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
presented and reported to outperform many well-known 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 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.

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 MOMSPSO 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-oppositional-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]:

\[ \text{OF} = \min f(x, u) \]  
\text{Subject to: } g(x, u) = 0 \]  
\[ h(x, u) \leq 0 \]  

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:

\[ OF = \min \left[ f_1(x, u), f_2(x, u), \ldots, f_M(x, u) \right] \]  

where \( f_i(x, u) \) indicates the \( i \)-th 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 \( U_l \) dominates solution \( U_j \) (denoted by \( U_l < U_j \)) only if both of the below conditions are satisfied [1]:

\[ f_i(U_l) \leq f_i(U_j), \quad \forall i \in \{1, 2, \ldots, M\} \]
\[ f_j(U_l) < f_j(U_j), \quad \exists j \in \{1, 2, \ldots, M\} \]  

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:

\[ f_{\text{cost}} = \sum_{i=1}^{N_G} \left( a_i + b_i P_{Gi} + c_i P_{Gi}^2 \right) \]  

where \( P_{Gi} \) indicates the real power output of the \( i \)-th generator; \( a_i, b_i, \) and \( c_i \) indicate the cost coefficients of the \( i \)-th 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]:

\[ f_{\text{loss}} = \sum_{k=1}^{N_L} g_k \left( |V_i|^2 + |V_j|^2 - 2|V_i||V_j|\cos\delta_{ij} \right) \]  

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 \( i \)-th and \( j \)-th bus and \( i \neq j \); \( \delta_{ij} \) indicates the phase difference among bus \( i \) and bus \( j \); \( N_L \) 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 \( f_{\text{voltage}} \) 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]:

\[ f_{V_D} = \sum_{i=1}^{N_{PQ}} |V_i - 1.0| \]  

where \( N_{PQ} \) indicates the number of PQ buses.

B. Multi-objective Optimization

1) Minimization of \( f_{\text{cost}} \) and \( f_{\text{loss}} \)

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:

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

2) Minimization of \( f_{\text{cost}} \) and \( f_{V_D} \)

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:

\[ F_2 = \min \left[ f_{\text{cost}}(x,u), f_{V_D}(x,u) \right] \]  

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]:

\[ P_{\text{gen}} - P_{\text{lo}} - \sum_{j=1}^{N_B} V_j \left( G_{ji} \cos \delta_j + B_{ji} \sin \delta_j \right) = 0, \quad i \in N_B \]  

\[ Q_{\text{gen}} - Q_{\text{lo}} - \sum_{j=1}^{N_B} V_j \left( G_{ji} \sin \delta_j - B_{ji} \cos \delta_j \right) = 0, \quad i \in N_B \]  

where \( N_B \) represents the amount of all system buses; \( P_{\text{gen}} \) and \( P_{\text{lo}} \) indicate the injected active power and active load demand on bus \( i \); \( Q_{\text{gen}} \) and \( Q_{\text{lo}} \) indicate the injected reactive power and reactive load demand on bus \( i \); \( G_{ji} \) and \( B_{ji} \) respectively indicate the real part and imaginary part of the \( ij \)th element of the node admittance matrix; \( \delta_j \) indicates the voltage phase difference between the \( i \)th and \( j \)th 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:

\[ V_{\text{min}} \leq V_{\text{gen}} \leq V_{\text{max}}, \quad i = 1, 2, ..., N_G \]  

\[ P_{\text{gen}}^{\min} \leq P_{\text{gen}} \leq P_{\text{gen}}^{\max}, \quad i = 1, 2, ..., N_G \]  

\[ Q_{\text{gen}}^{\min} \leq Q_{\text{gen}} \leq Q_{\text{gen}}^{\max}, \quad i = 1, 2, ..., N_G \]  

where \( N_G \) represents the number of generator buses.

ii. Transformer taps constraints:

\[ T_{i}^{\text{min}} \leq T_i \leq T_{i}^{\text{max}}, \quad i = 1, 2, ..., N_T \]  

where \( N_T \) represents the number of transformer branches.

iii. Shunt VAR compensator constraints:

\[ Q_{\text{C}}^{\min} \leq Q_{\text{C}} \leq Q_{\text{C}}^{\max}, \quad i = 1, 2, ..., N_C \]  

where \( N_C \) represents the number of reactive compensators.

iv. Security constraints:

\[ V_{\text{Li}}^{\min} \leq V_{\text{Li}} \leq V_{\text{Li}}^{\max}, \quad i = 1, 2, ..., N_{PQ} \]  

\[ S_{iL} \leq S_{iL}^{\max}, \quad i = 1, 2, ..., N_{TL} \]  

