Appl Intell (2018) 48:2304–2314
https://doi.org/10.1007/s10489-017-1081-2

Application of improved bat algorithm in optimal power flow problem

Yanbin Yuan[1] · Xiaotao Wu[2] · Pengtao Wang[2] · Xiaohui Yuan[2]

Published online: 7 November 2017

Abstract This paper proposes an improved bat algorithm to solve multi-objective optimal power flow problem (MOPF) based on the weighted method. The MOPF problem is formulated as a non-linear constrained optimization problem where two objective functions (minimization of fuel cost and emission) and various constraints are considered. After having found the Pareto solutions with the improved bat algorithm, the fuzzy set theory is used to find the compromise solution. Finally, the IEEE 57-bus system is applied to verify the performance of the proposed method for the MOPF problem. The results are compared with those obtained by the state-of-the-art optimization algorithms reported in literature. The simulation results demonstrate the superiority of the proposed method for solving the MOPF problem in terms of solution quality.

Keywords Bat algorithm · Multi-objective optimization · Optimal power flow · Fuzzy logic

1 Introduction

The optimal power flow (OPF) proposed by Carpentier in 1962 is one of the challenges for economic operation of power systems [1] and it has attracted the attention of many scholars. The main purpose of the OPF problem is to determine the values of control variables (active power and voltage magnitude of generator buses, reactive power of shunt capacitors and tap setting of transformers) for minimum of objective function, while satisfying various physical and operational constraints in power systems. In recent years, due to the crisis of fossil fuel, energy conservation has become a big issue for power systems operation. The traditional OPF problem only takes the minimization of the fuel cost into consideration, thus leading to pollution emissions. Therefore, this paper applies multi-objective optimization method to deal with two-objective OPF problem (minimum of fuel cost and emission).

Many traditional optimization methods have been adopted to solve the OPF problem in the past decades, such as linear programming [2], nonlinear programming [3], Newton method [4], integer quadratic programming [5] and interior point method [6]. When dealing with single objective OPF problem, the traditional methods still have some drawbacks, such as limitation by the continuity and differentiability of objective function and constraints, slow convergence and trapping into local extremum.

Over the past few years, heuristic algorithms have been developed in the solution of large-scale nonconvex nonsmooth constrained optimization problem [7–10]. Therefore, many scholars have also focused on the heuristic algorithm for the OPF problem, such as genetic algorithm [11, 12], particle swarm optimization [13], evolutionary programming [14], bacteria foraging algorithm [15], artificial bee colony algorithm [16], invasive weed optimization algorithm [17], gravitational search algorithm [18, 19] and differential evolution [20]. The results show that the heuristic algorithms have the superiority over the OPF problem. Recently, application of heuristic algorithms to

✉ Xiaohui Yuan
yxh71@163.com

[1] School of Resource and Environmental Engineering, Wuhan University of Technology, 430070, Wuhan, China

[2] School of Hydropower and Information Engineering, Huazhong University of Science and Technology, 430074, Wuhan, China

solve multi-objective optimal power flow (MOPF) problem becomes a hotspot. Many researchers have solved the MOPF problem with heuristic algorithms, such as shuffle frog leaping algorithm [21], teaching–learning based optimization method [22], imperialist competitive algorithm [23], biogeography-based optimization method [24], improved artificial bee colony algorithm [25], multi-objective evolutionary algorithm based decomposition approach [26] and improved strength Pareto evolutionary algorithm [27]. The results demonstrate that the heuristic algorithms are still a promising way for solving the MOPF problem.

