the population. In this study, the solution generated by mutation mechanism using Eq. (25) can be replaced by a quasi-opposite solution.

IV. MOQOCS Approach For MOOPF Problem

In this section, the evolutionary process of MOQOCS algorithm for solving MOOPF problem is described in details.

A. Initialization Individuals

For solving the MOOPF problem by MOCS and MOQOCS algorithms, the initial individuals should be randomly generated in the search space which can be represented by a matrix as follows:

$$U = \begin{bmatrix} u_1^1 & u_1^2 & \cdots & u_1^D \\ u_2^1 & u_2^2 & \cdots & u_2^D \\ \vdots & \vdots & \vdots & \vdots \\ u_N^1 & u_N^2 & \cdots & u_N^D \end{bmatrix} \quad (30)$$

where N represents the number of individuals; D indicates the dimension of individual that is the number of control variables. The nest is to create the quasi-opposition-based population of the initial population U using the QOBL mechanism in Section 3.2. Finally, compute the objective function value of all the $2N$ individuals.

B. Selection of Pareto-optimal Solutions and Gbest

For solving MOOPF problem, this paper uses non-dominated sorting and crowding distance sorting developed by Deb et al [9].

1) Non-dominated Sorting

According to non-dominated approach, we can sort all the solutions into different non-dominated levels, and specific operations are described as:

(1) Find the Pareto-optimal solutions in the current generation by Eq. (5) in Section 2, and assign these solutions to the highest rank which is recorded as *rank*=1.

(2) Remove temporarily the solutions of the upper front from the entire population, and generate new Pareto optimal solutions. Repeat step (1) to give the next-best rank which is recorded as *rank*=2.

(3) Repeat step (2) until all the solutions are identified to its non-domination level.

2) Crowding Distance Sorting

The crowing distance is a performance index to estimate the density of solutions which can reflect the distribution of the Pareto-optimal front [10]. The crowding distance of the ith solution is the average distance of the (i-1)th and the (i+1)th solutions on each objective. However, all the objective function needs to be normalized because different objective functions may differ greatly on the value size. So the crowding distance of the ith solution can be expressed as:

$$dis(i) = \sum_{j=1}^{M} \frac{f_j(i+1) - f_j(i-1)}{f_j^{\max} - f_j^{\min}} \quad (31)$$

where M indicates the number of optimal objective functions; $f_j(i)$ indicates the jth optimal objective of the ith solution; $f_j^{\max}$ and $f_j^{\min}$ indicate the largest and smallest values of the jth optimal objective. It's worth pointing out that the population needs sorting for every objective function before crowding-distance sorting. Then, the crowding distance of those two solutions with maximum and minimum function values is infinity.

After calculating the crowding-distance of all solutions, we can generate Pareto-optimal front according to two indices of non-dominant rank and crowding distance. The ith solution is superior to the jth solution if one of the following two formulas is satisfies.

$$rank(i) < rank(j) \tag{32}$$

$$rank(i) = rank(j) \text{ and } dis(i) > dis(j) \tag{33}$$

Finally, we can choose the best N solutions from all $2N$ solutions as the new population. In addition, the gbest is chosen randomly from the Pareto-optimal set to guide the evolution of population.

C. Feasibility-prior Domination Principle (FDP)

The MOOPF problem is a complex nonlinear optimization problem of power system, which has many constraints required to handle. The main problem is to handle the inequality constraints on state variables which include voltage constraint of the buses; output reactive power constraint of generator; output active power constraint of slack bus and apparent power of branch. In this paper, the value of constraint violation on state variables refers to the sum of all constraint violation values if a solution vector U_i violates its inequality constraints, which can be described as:

$$Constr(U_i) = \sum_{j=1}^{N_H} Constr\left[h_j(x,u)\right] \tag{34}$$

Where $h_j(x, u)$ represents the jth inequality constraint of the U_i; N_H represents the number of inequality constraints. If the value of $ConVio(U_i)$ is equal to zero, U_i is a feasible solution and it does not violate the inequality constraints.

It should be noted that the above non-domination method doesn't take into account the constraint problem, so FPD is proposed to modify the non-domination method for obtaining better Pareto-optimal solution. Here, a solution U_1 is considered to dominate another solution U_2 when one of the following conditions can be satisfied:

(1) U_1 is a feasible solution but U_2 is not.

(2) U_1 and U_2 are not feasible solutions and $ConVio(U_1)$ is smaller than $ConVio(U_2)$.

(3) U_1 and U_2 are feasible solutions and U_1 dominates U_2 according to Eq. (5).

The FPD gives a better rank to the feasible solution than the infeasible solution, which enhances the reliability of solutions and the convergence speed to the feasible area. After determining the dominated relationship of all solutions according to FPD, solution U is considered as Pareto optimal solution if no other solution dominates it in current generation.

D. Stopping Criteria

In this study, the iterative procedure of MOCS and MOQOCS methods are stopped when the maximum iteration number is reached. Otherwise update the population and search the better Pareto optimal solutions.

E. Best Compromise Solution

The Pareto-optimal solutions have been obtained after reaching the stop standard. Then, we should select a best compromise solution as the final solution. However, it is unable to judge accurately the quality of those different Pareto-optimal solutions due to the different practical requirements. In this paper, fuzzy theory is applied to solve the vague nature of judgment and determine a best compromise solution. The fuzzy membership $\mu_k(i)$ for the kth objective function of solution i can be defined as [10]:

$$\mu_k(i) = \begin{cases} 1, & f_k \le f_k^{\min} \\ \dfrac{f_k^{\max} - f_k}{f_k^{\max} - f_k^{\min}}, & f_k^{\min} < f_k < f_k^{\max} \\ 0, & f_k \ge f_k^{\max} \end{cases} \tag{35}$$

where $f_i^{\min}$ and $f_i^{\max}$ represent the minimum and maximum values of the kth optimal objective. Every solution has a normalized membership function which is the sum of membership values of all optimal objectives. The membership function of solution i can be expressed as:

$$\mu(i) = \frac{\sum_{k=1}^{M} \mu_k(i)}{\sum_{i=1}^{N}\sum_{k=1}^{M} \mu_k(i)} \tag{36}$$

where N indicates the number of Pareto-optimal solutions; M indicates the number of optimal objectives. Thus, the Pareto optimal solution with the maximum membership $\mu(i)$ can be considered as the best compromise solution.

F. Detailed Steps of MOQOCS Algorithm for MOOPF Problem

This paper is focused on solving the MOOPF problem by using the proposed MOQOCS algorithm, which employs QOBL and FDP concepts to improve the performance of CS method. For applying MOQOCS method to the complex and nonlinear MOOPF problem, the following procedure should be performed:

Step 1: Choose the parameters of MOQOCS algorithm such as population size and maximum iterations.