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 \( i \)th solution (for \( i = 1, 2, ..., N_P \)) at \( k \)th 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 \( i \)th cuckoo at \( (k+1) \)th iteration, the Lévy flight is performed as follows:

\[ U_i(k+1) = U_i(k) + \alpha \odot \text{Lévy} \]  

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

\[ \text{Lévy} \big( \lambda \big) = -u = \lambda^{1/\lambda}, \quad \left( 1 < \lambda \leq 3 \right) \]  

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 \( \text{Lévy} \) can be specifically calculated as follow [30, 31]:

\[ \text{Lévy} (\lambda) = \frac{\mu}{|v|^\lambda} \]  

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

\[ V_i(k) = U_i(k) + \alpha_0 \odot \frac{\mu}{|v|^\lambda} \big( U_i(k) - U_{\text{best}} \big) \]  

where \( \alpha_0 \) is the step size scaling factor; \( U_{\text{best}} \) 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:

\[
U_i(k+1) = \begin{cases} 
V_i(k) + \text{rand} \cdot (U_{i1} - U_{i2}), & \text{rand} < p_a \\
V_i(k), & \text{otherwise}
\end{cases}
\]  

(25)

where \( U_{i1} \) and \( U_{i2} \) 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:

\[
x^c = a + b - x
\]  

(26)

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

\[
x^c_i = a_i + b_i - x_i; \quad i = 1, 2, \ldots, d
\]  

(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:

\[
x^{qo} = \text{rand} \cdot \left( \frac{a + b}{2} \right) \cdot (a + b - x)
\]  

(28)

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

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

(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:

\[
U = \begin{bmatrix}
u_1^1 & u_1^2 & \cdots & u_1^D \\
u_2^1 & u_2^2 & \cdots & u_2^D \\
\vdots & \vdots & \ddots & \vdots \\
u_N^1 & u_N^2 & \cdots & u_N^D
\end{bmatrix}
\]  

(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 crowding 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 \( i \)th 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 \( i \)th solution can be expressed as:

\[
dis(i) = \frac{\sum_{j=1}^{M} f_j(i+1) - f_j(i-1)}{f_{j}^{\text{max}} - f_{j}^{\text{min}}}
\]  

(31)

where \( M \) indicates the number of optimal objective functions; \( f_j(i) \) indicates the \( j \)th optimal objective of the \( i \)th solution; \( f_{j}^{\text{max}} \) and \( f_{j}^{\text{min}} \) indicate the largest and smallest values of the \( j \)th 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 $i$th solution is superior to the $j$th solution if one of the following two formulas is satisfies.

\[
\text{rank}(i) < \text{rank}(j) \quad (32)
\]

\[
\text{rank}(i) = \text{rank}(j) \quad \text{and} \quad \text{dis}(i) > \text{dis}(j) \quad (33)
\]

Finally, we can choose the best $N$ solutions from all $2N$ solutions as the new population. In addition, the $g$best 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:

\[
\text{Constr}(U_i) = \sum_{j=1}^{N_U} \text{Constr} \left[ h_j(x,u) \right] \quad (34)
\]

Where $h_j(x,u)$ represents the $j$th inequality constraint of the $U_i$; $N_U$ represents the number of inequality constraints. If the value of $\text{ConVio}(U_i)$ 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 $U_i$ is considered to dominate another solution $U_j$ when one of the following conditions can be satisfied:

(1) $U_i$ is a feasible solution but $U_j$ is not.

(2) $U_i$ and $U_j$ are not feasible solutions and $\text{ConVio}(U_i)$ is smaller than $\text{ConVio}(U_j)$.

(3) $U_i$ and $U_j$ are feasible solutions and $U_i$ dominates $U_j$ 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 Pareo 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 $k$th objective function of solution $i$ can be defined as [10]:

\[
\mu_k(i) = \frac{f_k \leq f_k^{\min}}{f_k^{\max} - f_k^{\min}}, \quad f_k^{\min} < f_k < f_k^{\max} \quad (35)
\]

where $f_k^{\min}$ and $f_k^{\max}$ represent the minimum and maximum values of the $k$th 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:

\[
\mu(i) = \frac{\sum_{k=1}^{M} \mu_k(i)}{\sum_{k=1}^{M} \mu_k(i)} \quad (36)
\]

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 $g$best 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