In 2010, Yang X. proposed a new heuristic search algorithm named bat algorithm (BA) [28]. Nowadays, BA has been successfully applied to many industrial fields [29–34], which shows that BA is a promising method for solving optimization problems. In our previous work, the MOPF problem has been solved by improved strength Pareto evolutionary algorithm (ISPEA2) [27]. In ISPEA2, improved strategies of environmental selection, external elite population update and local search are embedded in the original strength Pareto evolutionary algorithm. Although the ISPEA2 obtains less total cost and emissions than some other optimization methods for solving the MOPF problem, we find out that it is necessary to further study optimization method for achieving higher quality solutions. Therefore, this paper proposes an improved multi-objective bat algorithm (IMOBA) to solve the OPF problem based on the advantages of bat algorithm and the shortcomings of the traditional optimization methods. Compared with ISPEA2, the proposed IMOBA adopts different strategy to handle multiple objective functions in the OPF problem and it is easy to be implemented. Moreover, improvement of strategies in IMOBA and ISPEA2 are far apart from each other. The main novelties of this paper can be briefly summarized as follows. To our best knowledge, this may be the first try of BA to the solution of the MOPF problem. We equip BA with suitable constraint handling technique to solve complicated optimization problem. To enhance the feasibility of Pareto solutions of the MOPF problem, a mixed constraint handling technique is put forward. The mechanism applies both heuristic-adjusted procedure and penalty function to handle various complicated constraints of the MOPF problem. In order to obtain uniformly distributed Pareto solutions in objective space, the fixed weight coefficient is substituted by the self-adaptive inertia weight in BA. In addition, the switch of dynamic flight mode is adopted to modify the velocity update strategy during the evolutionary process, that is, three types of flight modes (normal searching mode, approaching mode and attacking mode) are designed for update of velocity. Finally, the improved BA integrates fuzzy logic approach to determine the compromise solution of Pareto set of the MOPF problem. Furthermore, the IEEE 57-bus test system is used to verify the performance of the improved bat algorithm for solving single-objective and multi-objective OPF problem. To verify the superiority of the proposed method, the results are compared with those of state-of-the-art optimization algorithms and the original bat algorithm. The comparison demonstrates that the proposed method can obtain solutions with higher quality and it is superior to other methods for solving the OPF problem.

The rest of this paper is organized as follows. The MOPF problem is formulated in Section 2. Overview of multi-objective optimization problem is described in Section 3. Section 4 presents the improved bat algorithm for solving the OPF problem. Case study is provided in Section 5. Section 6 gives the conclusions.

2 Formulation of multi-objective optimal power flow problem

In general, the mathematical model of the MOPF problem can be formulated as:

$$Min\ J(x,u) = \{ J_1(x,u), \cdots, J_i(x,u), \cdots, J_m(x,u) \} \tag{1}$$

$$g_i(x,u) = 0, \quad i = 1, 2, \cdots, KL \tag{2}$$

$$h_j(x,u) \le 0, \quad j = 1, 2, \cdots, HL \tag{3}$$

where J_i $(x,\ u)$ is the i-th objective function of the OPF problem; m is the number of objective functions; $g_i(x,u)$ and $h_j(x,u)$ represent the i-th equality constraint and the j-th inequality constraint, respectively; KL and HL are the number of equality and inequality constraints, respectively.

u is control variables including generator active power P_G, generator voltage magnitude V_G, reactive power of shunt compensator Q_C and transformer tap setting T. It can be described as:

$$u = \left[P_{G_1}, \cdots P_{G_{N_G}}, V_{G_1}, \cdots V_{G_{N_G}}, Q_{C_1}, \cdots Q_{C_{N_C}}, T_1, \cdots T_{N_T} \right] \tag{4}$$

where N_G is the number of generators; N_C is the number of shunt compensators; N_T is the number of transformers.

x is the dependent variables including active power of the slack bus P_{G_1}, load bus voltage magnitude V_L, generator reactive power Q_G and transmission line loading S_L:

$$x = \left[P_{G_1},\ V_{L_1}, \cdots, V_{L_{N_L}},\ Q_{G_1}, \cdots, Q_{G_{N_G}},\ S_{L_1}, \cdots, S_{L_{N_L}} \right] \tag{5}$$

where N_L is the number of transmission lines.

2.1 Objective function

Two objective functions are composed of the total generation fuel cost and emissions of power systems.

2.1.1 Fuel cost function

The minimization of the total fuel cost of power systems can be described as:

$$Min\ FC(x,\ u) = \sum_{i=1}^{N_G} f_i(P_{G_i}) \tag{6}$$

$f_i(P_{G_i})$ is the fuel cost of the i-th generator which can be expressed as:

$$f_i(P_{G_i}) = a_i P_{G_i}^2 + b_i P_{G_i} + c_i \tag{7}$$

where a_i, b_i, c_i are the fuel cost coefficients of the i-th generator; P_{G_i} is the active power of the i-th generator.