Step 2: Generate the initial population as described in Section 4.1 and create the corresponding quasi-opposite population.

Step 3: Calculate the values of objective functions and constraint violations for all $2N$ individuals and select N solutions as the initial population.

Step 4: Modify the population according to the proposed MOQOCS algorithm. Calculate the optimal objectives of the current population and obtain the value of constraint violations.

Step 5: For the parent population and current population, calculate non-domination level and crowding distance. Then, choose the best N solution according to the constraint-dominated sorting and crowding distance sorting.

Step 6: The Gbest solution is chosen randomly from the Pareto-optimal set to guide the evolution of population.

Step 7: If the maximum cycle number is satisfied, stop the iteration and record the Pareto optimal solutions, otherwise go back to Step 4.

Step 8: The N optimal solutions obtained finally is the Pareto-optimal set which can form the Pareto front of the MOOPF problem.

Step 9: Determine the best compromise solution as described in Section 4.5.

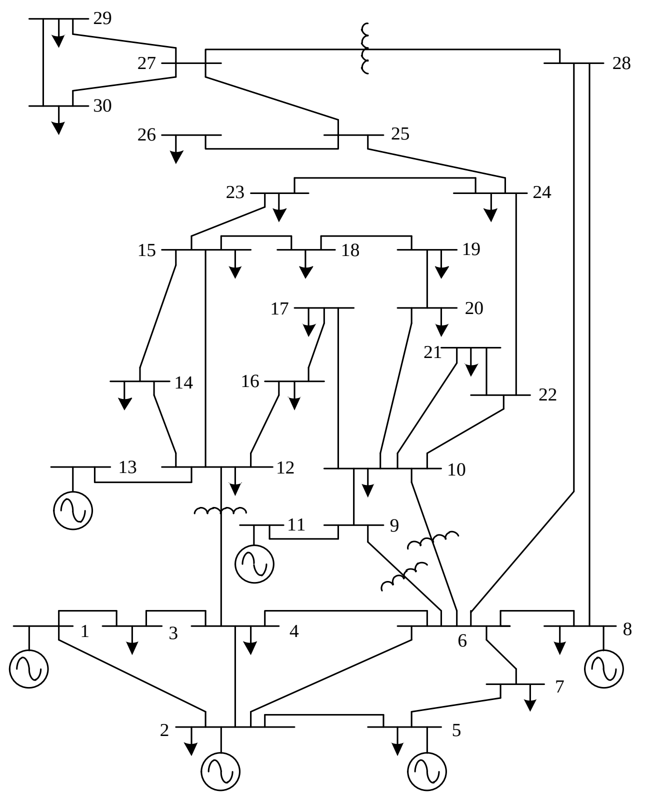

Fig. 1. The system structure diagram of IEEE 30-bus system.

TABLE I
MAIN CHARACTERISTICS OF IEEE 30-BUS SYSTEM

Characteristics	IEEE 30	
	Number	Details
Buses	30	-
Branches	41	-
Generators	6	Buses: 1, 2, 5, 8, 11 and 13
Capacitor banks	9	Buses: 10, 12, 15, 17, 20, 21, 23, 24 and 29
Transformers	4	Branches: 16-9, 6-10, 4-12 and 28-27
Control variables	24	-

TABLE II
CONTROL VARIABLES SETTINGS OF IEEE 30-BUS SYSTEM

Control variables	Min	Max	Step
P_2(MW)	20	80	Continuous
P_5(MW)	15	50	--
P_8(MW)	10	35	--
P_{11}(MW)	10	30	--
P_{13}(MW)	12	40	--
V_G(p.u.)	0.95	1.10	--
T (p.u.)	0.90	1.10	0.01
Q_C (p.u.)	0.00	0.05	0.001

TABLE III
GENERATOR COST COEFFICIENTS OF IEEE 30-BUS SYSTEM

Bus no.	Cost coefficients		
	a	b	c
1	0.00	2.00	0.00375
2	0.00	1.75	0.01750
5	0.00	1.00	0.06250
8	0.00	3.25	0.00834
11	0.00	3.00	0.02500
13	0.00	3.00	0.02500

TABLE IV
OPTIMAL SOLUTIONS FOR CASE 1 OF IEEE 30-BUS SYSTEM

Control variables	MOQOCS	MOCS	MOPSO
P_1(MW)	141.8487	155.1729	121.3579
P_2(MW)	54.5133	52.7212	62.1462
P_5(MW)	33.1509	37.3662	35.6263
P_8(MW)	35.0000	34.2326	34.8620
P_{11}(MW)	26.4683	26.9567	23.0183
P_{13}(MW)	23.8736	26.0591	27.4026
V_1(p.u.)	1.0994	1.0561	1.0646
V_2(p.u.)	1.0907	1.0432	1.0531
V_5(p.u.)	1.0673	1.0177	1.0255
V_8(p.u.)	1.0792	1.0275	1.0330
V_{11}(p.u.)	1.0995	1.0901	1.0754
V_{13}(p.u.)	1.0949	1.0977	1.1000
T_{11}(p.u.)	1.0700	0.9400	1.0000
T_{12}(p.u.)	0.9000	0.9400	0.9300
T_{15}(p.u.)	0.9800	0.9600	0.9900
T_{36}(p.u.)	0.9700	0.9400	0.9500
Q_{C10}(p.u.)	0.0240	0.0100	0.0170
Q_{C12}(p.u.)	0.0180	0.0190	0.0120
Q_{C15}(p.u.)	0.0150	0.0080	0.0250
Q_{C17}(p.u.)	0.0230	0.0240	0.0090
Q_{C20}(p.u.)	0.0230	0.0210	0.0140
Q_{C21}(p.u.)	0.0240	0.0230	0.0120
Q_{C23}(p.u.)	0.0200	0.0160	0.0160
Q_{C24}(p.u.)	0.0240	0.0200	0.0170
Q_{C29}(p.u.)	0.0090	0.0090	0.0170
Fuel cost ($/h)	836.4424	849.013601	850.2705
Loss (MW)	4.9040	4.95520324	4.9820
VD. (p.u.)	1.8829	1.395367908	1.1957