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>IEEE 30</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buses</td>
<td>30</td>
</tr>
<tr>
<td>Branches</td>
<td>41</td>
</tr>
<tr>
<td>Generators</td>
<td>6</td>
</tr>
<tr>
<td>Capacitor banks</td>
<td>9</td>
</tr>
<tr>
<td>Transformers</td>
<td>4</td>
</tr>
<tr>
<td>Control variables</td>
<td>24</td>
</tr>
</tbody>
</table>

### Table II

<table>
<thead>
<tr>
<th>Control variables</th>
<th>Min</th>
<th>Max</th>
<th>Step</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_i$ (MW)</td>
<td>15</td>
<td>50</td>
<td>--</td>
</tr>
<tr>
<td>$P_i$ (MW)</td>
<td>10</td>
<td>35</td>
<td>--</td>
</tr>
<tr>
<td>$P_i$ (MW)</td>
<td>10</td>
<td>30</td>
<td>--</td>
</tr>
<tr>
<td>$P_i$ (MW)</td>
<td>12</td>
<td>40</td>
<td>--</td>
</tr>
<tr>
<td>$V_i$ (p.u.)</td>
<td>0.95</td>
<td>1.10</td>
<td>0.01</td>
</tr>
<tr>
<td>$Q_i$ (p.u.)</td>
<td>0.00</td>
<td>0.05</td>
<td>0.001</td>
</tr>
</tbody>
</table>

### Table III

<table>
<thead>
<tr>
<th>Bus no.</th>
<th>$a$</th>
<th>$b$</th>
<th>$c$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00</td>
<td>2.00</td>
<td>0.00375</td>
</tr>
<tr>
<td>2</td>
<td>0.00</td>
<td>1.75</td>
<td>0.01750</td>
</tr>
<tr>
<td>5</td>
<td>0.00</td>
<td>1.00</td>
<td>0.06250</td>
</tr>
<tr>
<td>8</td>
<td>0.00</td>
<td>3.25</td>
<td>0.00834</td>
</tr>
<tr>
<td>11</td>
<td>0.00</td>
<td>3.00</td>
<td>0.02500</td>
</tr>
<tr>
<td>13</td>
<td>0.00</td>
<td>3.00</td>
<td>0.02500</td>
</tr>
</tbody>
</table>