2.1.2 Emission function

The minimization of NO_X and SO_X emissions in power systems per hour can be defined by:

$$Min\ EM(x,\ u) = \sum_{i=1}^{N_G} e_i(P_{G_i}) \tag{8}$$

$$e_i(P_{G_i}) = \gamma_i P_{G_i}^2 + \beta_i P_{G_i} + \alpha_i + \varsigma_i \exp(\lambda_i P_{G_i}) \tag{9}$$

where e_i is the amount of the i-th generator emission; α_i, β_i, γ_i, ζ_i, λ_i are the emission coefficients of the i-th generator.

2.2 Constraints

2.2.1 Inequality constraints

1) Generator constraints

The values of active power, reactive power and voltage magnitude of each generator in power systems should be restricted between their maximum and minimum values.

$$P_{G_i}^{\min} \le P_{G_i} \le P_{G_i}^{\max}, \quad i = 1, 2, \ldots, N_G \tag{10}$$

$$Q_{G_i}^{\min} \le Q_{G_i} \le Q_{G_i}^{\max}, \quad i = 1, 2, \ldots, N_G \tag{11}$$

$$V_{G_i}^{\min} \le V_{G_i} \le V_{G_i}^{\max}, \quad i = 1, 2, \ldots, N_G \tag{12}$$

where $P_{G_i}^{\min}$ and $P_{G_i}^{\max}$ are the minimum and maximum active power limits of the i-th generator; Q_{G_i} is the reactive power of the i-th generator; $Q_{G_i}^{\min}$ and $Q_{G_i}^{\max}$ are the minimum and maximum reactive power limits of the i-th generator; V_{G_i} is the voltage of the i-th generator; $V_{G_i}^{\min}$ and $V_{G_i}^{\max}$ are the minimum and maximum voltage limits of the i-th generator.

2) Reactive compensation constraints

$$Q_{C_i}^{\min} \le Q_{C_i} \le Q_{C_i}^{\max}, \quad i = 1, 2, \cdots, N_C \tag{13}$$

where Q_{C_i} is the reactive power injection by the i-th shunt compensator; $Q_{C_i}^{\min}$ and $Q_{C_i}^{\max}$ are the minimum and maximum reactive power injection limits of the i-th shunt compensator.

3) Transformer tap-setting constraints

$$T_i^{\min} \le T_i \le T_i^{\max}, \quad i = 1, 2, \ldots, N_T \tag{14}$$

where T_i is the tap-setting of the i-th transformer; $T_i^{\min}$ and $T_i^{\max}$ are the minimum and maximum tap-setting limits of the i-th transformer.

4) Load bus voltage constraints

$$V_{L_i}^{\min} \le V_{L_i} \le V_{L_i}^{\max}, \quad i = 1, 2, \cdots, N_{TL} \tag{15}$$

where V_{L_i} is the voltage of the i-th load bus; $V_{L_i}^{\min}$ and $V_{L_i}^{\max}$ are the minimum and maximum voltage limits of the i-th load bus; N_{TL} is the number of load buses.

5) Power flow of transmission line constraints

Considering the security of power systems, each transmission line has maximum power flow.

$$S_{L_i} \le S_{L_i}^{\max}, \quad i = 1, 2, \ldots, N_L \tag{16}$$

where S_{L_i} is the transmission line loading of the i-th branch; $S_{L_i}^{\max}$ is the maximum apparent power flow limit of the i-th branch.

2.2.2 Equality constraints

1) Power flow equations

The active power and reactive power of each bus in power systems should satisfy power flow equations:

$$P_{G_i} - P_{D_i} = V_i \sum_{j=1}^{N_B} V_j \left(G_{ij} \cos\theta_{ij} + B_{ij} \sin\theta_{ij}\right) \tag{17}$$

$$Q_{G_i} - Q_{D_i} = V_i \sum_{j=1}^{N_B} V_j \left(G_{ij} \sin\theta_{ij} - B_{ij} \cos\theta_{ij}\right) \tag{18}$$

where P_{G_i} and Q_{G_i} are the active and reactive power at bus i; P_{D_i} and Q_{D_i} are the active and reactive power loads at bus i; G_{ij} and B_{ij} are respectively the conductance and susceptance of the transmission line connecting the i-th bus and the j-th bus; V_i and V_j are the voltage magnitudes of bus i and bus j; θ_{ij} is the difference of voltage phase angle between bus i and bus j.