TABLE V
OPTIMAL SOLUTIONS FOR CASE 2 OF IEEE 30-BUS SYSTEM

Control variables	MOQOCS	MOCS	MOPSO
P_1(MW)	141.7724	123.9168	126.4749
P_2(MW)	48.8748	49.6020	49.1051
P_5(MW)	21.3345	21.7950	21.5987
P_8(MW)	20.4854	21.1487	21.1370
P_{11}(MW)	11.8675	12.1259	11.3384
P_{13}(MW)	12.1742	12.0000	12.0325
V_1(p.u.)	1.0998	1.1000	1.1000
V_2(p.u.)	1.0802	1.0803	1.0843
V_5(p.u.)	1.0496	1.0453	1.0591
V_8(p.u.)	1.0516	1.0507	1.0564
V_{11}(p.u.)	1.0328	1.0598	1.0692
V_{13}(p.u.)	1.0342	1.0344	1.0521
T_{11}(p.u.)	1.0900	1.0500	1.0400
T_{12}(p.u.)	0.9600	1.0500	1.0600
T_{15}(p.u.)	1.0800	1.0700	1.0800
T_{36}(p.u.)	1.0100	1.0100	0.9900
Q_{C10}(p.u.)	0.0040	0.0230	0.0110
Q_{C12}(p.u.)	0.0200	0.0100	0.0150
Q_{C15}(p.u.)	0.0200	0.0160	0.0090
Q_{C17}(p.u.)	0.0150	0.0170	0.0150
Q_{C20}(p.u.)	0.0240	0.0210	0.0150
Q_{C21}(p.u.)	0.0240	0.0130	0.0230
Q_{C23}(p.u.)	0.0210	0.0220	0.0150
Q_{C24}(p.u.)	0.0250	0.0250	0.0160
Q_{C29}(p.u.)	0.0110	0.0070	0.0010
Fuel cost ($/h)	799.9640	800.0395	800.2388
Loss (MW)	8.9177	8.7951	8.9277
VD. (p.u.)	0.3776	0.3991	0.4063

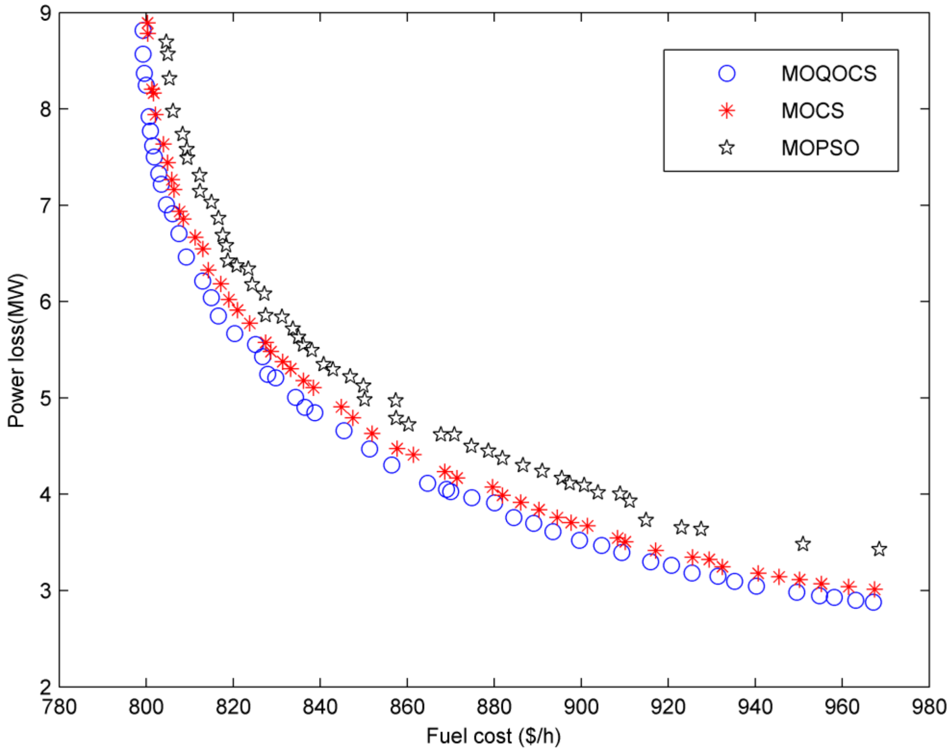

Fig. 2. Pareto optimal fronts of MOQOCS, MOCS and MOPSO for Case 1 of the IEEE 30-bus system.

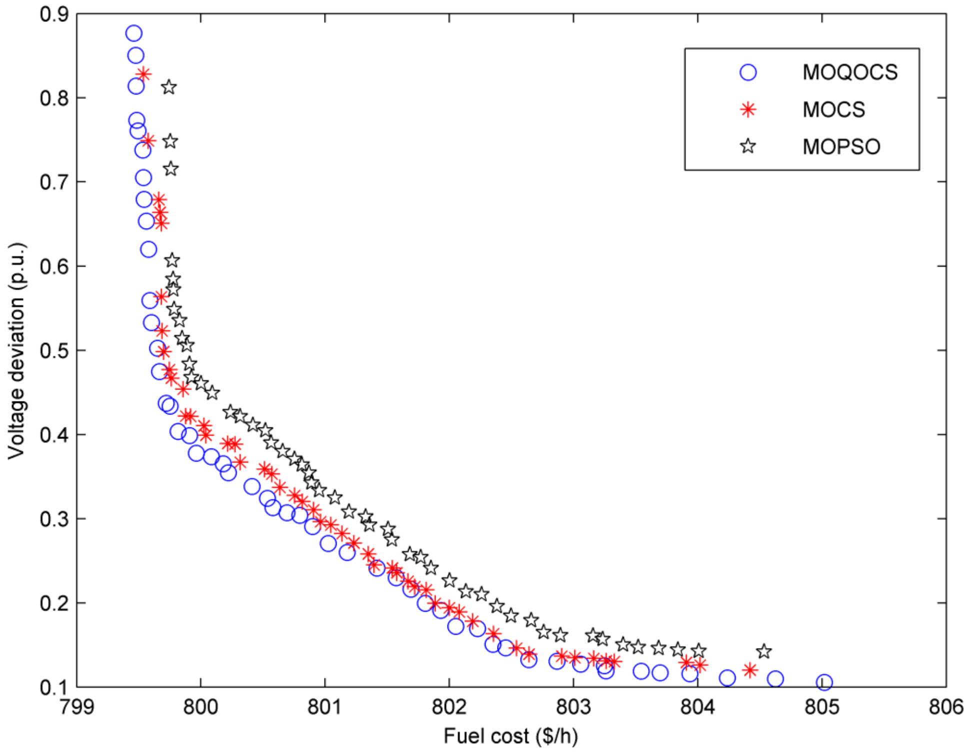

Fig. 3. Pareto optimal fronts of MOQOCS, MOCS and MOPSO for Case 2 of the IEEE 30-bus system.