### Table IV

<table>
<thead>
<tr>
<th>Control variables</th>
<th>MOOCQS</th>
<th>MOCS</th>
<th>MOPSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_i$ (MW)</td>
<td>141.8487</td>
<td>155.1729</td>
<td>121.3579</td>
</tr>
<tr>
<td>$Q_i$ (MW)</td>
<td>54.5133</td>
<td>52.7212</td>
<td>62.1462</td>
</tr>
<tr>
<td>$P_i$ (MW)</td>
<td>33.1509</td>
<td>37.3662</td>
<td>35.6263</td>
</tr>
<tr>
<td>$P_i$ (MW)</td>
<td>35.0000</td>
<td>34.2326</td>
<td>34.8620</td>
</tr>
<tr>
<td>$P_i$ (MW)</td>
<td>26.4683</td>
<td>26.9567</td>
<td>23.0183</td>
</tr>
<tr>
<td>$P_i$ (MW)</td>
<td>23.8736</td>
<td>26.0591</td>
<td>27.4026</td>
</tr>
<tr>
<td>$V_i$ (p.u.)</td>
<td>1.0994</td>
<td>1.0561</td>
<td>1.0646</td>
</tr>
<tr>
<td>$V_i$ (p.u.)</td>
<td>1.0907</td>
<td>1.0432</td>
<td>1.0531</td>
</tr>
<tr>
<td>$V_i$ (p.u.)</td>
<td>1.0673</td>
<td>1.0177</td>
<td>1.0255</td>
</tr>
<tr>
<td>$V_i$ (p.u.)</td>
<td>1.0792</td>
<td>1.0275</td>
<td>1.0330</td>
</tr>
<tr>
<td>$V_i$ (p.u.)</td>
<td>1.0995</td>
<td>1.0901</td>
<td>1.0754</td>
</tr>
<tr>
<td>$V_i$ (p.u.)</td>
<td>1.0949</td>
<td>1.0977</td>
<td>1.1000</td>
</tr>
<tr>
<td>$T_i$ (p.u.)</td>
<td>1.0700</td>
<td>0.9400</td>
<td>1.0000</td>
</tr>
<tr>
<td>$T_i$ (p.u.)</td>
<td>0.9000</td>
<td>0.9400</td>
<td>0.9300</td>
</tr>
<tr>
<td>$T_i$ (p.u.)</td>
<td>0.9800</td>
<td>0.9600</td>
<td>0.9900</td>
</tr>
<tr>
<td>$T_i$ (p.u.)</td>
<td>0.9700</td>
<td>0.9400</td>
<td>0.9500</td>
</tr>
<tr>
<td>$Q_{C2}$ (p.u.)</td>
<td>0.0240</td>
<td>0.0100</td>
<td>0.0170</td>
</tr>
<tr>
<td>$Q_{C3}$ (p.u.)</td>
<td>0.0180</td>
<td>0.0190</td>
<td>0.0120</td>
</tr>
<tr>
<td>$Q_{C4}$ (p.u.)</td>
<td>0.0150</td>
<td>0.0080</td>
<td>0.0250</td>
</tr>
<tr>
<td>$Q_{C5}$ (p.u.)</td>
<td>0.0230</td>
<td>0.0240</td>
<td>0.0090</td>
</tr>
<tr>
<td>$Q_{C6}$ (p.u.)</td>
<td>0.0230</td>
<td>0.0210</td>
<td>0.0140</td>
</tr>
<tr>
<td>$Q_{C7}$ (p.u.)</td>
<td>0.0240</td>
<td>0.0230</td>
<td>0.0120</td>
</tr>
<tr>
<td>$Q_{C8}$ (p.u.)</td>
<td>0.0200</td>
<td>0.0160</td>
<td>0.0160</td>
</tr>
<tr>
<td>$Q_{C9}$ (p.u.)</td>
<td>0.0240</td>
<td>0.0200</td>
<td>0.0170</td>
</tr>
<tr>
<td>$Q_{C10}$ (p.u.)</td>
<td>0.0090</td>
<td>0.0090</td>
<td>0.0170</td>
</tr>
<tr>
<td>Fuel cost ($/h)</td>
<td>836.4424</td>
<td>849.013601</td>
<td>850.2705</td>
</tr>
<tr>
<td>Loss (MW)</td>
<td>4.9040</td>
<td>4.95520324</td>
<td>4.9820</td>
</tr>
<tr>
<td>VD. (p.u.)</td>
<td>1.8829</td>
<td>1.395367908</td>
<td>1.1957</td>
</tr>
</tbody>
</table>