2) Power balance equation

$$\sum_{i=1}^{N_G} P_{G_i} = P_{load} + P_{loss} \tag{19}$$

where P_{load} is the total load of the system; P_{loss} is the total network loss.

3 Overview of multi-objective optimization problem

A multi-objective optimization problem can be mathematically described as [35]:

$$Min\ F(x) = \{ f_1(x),\ f_2(x), \cdots, f_m(x) \} \tag{20}$$

where $f_i(x)$ is the i-th objective function; x is the decision variable and m is the number of objective functions.

3.1 Pareto set

As there are many conflict objectives in multi-objective optimization problem, an optimal solution of a certain objective may be the worst solution for other objectives. The improvement of one objective will deteriorate another objective simultaneously [36]. Assuming x_1 and x_2 are two solutions of a multi-objective optimization problem, when each objective function value of x_1 is not worse than that of x_2, and x_1 can find at least one objective better than the corresponding one of x_2, namely, the (21) and (22) are satisfied, we can say solution x_1 dominates solution x_2, expressed as $x_1 \succ x_2$ [37].

$$\forall\, i \in \{1, 2, \cdots, m\} :\ f_i(x_1) \le f_i(x_2) \tag{21}$$

$$\exists\, j \in \{1, 2, \cdots, m\} :\ f_j(x_1) < f_j(x_2) \tag{22}$$

For a given multi-objective problem, the Pareto solution (or the non-dominated solution) x^* can be defined as: x^* is a feasible solution and there are no other solutions that dominate x^* in the feasible region Ω [38]:

$$\neg\exists\, x_k \in \Omega :\ x_k \succ x* \tag{23}$$

where Ω is the feasible region of multi-objective optimization problem.

All Pareto solutions x^* form Pareto set (PS), which can be expressed as:

$$PS = \{ x* \in \Omega \mid \neg\exists\, y \in \Omega,\ F(y) \succ F(x*) \} \tag{24}$$

Another important concept is Pareto front, which is represented by PF. It is the set of the values of objective functions corresponding to the Pareto solutions in PS [39]:

$$PF = \{F(x) = \{f_1(x), f_2(x), \cdots, f_m(x)\} | x \in PS\} \tag{25}$$

3.2 Compromise solution

After having found the Pareto set, the decision maker should choose a compromise solution. Fuzzy set theory is applied to determine compromise solution. The corresponding procedures can be described as follows.

Step 1: Record the maximum value $F_i^{\max}$ and minimum value $F_i^{\min}$ of the i-th objective function after searching over the Pareto set.

Step 2: Use (26) to calculate u_{i}^k for the k-th non-dominated solution.

$$u_{\mathrm{i}}^k = \begin{cases} 1 & if\ F_i < F_i^{\min} \\ \frac{F_i^{\max} - F_i}{F_i^{\max} - F_i^{\min}} & if\ F_i^{\min} \le F_i \le F_i^{\max} \\ 0 & if\ F_i > F_i^{\max} \end{cases} \tag{26}$$

Step 3: Equation (27) is adopted to normalize u_{i}^k for the k-th non-dominated solution.

$$U^k = \frac{\sum_{i=1}^{N} u_i^k}{\sum_{k=1}^{M}\sum_{i=1}^{N} u_i^k} \tag{27}$$

where N and M are the number of objective functions and Pareto solutions, respectively.

After implementing the above steps, the values of U^k for all non-dominated solutions can be obtained. The one with the maximum U^k in Pareto set is chosen as the compromise solution of the MOPF problem.

4 Improved multi-objective bat algorithm and application to the MOPF problem

4.1 Multi-objective optimization bat algorithm

4.1.1 Overview of bat algorithm

The bat algorithm (BA) proposed by Yang in 2010 is a bionics algorithm [28], which is derived from simulation of the bat's foraging behavior by echolocation. The bat is the flying mammal, which has the amazing echolocation ability for navigating and searching for prey. During the process of prey searching, a bat releases a series of loud ultrasound waves. According to the echoes, it uses time delay of two ears and the loudness variation to identify the prey position in searching space. In BA, the similar searching mechanism is employed to solve the following optimization problem.

$$\begin{aligned} &\min\ F(X) \\ &\text{s.t. } X \in R^D \end{aligned} \tag{28}$$

Assume $X_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,D})$ is a feasible solution, which is represented by a bat in BA, the first step randomly generates N bats in searching space R^D:

$$\begin{aligned} x_{i,j} &= x_j^{min} + rand\,(0,1) \times \left(x_j^{max} - x_j^{min}\right) \\ i &= 1, 2, \ldots, N;\ j = 1, 2, \ldots, D \end{aligned} \tag{29}$$

where N is the number of bats; D is the number of variables; $x_j^{\max}$ and $x_j^{\min}$ are the maximum and minimum limit values of the j-th variable, respectively.