V. Simulation Results

To validate the effectiveness of MOQOCS approach for solving MOOPF problem, we applied MOCS, MOQOCS approaches to the IEEE 30-bus and IEEE 57-bus system. The population size is set to 50 and the maximum cycle numbers are set to 300 and 500, respectively. The multi-objective PSO is performed in the same simulation environment and compared with the proposed methods. All the optimization

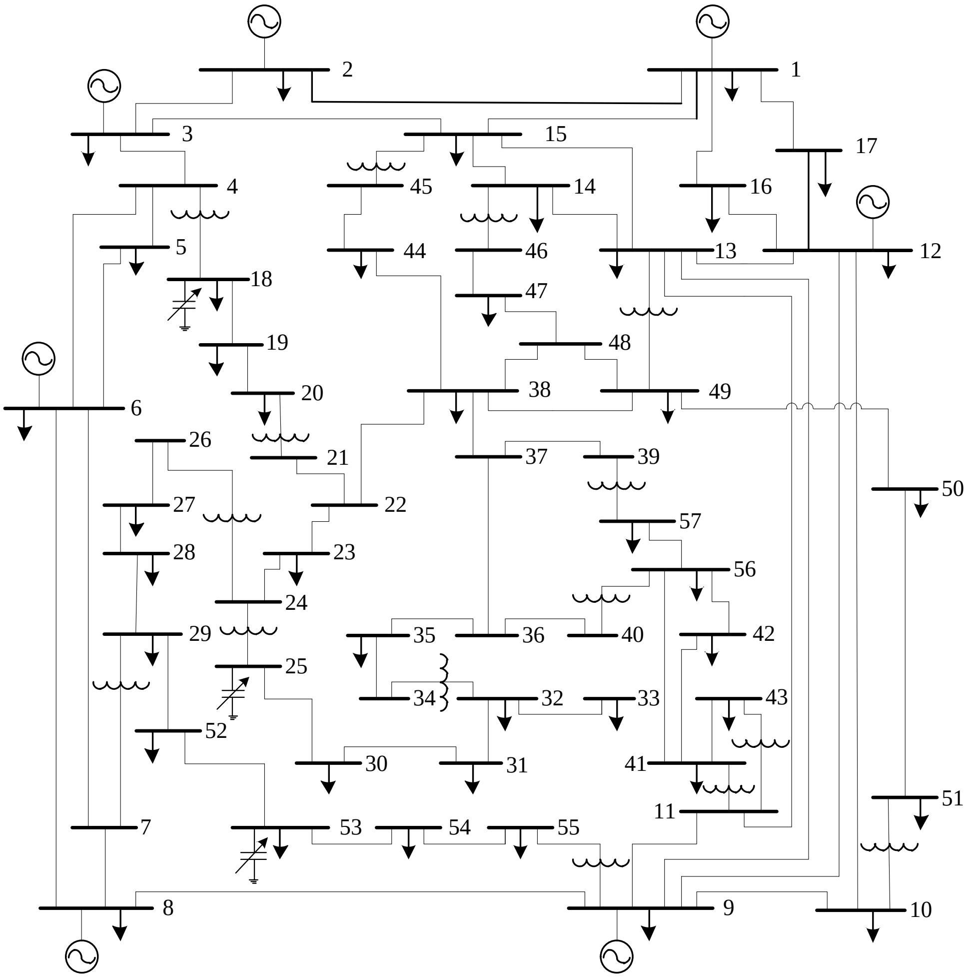

Fig. 4. The system structure diagram of IEEE 57-bus system.

programs are coded in MATLAB 2014a programming language and run on a 2.53 GHz personal computer with 4 GB RAM.

A. IEEE 30-bus System

The main characteristics of IEEE 30-bus system have been shown in Table I and its detailed data is obtained from [35]. The specific system structure diagram is presented in Fig. 1, from which we can see that the 30-bus system has 6 generators and 4 transformers. The total power demands of the test system are ($2.834+j1.262$) in p.u, respectively, at 100 MVA base [3]. This test system has 24 control variables which consist of the active power of PV buses, voltages magnitudes of generator buses, transformer ratio and shunt reactive power compensating. Table II shows the limits of system control variables and the step size of discrete variables. In addition, the fuel cost coefficients of generators of this system are listed in Table III.

TABLE VI
MAIN CHARACTERISTICS OF IEEE 57-BUS SYSTEM

Characteristics	IEEE 57	
	Number	Details
Buses	57	-
Branches	80	-
Generators	7	Buses: 1, 2, 3, 6, 8, 9 and 12
Capacitor banks	3	Buses: 18, 25 and 53
Transformers	17	Branches: 19, 20, 31, 35, 36, 37, 41, 46, 54, 58, 59, 65, 66, 71, 73, 76 and 80
Control variables	33	-

TABLE VII
CONTROL VARIABLES SETTINGS OF IEEE 57-BUS SYSTEM

Control variables	Min	Max	Step
P_2(MW)	0	100	Continuous
P_3(MW)	0	140	--
P_6(MW)	0	100	--
P_8(MW)	0	550	--
P_9(MW)	0	100	--
P_{12}(MW)	0	410	--
V_G(p.u.)	0.95	1.10	--
T (p.u.)	0.90	1.10	0.01
Q_C (p.u.)	0.00	0.30	0.01

TABLE VIII
GENERATOR COST COEFFICIENTS OF IEEE 57-BUS SYSTEM

Bus no.	Cost coefficients		
	a	b	c
1	0.00	20	0.0775795
2	0.00	40	0.01
3	0.00	20	0.25
6	0.00	40	0.01
8	0.00	20	0.0222222
9	0.00	40	0.01
12	0.00	20	0.0322581