### Table V

<table>
<thead>
<tr>
<th>Control variables</th>
<th>MOOCQS</th>
<th>MOCS</th>
<th>MOPSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_i$ (MW)</td>
<td>141.7724</td>
<td>123.9168</td>
<td>126.4749</td>
</tr>
<tr>
<td>$Q_i$ (MW)</td>
<td>48.8748</td>
<td>49.6020</td>
<td>49.1051</td>
</tr>
<tr>
<td>$P_i$ (MW)</td>
<td>21.3345</td>
<td>21.7950</td>
<td>21.5987</td>
</tr>
<tr>
<td>$P_i$ (MW)</td>
<td>20.4854</td>
<td>21.1487</td>
<td>21.1370</td>
</tr>
<tr>
<td>$P_i$ (MW)</td>
<td>11.8675</td>
<td>12.1259</td>
<td>11.3384</td>
</tr>
<tr>
<td>$P_i$ (MW)</td>
<td>12.1742</td>
<td>12.0000</td>
<td>12.0325</td>
</tr>
<tr>
<td>$V_i$ (p.u.)</td>
<td>1.0998</td>
<td>1.1000</td>
<td>1.1000</td>
</tr>
<tr>
<td>$V_i$ (p.u.)</td>
<td>1.0802</td>
<td>1.0803</td>
<td>1.0843</td>
</tr>
<tr>
<td>$V_i$ (p.u.)</td>
<td>1.0496</td>
<td>1.0453</td>
<td>1.0591</td>
</tr>
<tr>
<td>$V_i$ (p.u.)</td>
<td>1.0516</td>
<td>1.0507</td>
<td>1.0564</td>
</tr>
<tr>
<td>$V_i$ (p.u.)</td>
<td>1.0328</td>
<td>1.0598</td>
<td>1.0692</td>
</tr>
<tr>
<td>$V_i$ (p.u.)</td>
<td>1.0342</td>
<td>1.0344</td>
<td>1.0521</td>
</tr>
<tr>
<td>$T_i$ (p.u.)</td>
<td>1.0900</td>
<td>1.0500</td>
<td>1.0400</td>
</tr>
<tr>
<td>$T_i$ (p.u.)</td>
<td>0.9600</td>
<td>1.0500</td>
<td>1.0600</td>
</tr>
<tr>
<td>$T_i$ (p.u.)</td>
<td>1.0800</td>
<td>1.0700</td>
<td>1.0800</td>
</tr>
<tr>
<td>$T_i$ (p.u.)</td>
<td>1.0100</td>
<td>1.0100</td>
<td>0.9900</td>
</tr>
<tr>
<td>$Q_{C2}$ (p.u.)</td>
<td>0.0040</td>
<td>0.0230</td>
<td>0.0110</td>
</tr>
<tr>
<td>$Q_{C3}$ (p.u.)</td>
<td>0.0200</td>
<td>0.0100</td>
<td>0.0150</td>
</tr>
<tr>
<td>$Q_{C4}$ (p.u.)</td>
<td>0.0200</td>
<td>0.0160</td>
<td>0.0090</td>
</tr>
<tr>
<td>$Q_{C5}$ (p.u.)</td>
<td>0.0150</td>
<td>0.0170</td>
<td>0.0150</td>
</tr>
<tr>
<td>$Q_{C6}$ (p.u.)</td>
<td>0.0240</td>
<td>0.0210</td>
<td>0.0150</td>
</tr>
<tr>
<td>$Q_{C7}$ (p.u.)</td>
<td>0.0240</td>
<td>0.0130</td>
<td>0.0230</td>
</tr>
<tr>
<td>$Q_{C8}$ (p.u.)</td>
<td>0.0210</td>
<td>0.0220</td>
<td>0.0150</td>
</tr>
<tr>
<td>$Q_{C9}$ (p.u.)</td>
<td>0.0250</td>
<td>0.0250</td>
<td>0.0160</td>
</tr>
<tr>
<td>$Q_{C10}$ (p.u.)</td>
<td>0.0110</td>
<td>0.0070</td>
<td>0.0010</td>
</tr>
<tr>
<td>Fuel cost ($/h)</td>
<td>799.9640</td>
<td>800.0395</td>
<td>800.2388</td>
</tr>
<tr>
<td>Loss (MW)</td>
<td>8.9177</td>
<td>8.7951</td>
<td>8.9277</td>
</tr>
<tr>
<td>VD. (p.u.)</td>
<td>0.3776</td>
<td>0.3991</td>
<td>0.4063</td>
</tr>
</tbody>
</table>
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
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>
<thead>
<tr>
<th>Control variables</th>
<th>Min</th>
<th>Max</th>
<th>Step</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_2$ (MW)</td>
<td>0</td>
<td>100</td>
<td>Continuous</td>
</tr>
<tr>
<td>$P_3$ (MW)</td>
<td>0</td>
<td>140</td>
<td>--</td>
</tr>
<tr>
<td>$P_6$ (MW)</td>
<td>0</td>
<td>100</td>
<td>--</td>
</tr>
<tr>
<td>$P_8$ (MW)</td>
<td>0</td>
<td>550</td>
<td>--</td>
</tr>
<tr>
<td>$P_9$ (MW)</td>
<td>0</td>
<td>100</td>
<td>--</td>
</tr>
<tr>
<td>$P_{12}$ (MW)</td>
<td>0</td>
<td>410</td>
<td>--</td>
</tr>
<tr>
<td>$V_G$ (p.u.)</td>
<td>0.95</td>
<td>1.10</td>
<td>0.01</td>
</tr>
<tr>
<td>$T_0$ (p.u.)</td>
<td>0.90</td>
<td>1.10</td>
<td>0.01</td>
</tr>
<tr>
<td>$Q_C$ (p.u.)</td>
<td>0.00</td>
<td>0.30</td>
<td>0.01</td>
</tr>
</tbody>
</table>

![Fig. 4. The system structure diagram of IEEE 57-bus system.](Advance online publication: 28 May 2018)
variables and objective values of MOCS, MOQOCS and MOQOCS are better than other two methods, which has the best Pareto-optimal front. These simulation results clearly presents that the most of Pareto optimal solutions obtained by MOQOCS method and other algorithms are given in Table IV. It can be seen that the multi-objective function (p.u.) from which we can see that this system has 7 generators and 17 transformers. The total load demands of the 57-bus system 3.364) in p.u, respectively, at 100 MVA base. 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.

### 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.
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
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]:

\[
C(A, B) = \frac{|\{b \in B, \exists a \in A : a < b\}|}{|B|}
\]

(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