After initialization of each bat, the bats begin their movement for the prey using the way of echolocation. The movement of each bat can be described as follows.

$$Q_i^t = Q_{\min} + (Q_{\max} - Q_{\min}) \cdot rand(0, 1) \tag{30}$$

$$v_{ij}^{t+1} = v_{ij}^t + (x_{ij}^t - x_{best,j}) \cdot Q_{ij}^t \tag{31}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{32}$$

where Q_i is the corresponding frequency of the i-th bat; Q_{max} and Q_{min} are the values of frequency limit; $rand(0,1)$ is a random number in the range of 0 and 1; v_i^t and x_i^t are respectively the velocity and position of the i-th bat at time t; x_{best} is the best position of the current bat.

According to (30), each bat is randomly assigned a frequency. Using the frequency, the position x_i^t at time t and the current optimal position x_{best}, we can get the velocity v_i^{t+1} of the i-th bat at time $t+1$ with (31). After getting v_i^{t+1}, the position of the i-th bat will be updated using (32).

After the position update is carried out on all bats, random walk is employed to search optimal solution locally. The current optimal bat begins the local search by using the loudness and random walk.

$$x^t = x_{best}^t + \varepsilon \cdot A^t \tag{33}$$

where ε is a random number and$\varepsilon \in (-1, 1)$; A^t is the loudness at time t.

It should be noticed that the local search controlled by the loudness A^t is launched with the probability r_i, which is called the pulse emission rate in BA. A random number in the range of 0 and 1 is generated and compared with the pulse emission rate. If the former is larger, then they begin search locally using (33); otherwise, the current optimal bat position x_{best} does not change. So the loudness A^t decides the random walk range of the local search, and whether to begin local search depends on the pulse emission rate r_i. In nature, after a bat finds the prey, the loudness will decrease while the pulse emission rate will increase. The variations of two parameters in BA imitate bat's behavior in nature, and the variation of two parameters can be depicted as:

$$A_i^{t+1} = \alpha \cdot A_i^t \tag{34}$$

$$r_i^{t+1} = r^0 \cdot [1 - \exp(-\gamma t)] \tag{35}$$

where α and γ are two constants in the range of 0 and 1.

4.1.2 Approach of bat algorithm to multi-objective optimization problem

The key idea of BA for solving multi-objective optimization problem is to convert multiple objective functions into a single objective function with weighted method.

$$f_{sum} = \omega_1 \cdot f_1 + \omega_2 \cdot f_2 + \cdots + \omega_N \cdot f_N \tag{36}$$

where f_{sum} is the weight of all objective functions; ω_i is weight coefficient of the i-th objective function, and it must satisfy $\omega_1 + \omega_2 + \cdots + \omega_N = 1$ and $0 < \omega_i < 1$; N is number of objective functions.

Each of weight coefficient ω_i can be either generated randomly between (0, 1) or generated with uniform distribution. This paper generates ω_1 and ω_2 as follows.

$$\omega_1 = \frac{k}{N_{pareto}}, \qquad \omega_2 = 1 - \omega_1 \tag{37}$$

where N_{pareto} is the total number of Pareto solutions; k is an integer variable in the range of 1 and N_{pareto}, which is used to calculate the k-th Pareto solution. According to (37), ω_1 increases from $1/N_{pareto}$ to 1 and ω_2 decreases from $(1-\frac{1}{N_{pareto}})$ to 0.

4.2 Improvement strategies of multi-objective optimization bat algorithm

The velocity update strategy in multi-objective optimization bat algorithm (MOBA) plays an important role in optimization process. The velocity update strategy in the traditional BA only considers the best position information without using the population information, which may either trap into local extremum or be slow convergence. In order to overcome the shortcomings of MOBA, this paper proposed an improved multi-objective bat algorithm (IMOBA). In IMOBA, two ways of improvement have been put forward: (i) add self-adaptive inertia weight to velocity update rule so as to fully utilize population information; (ii) dynamic flight mode is applied to modify velocity update strategy. In IMOBA, three types of flight modes are designed for velocity update strategy: searching mode, approaching mode and attacking mode.