TABLE IX
OPTIMAL SOLUTIONS FOR CASE 1 OF IEEE 57-BUS SYSTEM

Control variables	MOQOCS	MOCS	MOPSO
P_1(MW)	162.8950	158.3636	163.8368
P_2(MW)	63.0847	75.9674	81.8002
P_3(MW)	64.1195	61.9222	65.8043
P_6(MW)	96.3255	98.7284	87.1061
P_8(MW)	366.3409	357.9396	356.1229
P_9(MW)	99.3859	99.8737	99.3693
P_{12}(MW)	410.0000	410.0000	409.4457
V_1(p.u.)	1.0463	1.0251	1.0390
V_2(p.u.)	1.0441	1.0217	1.0333
V_3(p.u.)	1.0452	1.0165	1.0186
V_6(p.u.)	1.0542	1.0142	1.0005
V_8(p.u.)	1.0587	1.0072	0.9890
V_9(p.u.)	1.0353	0.9889	0.9803
V_{12}(p.u.)	1.0426	0.9961	0.9992
$T_{4\text{-}18}$(p.u.)	1.0800	1.0100	1.0000
$T_{4\text{-}18}$(p.u.)	0.9200	1.0800	0.9600
$T_{21\text{-}20}$(p.u.)	1.0200	1.0800	1.0200
$T_{24\text{-}25}$(p.u.)	1.1000	1.0400	1.0900
$T_{24\text{-}25}$(p.u.)	0.9900	0.9800	1.1000
$T_{24\text{-}26}$(p.u.)	1.0200	1.0300	1.0000
$T_{7\text{-}29}$(p.u.)	1.0100	0.9500	0.9400
$T_{34\text{-}32}$(p.u.)	0.9600	0.9700	0.9800
$T_{11\text{-}41}$(p.u.)	0.9100	0.9000	0.9000
$T_{15\text{-}45}$ (p.u.)	0.9700	0.9400	0.9500
$T_{14\text{-}46}$(p.u.)	0.9700	0.9200	0.9400
$T_{10\text{-}51}$(p.u.)	0.9700	0.9400	0.9300
$T_{13\text{-}49}$(p.u.)	0.9300	0.9000	0.9100
$T_{11\text{-}43}$(p.u.)	0.9900	0.9200	0.9200
$T_{40\text{-}56}$(p.u.)	0.9900	1.0000	1.0100
$T_{39\text{-}57}$(p.u.)	0.9900	0.9800	0.9600
$T_{9\text{-}55}$(p.u.)	0.9700	0.9600	0.9600
Q_{C18}(p.u.)	0.0700	0.1700	0.1200
Q_{C25}(p.u.)	0.1800	0.1200	0.2700
Q_{C53}(p.u.)	0.1000	0.1600	0.1400
Fuel cost ($/h)	42141.6241	42176.1426	42267.9647
Loss (MW)	11.3515	11.9950	12.6854
VD. (p.u.)	1.2530	1.0976	1.0212

TABLE X
OPTIMAL SOLUTIONS FOR CASE 2 OF IEEE 57-BUS SYSTEM

Control variables	MOQOCS	MOCS	MOPSO
P_1(MW)	143.7306	142.6769	143.2937
P_2(MW)	92.0984	96.1313	93.1893
P_3(MW)	45.8069	44.7229	44.2879
P_6(MW)	67.9360	69.2148	68.4100
P_8(MW)	461.8063	456.8351	463.5491
P_9(MW)	95.6923	94.2731	90.4651
P_{12}(MW)	360.1945	363.3278	364.6325
V_1(p.u.)	1.0166	1.0528	1.0000
V_2(p.u.)	1.0168	1.0476	0.9971
V_3(p.u.)	1.0177	1.0324	0.9881
V_6(p.u.)	1.0392	1.0243	1.0029
V_8(p.u.)	1.0594	1.0269	1.0072
V_9(p.u.)	1.0257	1.0078	0.9810
V_{12}(p.u.)	1.0204	1.0135	0.9901
$T_{4\text{-}18}$(p.u.)	1.0300	1.0000	0.9900
$T_{4\text{-}18}$(p.u.)	1.0200	0.9900	1.0500
$T_{21\text{-}20}$(p.u.)	0.9800	0.9900	0.9800
$T_{24\text{-}25}$(p.u.)	0.9600	1.0000	1.0600
$T_{24\text{-}25}$(p.u.)	1.1000	1.1000	1.0600
$T_{24\text{-}26}$(p.u.)	1.0000	1.0300	1.0100
$T_{7\text{-}29}$(p.u.)	1.0300	0.9900	0.9800
$T_{34\text{-}32}$(p.u.)	0.9200	0.9300	0.9300
$T_{11\text{-}41}$(p.u.)	0.9000	0.9000	0.9000
$T_{15\text{-}45}$ (p.u.)	0.9400	0.9700	0.9200
$T_{14\text{-}46}$(p.u.)	0.9600	0.9800	0.9200
$T_{10\text{-}51}$(p.u.)	1.0100	0.9800	0.9600
$T_{13\text{-}49}$(p.u.)	0.9100	0.9200	0.9000
$T_{11\text{-}43}$(p.u.)	0.9700	0.9400	0.9000
$T_{40\text{-}56}$(p.u.)	1.0100	1.0200	1.0500
$T_{39\text{-}57}$(p.u.)	0.9100	0.9300	0.9200
$T_{9\text{-}55}$(p.u.)	1.0200	1.0000	0.9700
Q_{C18}(p.u.)	0.1500	0.0700	0.2500
Q_{C25}(p.u.)	0.1500	0.1800	0.1900
Q_{C53}(p.u.)	0.2400	0.2200	0.2000
Fuel cost ($/h)	41734.0015	41740.5413	41760.6149
Loss (MW)	16.4648	16.3818	17.0276
VD. (p.u.)	0.6766	0.7251	0.7808

1) Case 1: Minimizing Fuel Cost and Power Losses

The multi-objective function F_1 is applied as the optimal objective in this case, which optimizes the fuel cost and power losses simultaneously. The obtained results by the MOQOCS algorithm and other algorithms are given in Table IV. It can be seen that the best compromise solution is obtained by the proposed MOQOCS method, with the minimum fuel cost of 836.4424 $/h and minimum power loss of 4.9040 MW. Moreover, the Pareto optimal fronts of MOQOCS, MOCS and MOPSO approaches are shown in Fig. 2. This figure presents that the most of Pareto optimal solutions obtained by MOQOCS are better than other two methods, which has the better Pareto-optimal front. These simulation results clearly prove the superiority of MOQOCS method.

2) Case 2: Minimizing Fuel Cost and Voltage Deviation

In this case, another multi-objective function F_2 is considered as the optimal objective, which optimizes fuel cost and voltage deviation at the same time. The optimal control variables and objective values of MOCS, MOQOCS and MOPSO algorithms are shown in Table V. As seen is this table, the MOQOCS approach obtain the best compromise solution between the three approaches, with the optimal fuel cost of 799.9640 $/h and optimal voltage deviation of 0.3776 p.u. To clearly compare the performance of these methods, the Pareto optimal fronts of MOQOCS, MOCS and MOPSO methods are given in Fig. 3. It can be seen that MOQOCS obtains the better Pareto-optimal front, and most solutions obtained by MOQOCS are superior to MOCS and MOPSO approaches.

B. IEEE 57-bus System

The IEEE 57-bus power flow test system, which is a larger scale power system, has been adopted to further evaluate the performance of MOQOCS algorithm in this paper. The main characteristics of this test system have been shown in Table VI and the detailed system data can be obtained from [1]. The system structure diagram of this system is presented in Fig. 4, from which we can see that this system has 7 generators and 17 transformers. The total load demands of the 57-bus system are (12.508+j3.364) in p.u, respectively, at 100 MVA base. Table VII shows the limits of system control variables and the step size of discrete variables. In addition, the fuel cost coefficients of generators for this system are presented in Table VIII.

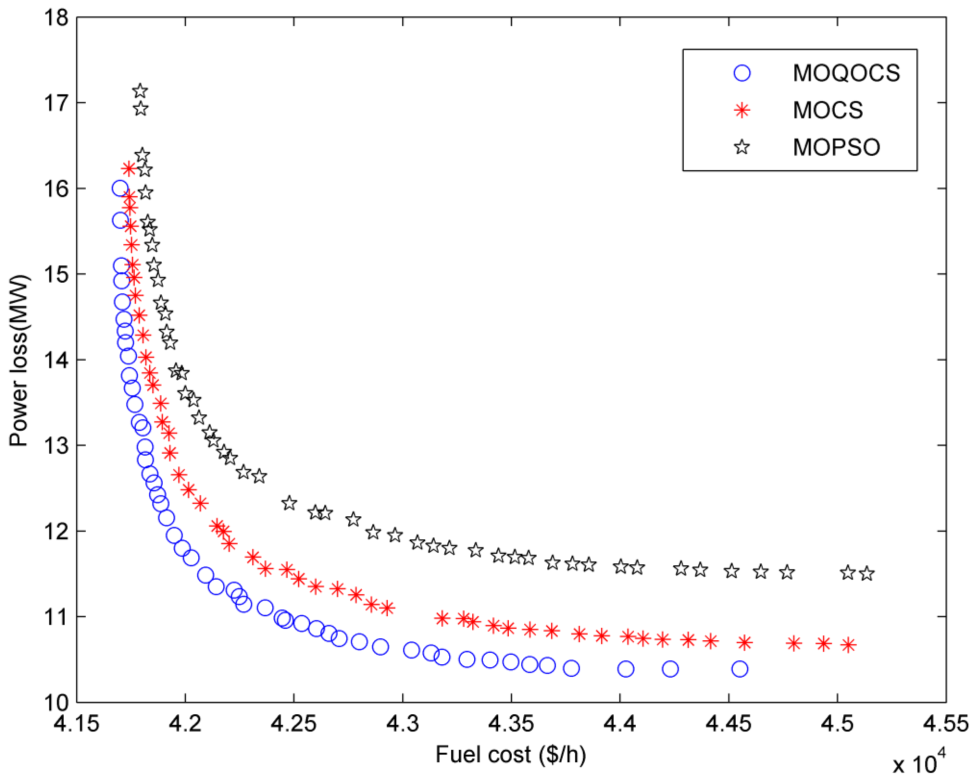

Fig. 5. Pareto optimal fronts of MOQOCS, MOCS and MOPSO for Case 1 of IEEE 57-bus system.

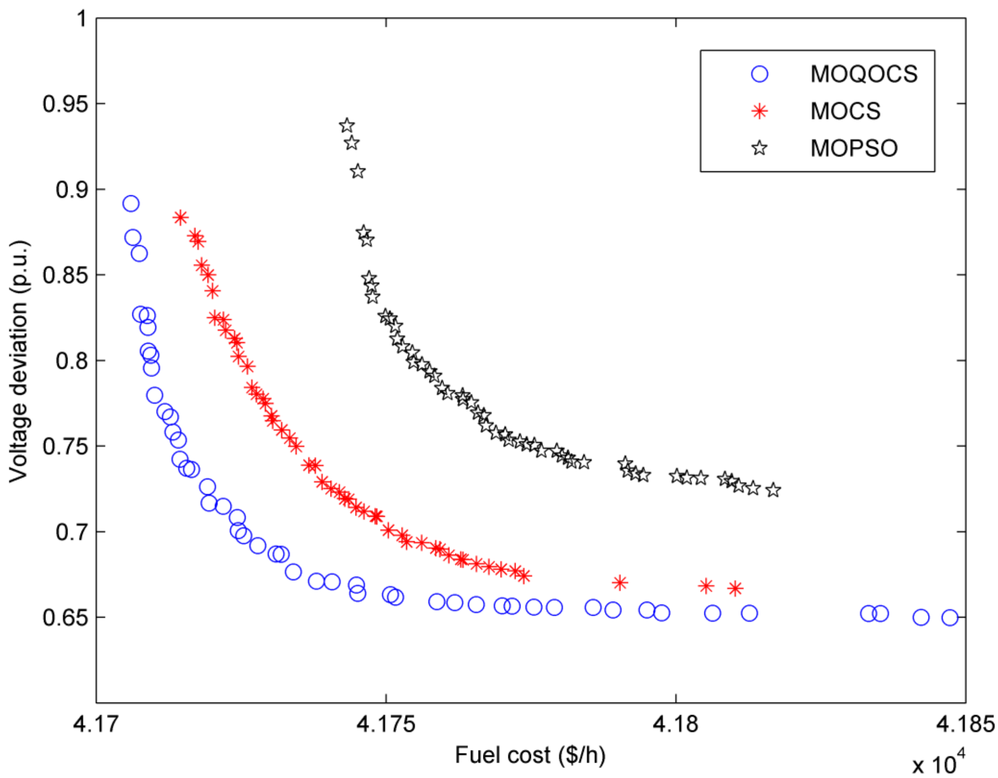

Fig. 6. Pareto optimal fronts of MOQOCS, MOCS and MOPSO for Case 2 of IEEE 57-bus system.

1) Case 1: Minimizing Fuel Cost and Power Losses

In this case, the simulation experiment is to minimize fuel cost and power losses by MOQOCS, MOCS and MOPSO methods. The obtained best compromise solutions and the optimal control variables are presented in Table IX. As seen in Table IX, the optimal fuel cost and power losses by using MOQOCS approach are 42141.6241 $/h and 11.3515 MW, which are better than those results of other two algorithms. In

TABLE XI
CPU AVERAGE TIMES (S) OF DIFFERENT METHODS ON TWO TEST SYSTEMS

	MOQOCS	MOCS	MOPSO
IEEE 30-bus system			
Case 1	83.9740	81.7112	82.1839
Case 2	75.6134	74.0871	73.8015
IEEE 57-bus system			
Case 1	267.4018	265.0341	264.1274
Case 2	255.8187	252.3415	254.0726

TABLE XII
AVERAGE RESULTS OF C-METRIC FOR DIFFERENT CASES

	C (A, B)	C (B, A)	C (A, C)	C (C, A)
IEEE 30-bus system				
Case 1	0.8476	0.0443	0.9229	0.0016
Case 2	0.9137	0.0063	0.9639	0.0000
IEEE 57-bus system				
Case 1	0.8712	0.0354	0.9317	0.0085
Case 2	0.9229	0.0031	0.9763	0.0000

addition, the Pareto optimal fronts of MOQOCS, MOCS and MOPSO approaches are shown in Fig. 5. It is obvious that MOQOCS method obtains better Pareto-optimal solutions than MOCS and MOPSO methods. From the results, it can be seen that MOQOCS method is also applicable to the large-scale system and has greater advantages compared with other algorithms.