4.2.1 Self-adaptive inertia weight strategy

To dynamically adjust the change of velocity with the best position information, this study adds self-adaptive inertia weight to the rule of velocity update. The adjustment method is as follows. When the bat is far away from the prey, the bat flies at a relatively faster speed (using a larger inertia weight coefficient) so as to be close to the prey as soon as possible. While the bat is close to the prey, it will use a relatively slower flight speed (smaller inertia weight coefficient) so as to gradually approach and capture the prey (finding the optimal solution). After using the self-adaptive inertia weight for update of velocity, the bat can dynamically adjust its flight speed and direction in the prey searching process.

The strategy of velocity update by adding the self-adaptive inertia weight can be described as:

$$v_{ij}^{t+1} = \mu_{ij} \cdot v_{ij}^t + (x_{ij}^t - x_{best,j}) \cdot Q_{ij}^t \tag{38}$$

$$\mu_{ij}^{t} = \mu_0 \cdot \left(1 - \exp\left(-k \cdot \left|x_{ij}^{t-1} - x_{best,j}^{t-1}\right|\right)\right) \tag{39}$$

where μ_{ij} is the self-adaptive inertia weight; k is the amplitude of regulating coefficient.

4.2.2 Dynamic flight mode

1) Normal searching mode

In this mode, the bats do not find the prey and they only search around their own best region by memory. The strategy of velocity update can be shown as:

$$v_{ij}^{t+1} = \mu_{ij} \cdot v_{ij}^{t} + (x_{gj} - x_{ij}^{t-1}) \cdot Q_{ij}^{t} \tag{40}$$

where μ_{ij} is the self-adaptive inertia; x_{gj} is the j-th coordinate of the best position found so far.

2) Approaching mode

When a bat finds the prey which does not reach the attacking position, the bat will fly directly to approach it, which can be described as follows.

$$v_{ij}^{t+1} = \mu_{ij} \cdot v_{ij}^{t} + c \cdot rand(0,1) \cdot (x_{ij}^{t} - x_{best,j}) \tag{41}$$

where c is velocity regulation factor.

3) Attacking mode

When a bat arrives at the attacking position of the prey, it will adopt a flexible way to attack the prey in order to increase the probability of the capture for the prey.

$$\begin{cases} v_{ij}^{t+1} = x_{best,k} \\ x_{ij}^{t+1} = v_{ij}^{t+1} \end{cases} \tag{42}$$

where k is a random integer.

In IMOBA, the flight mode switch basis is proposed to implement three flight modes alternately. Before presenting the switch basis, a parameter should be defined.

$$\psi_i = ||x_{best} - x_i||, \quad q_i = \psi_i / \sum_{j=1}^{popsize} \psi_j \tag{43}$$

The larger the distance between the bat i and the best position is, the larger q_i will be given. The flight mode switch basis can be determined as follows.

(i) If $q_i > \beta$, the bat will select the searching mode to update velocity.

(ii) If $|t/T_{max} - 0.5|\beta < q_i < \beta$, the bat will select the approaching mode to approach the prey.

(iii) If $q_i < C|t/T_{max} - 0.5|\beta$, the bat will select the attacking mode to attack the prey.

4.3 Application of IMOBA in the MOPF problem

4.3.1 Handling constraints

When using IMOBA to solve the MOPF problem, the control variables are represented by the bats. First, the power flow should be calculated before calculating the objective functions. Secondly, the equality and inequality constraints of the MOPF problem should be satisfied. For the equality constraints (17) and (18), the Newton-Raphson method is adopted to calculate the power flow, whose calculation steps are shown in Fig. 1.

The penalty function method is applied to handle the inequality constraints. The penalty function is formed as follows.

$$J_{aug} = f_o + f_p \tag{44}$$

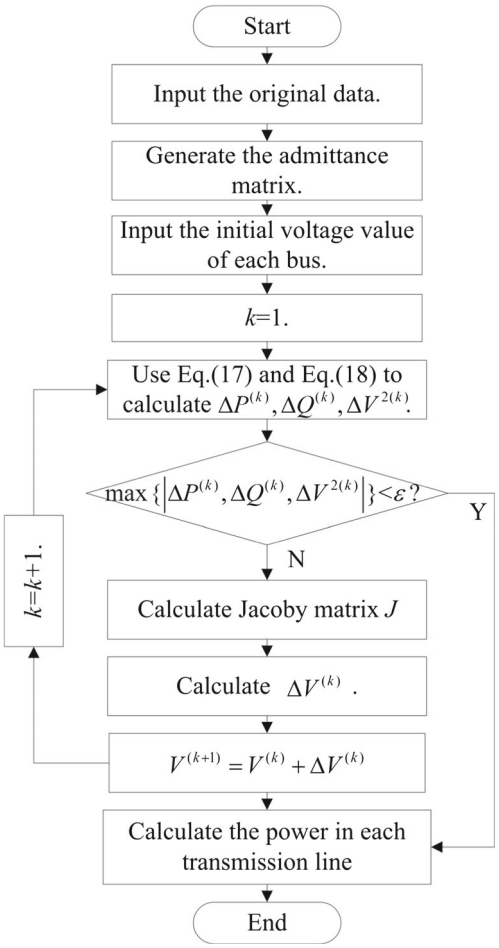

Fig. 1 Flow chart of Newton-Raphson power flow calculation method

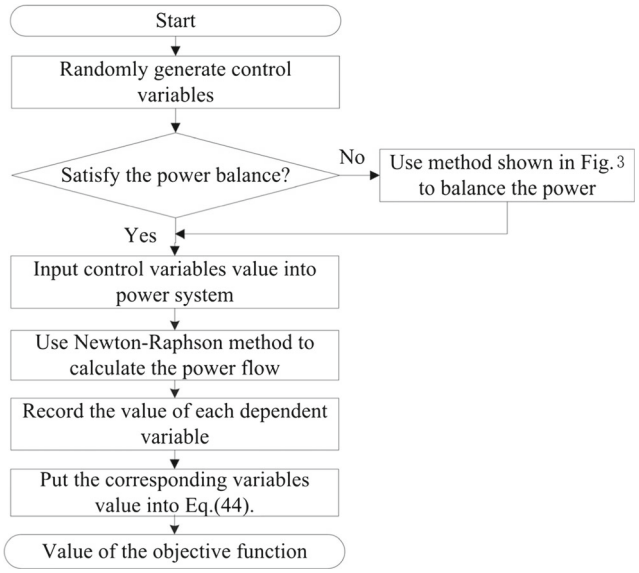

Fig. 2 Flow chart of the OPF objective function calculation

where f_o is the objective function in (36); f_p is the penalty term considering the dependent variables constraints, which can be defined as follows.

$$f_p = k_P\left(P_{G1}-P_{G1}^{lim}\right)^2+k_Q\sum_{i=1}^{N_G}\left(Q_{Gi}-Q_{Gi}^{lim}\right)^2 +k_V\sum_{i=1}^{N_L}\left(V_{Li}-V_{Li}^{lim}\right)^2+k_S\sum_{i=1}^{N_L}\left(S_{Li}-S_{Li}^{lim}\right)^2 \quad (45)$$

where k_P, k_Q, k_V and k_S are the penalty coefficients; x^{lim} is the maximum or minimum value of the corresponding dependent variable, which can be defined as:

$$x^{lim}=\begin{cases} x^{\max} & if\ x>x^{\max} \\ x^{\min} & if\ x<x^{\min} \end{cases} \quad (46)$$

After finishing the power flow calculation and the penalty function process, calculation of objective functions is shown in Fig. 2.

4.3.2 Implementation procedures of IMOBA for the MOPF problem

The main steps of IMOBA for solving the MOPF problem can be summarized as follows.

Step 1. Initialize the parameters of IMOBA; let the calculation time $k = 1$ and use (37) to generate ω_1 and ω_2.

Step 2. Generate the initial population and let the iteration time $i = 0$.

Step 3. Perform the active power balance treatment shown in Fig. 3. Calculate power flow with Newton-Raphson method; then record the dependent variables.

Step 4. Put into the corresponding control variables to fuel cost function and emission function; then use the weighted method to combine two functions together as a single objective function. Use the penalty function method to calculate each bat's objective function, and compare them to find the best position of the current bat.