2) Case 2: Minimizing Fuel Cost and Voltage Deviation

Another multi-objective function is considered for the IEEE-57 test system in this case, which optimizes the fuel cost and voltage deviation at the same time. The obtained best compromise solutions and the optimal control variables of MOCS, MOQOCS and MOPSO algorithms are summarized in Table X. It can be observed from table X that MOQOCS approach obtain the better compromise solution among the three approaches, with the optimal fuel cost of 41734.0015 \$/h and optimal voltage deviation of 0.6766 p.u. Fig. 6 presents the Pareto optimal fronts of this optimization problem for 57-bus system. It is obvious that MOQOCS obtains better distributed solution which is closer to the true Pareto front than MOCS and MOPSO methods.

C. Performance Measure

1) Computation Efficiency

In order to measure the computational efficiency, CPU average times of three different algorithms by 30 independent runs are summarized in Table XI. From this table, it is obvious that those computational times are almost synchronized for the same optimization problem, and the proposed methods can obtain superior solution within the suitable time for MOOPF problem.

2) Quality Indicator based on C-metric

The superiority of the MOQOCS method has been demonstrated according to the best compromise solutions and Pareto optimal fronts in previous section. In this paper, *C*-metric is adopted as the quality indicator which is the most common method to assess the quality of the obtained Pareto optimal sets. *C*-metric can compare two non-dominated sets (*A*, *B*) which derived from two different methods, which can be defined as [36]:

$$C(A,B)=\frac{\left|\{b\in B,\exists a\in A: a\prec b\}\right|}{|B|} \tag{37}$$

where *C* (*A*, *B*) is the percentage of individuals in set *B* dominated by any individual in another set *A*. If *C* (*A*, *B*)=1, all solutions in set *B* are dominated by solutions in set *A* [37]. If *C* (*A*, *B*) = 0, no solution in set *B* is covered by set *A*. Both *C* (*A*, *B*) and *C* (*B*, *A*) should be considered since the non-symmetry of *C*-metric and $C(A, B) + C(B, A) \neq 1$.

The calculated results of *C*-metric for all cases are illustrated in Table XII, where A, B, C indicate MOQOCS, MOCS and MOPSO methods. Table XII shows that for IEEE 30-bus system, the solutions of MOQOCS can dominate 84.76% and 92.29% solutions obtained by QOCS and QOPSO for Case 1. MOQOCS generate optimal sets can dominate 91.37% and 96.39% solutions derived from QOCS and QOPSO for Case 2. Table XII indicates that for IEEE 57-bus system, QOCS and QOPSO have 87.12% and 93.17% solutions dominated by those solutions of MOQOCS for Case 1. Likewise, MOQOCS obtains optimal set can dominate 92.29% and 97.63% solutions produced by QOCS and QOPSO for Case 2.

VI. CONCLUSION

The MOOPF is a multi-variable, multi-constraint and nonlinear optimization problem which requires to optimizing multiple objectives simultaneously. This paper first applies the MOCS method to solve the MOOPF problem and obtain better Pareto optimal solutions than CS. Moreover, the quasi-oppositional based learning is introduced to propose the MOQOCS algorithm for speeding up convergence and enhancing the quality of solution. In MOCS and MOQOCS methods, FDP is proposed to determine the dominance relation of different solutions based on the sum of constraint violation values, which can strengthen the feasibility of simulation result. And crowding distance sorting is considered to enhance the diversity of Pareto-optimal solution and obtain better distributed Pareto optimal front. MOCS and MOQOCS methods have been validated on IEEE 30-bus and IEEE 57-bus test systems for solving MOOPF problem. The results prove that MOQOCS method is more efficient to find the Pareto-optimal solutions and best compromise solution than MOCS and MOPSO methods. The curves of Pareto optimal fronts show that MOQOCS method generates better distributed solutions compared with other two algorithms. Therefore, it can be concluded that the proposed MOQOCS method is an efficient and reliable algorithm for MOOPF problem.

REFERENCES

[1] M. Ghasemi, S. Ghavidel, M. M. Ghanbarian, M. Gharibzadeh and A. A. Vahed. "Multi-objective optimal power flow considering the cost, emission, voltage deviation and power losses using multi-objective modified imperialist competitive algorithm," *Energy*, vol. 78, pp. 276-289, 2014.

[2] R. P. Singh, V. Mukherjee and S. P. Ghoshal. "Particle swarm optimization with an aging leader and challengers algorithm for the solution of optimal power flow problem," *Applied Soft Computing*, vol. 40, pp. 161-177, 2016.

[3] R. Roy and H. T. Jadhav. "Optimal power flow solution of power system incorporating stochastic wind power using Gbest guided artificial bee colony algorithm," *International Journal of Electrical Power & Energy Systems*, vol. 64, pp. 562-578, 2015.

[4] J. Zhang, Q. Tang, P. Li, D. Deng and Y. Chen. "A modified MOEA/D approach to the solution of multi-objective optimal power flow problem," *Applied Soft Computing*, vol. 47, pp. 494-514, 2016.

[5] X. Yuan, B. Zhang, P. Wang, J. Liang, Y. Yuan, Y. Huang and X. Lei. "Multi-objective optimal power flow based on improved strength Pareto evolutionary algorithm," *Energy*, vol. 122, pp. 70-82, 2017.

[6] K. Abaci and V. Yamacli. "Differential search algorithm for solving multi-objective optimal power flow problem," *International Journal of Electrical Power & Energy Systems*, vol. 79, pp. 1-10, 2016.

[7] H. Xiong, H. Cheng and H. Li. "Optimal reactive power flow incorporating static voltage stability based on multi-objective adaptive immune algorithm," *Energy Conversion and Management*, vol. 49, no. 5, pp. 1175-1181, 2008.

[8] A. E. Chaib, H. R. E. H. Bouchekara, R. Mehasni and M. A. Abido. "Optimal power flow with emission and non-smooth cost functions using backtracking search optimization algorithm," *International Journal of Electrical Power & Energy Systems*, vol. 81, pp. 64-77, 2016.

[9] 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.

[10] S. Sivasubramani and K. S. Swarup. "Multi-objective harmony search algorithm for optimal power flow problem," *International Journal of Electrical Power & Energy Systems*, vol. 33, no. 3, pp. 745-752, 2011.

[11] M. Ghasemi, S. Ghavidel, M. M. Ghanbarian, H. R. Massrur and M. Gharibzadeh. "Application of imperialist competitive algorithm with its modified techniques for multi-objective optimal power flow problem: A comparative study," *Information Sciences*, vol. 281, pp. 225-247, 2014.

[12] X. Yang and S. Deb. "Cuckoo search via Lévy flights,". in *World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), IEEE Publications, USA*, 2009. pp. 210-214

[13] A. Layeb and S. R. Boussalia. "A novel quantum inspired cuckoo search algorithm for bin packing problem," *International Journal of Information Technology and Computer Science*, vol. 4, no. 5, pp. 58-67, 2012.

[14] G. Sun, Y. Liu, J. Li, Y. Zhang and A. A. Wang. "Sidelobe reduction of large-scale antenna array for 5G beamforming via hierarchical cuckoo search," *Electronics Letters*, vol. 53, no. 16, pp. 1158-1160, 2017.

[15] S. Araghi, A. Khosravi and D. Creighton. "Intelligent cuckoo search optimized traffic signal controllers for multi-intersection network," *Expert Systems with Applications*, vol. 42, no. 9, pp. 4422-4431, 2015.

[16] S. A. Medjahed, T. A. Saadi, A. Benyettou and M. Ouali. "Binary cuckoo search algorithm for band selection in hyperspectral image classification," *IAENG International Journal of Computer Science*, vol. 42, no. 3, pp. 183-191, 2015.

[17] Y. Zhou, H. Zhen, Q. Luo and J. Wu. "An improved cuckoo search algorithm for solving planar graph coloring problem," *Applied Mathematics & Information Sciences*, vol. 7, no. 2, pp. 785-792, 2013.

[18] S. Suresh and S. Lal. "An efficient cuckoo search algorithm based multilevel thresholding for segmentation of satellite images using different objective functions," *Expert Systems with Applications*, vol. 58, no. C, pp. 184-209, 2016.

[19] X. Yang and S. Deb. "Multiobjective cuckoo search for design optimization," *Computers & Operations Research*, vol. 40, no. 6, pp. 1616-1624, 2013.

[20] A. Zamani, S. Tavakoli and S. Etedali. "Fractional order PID control design for semi-active control of smart base-isolated structures: A multi-objective cuckoo search approach," *ISA Transactions*, vol. 67, pp. 222-232, 2017.

[21] Z. Wang and Y. Li. "Irreversibility analysis for optimization design of plate fin heat exchangers using a multi-objective cuckoo search algorithm," *Energy Conversion and Management*, vol. 101, pp. 126-135, 2015.

[22] A. Ramesh Kumar and L. Premalatha. "Optimal power flow for a deregulated power system using adaptive real coded biogeography-based optimization," *International Journal of Electrical Power & Energy Systems*, vol. 73, pp. 393-399, 2015.

[23] R. Shang, Y. Wang, J. Wang, L. Jiao, S. Wang and L. Qi. "A multi-population cooperative coevolutionary algorithm for multi-objective capacitated arc routing problem," *Information Sciences*, vol. 277, pp. 609-642, 2014.

[24] I. Alberto, C. A. Coello Coello and P. M. Mateo. "A comparative study of variation operators used for evolutionary multi-objective optimization," *Information Sciences*, vol. 273, pp. 33-48, 2014.

[25] M. Ghasemi, S. Ghavidel, M. M. Ghanbarian and M. Gitizadeh. "Multi-objective optimal electric power planning in the power system using Gaussian bare-bones imperialist competitive algorithm," *Information Sciences*, vol. 294, pp. 286-304, 2015.

[26] A. M. Shaheen, R. A. El-Sehiemy and S. M. Farrag. "Solving multi-objective optimal power flow problem via forced initialised differential evolution algorithm," *IET Generation, Transmission & Distribution*, vol. 10, no. 7, pp. 1634-1647, 2016.

[27] A. A. Mohamed, Y. S. Mohamed, A. A. M. El-Gaafary and A. M. Hemeida. "Optimal power flow using moth swarm algorithm," *Electric Power Systems Research*, vol. 142, pp. 190-206, 2017.

[28] T. T. Nguyen and D. N. Vo. "Modified cuckoo search algorithm for short-term hydrothermal scheduling," *International Journal of Electrical Power & Energy Systems*, vol. 65, pp. 271-281, 2015.

[29] S. Li and J. Wang. "Improved cuckoo search algorithm with novel searching mechanism for solving unconstrained function optimization problem," *IAENG International Journal of Computer Science*, vol. 44, no. 1, pp. 8-12, 2017.

[30] L. Wang, Y. Zhong and Y. Yin. "Nearest neighbour cuckoo search algorithm with probabilistic mutation," *Applied Soft Computing*, vol. 49, pp. 498-509, 2016.

[31] J. Wang and S. Li. "PID decoupling controller design for electroslag remelting process using cuckoo search algorithm with self-tuning dynamic searching mechanism," *Engineering Letters*, vol. 25, no. 2, pp. 125-133, 2017.

[32] H. Tizhoosh. "Opposition-Based Learning: ANew Scheme for Machine Intelligence,". in *International Conference on International Conference on Computational Intelligence for Modelling, Control and Automation*, 2005. pp. 695-701

[33] B. Mandal and P. K. Roy. "Optimal reactive power dispatch using quasi-oppositional teaching learning based optimization," *International Journal of Electrical Power & Energy Systems*, vol. 53, pp. 123-134, 2013.

[34] M. Basu. "Quasi-oppositional group search optimization for hydrothermal power system," *International Journal of Electrical Power & Energy Systems*, vol. 81, pp. 324-335, 2016.

[35] S. Duman, U. U. G Ven, Y. S Nmez and N. Y R Keren. "Optimal power flow using gravitational search algorithm," *Energy Conversion and Management*, vol. 59, pp. 86-95, 2012.

[36] 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.

[37] M. A. Medina, S. Das, C. A. Coello Coello and J. M. Ram Rez. "Decomposition-based modern metaheuristic algorithms for multi-objective optimal power flow-A comparative study," *Engineering Applications of Artificial Intelligence*, vol. 32, pp. 10-20, 2014.

[38] R. G. Babukartik. "Hybrid algorithm using the advantage of ACO and cuckoo search for job scheduling," *International Journal of Information Technology Convergence and Services*, vol. 2, no. 4, pp. 25-34, 2012.