Step 5. Initialize the pulse emission rate, and calculate the self-adaptive inertia weight μ_{ij} for each bat. Use the flight mode switch basis to choose the flight mode and update the bat's position.

Step 6. Generate a random number r_1. If $r_1 > r_i$, perform random walk around the best bat, and update the position of the best bat based on the random walk.

Step 7. Generate a random number r_2. If $r_2 > A_i$ and the bat's position is improved, replace the old position with the new one.

Step 8. When the condition of Step7 is satisfied, (34) and (35) are used to update the loudness and the pulse emission rate; otherwise, ignore Step 8.

Step 9. Record the current optimal solution; if $i < Maxiter$, go to Step 3; otherwise, go to Step 10.

Step 10. Update the two weights using (37) and use weighted method to calculate the next Pareto solution.

Step 11. $k = k + 1$. If $k = N_{pareto}$, output the Pareto front and the corresponding control variables; otherwise, go to Step 3.

5 Case study

In this section, case study is carried out to investigate the performance of the proposed IMOBA for solving the OPF problem. The IEEE 57-bus test system is selected to validate the effectiveness of the IMOBA. All simulations are implemented in MATLAB on a laptop with 2.8 GHz Intel Core i5-3360 CPU and 8GB RAM. The main parameters of the IMOBA are set as: $Popsize = 20$, $Q_{max} = 2$, $Q_{min} = 0$, $\alpha = 0.9$, $\gamma = 0.9$, $Maxiter = 200$ and $N_{pareto} = 30$.

5.1 System data

The IEEE 57-bus system has seven generators placed on the buses 1, 2, 3, 6, 8, 9, and 12. Seventeen transformers are equipped in the system, and three reactive power compensators are respectively installed in the buses 18, 25, 53.

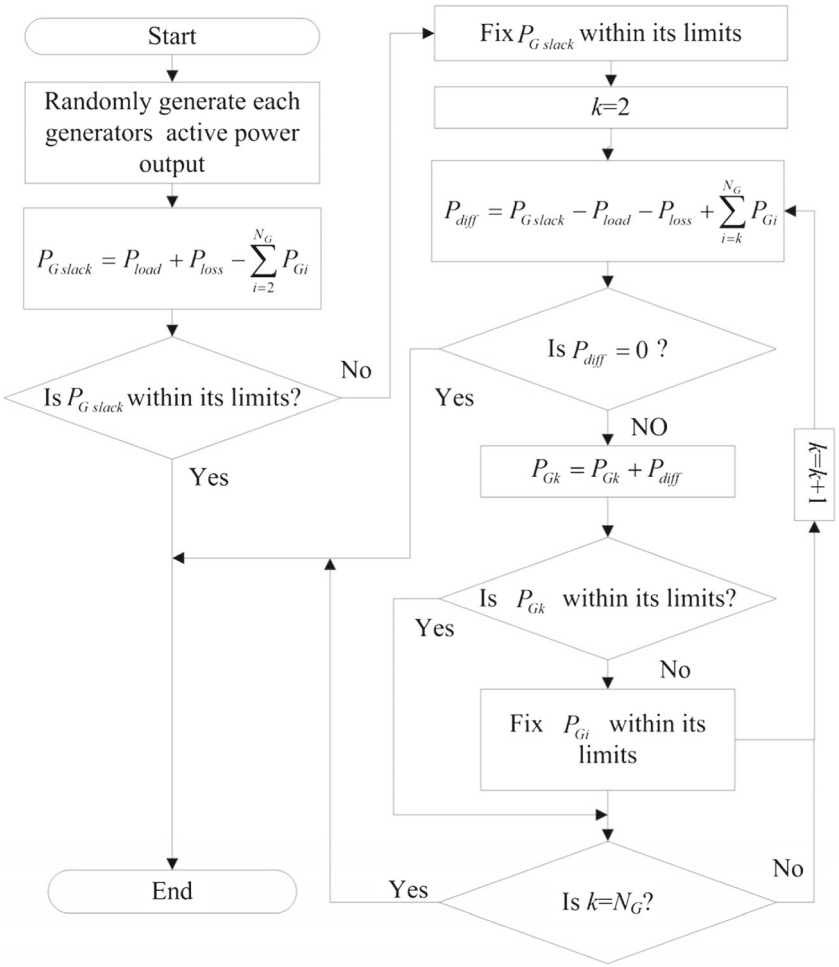

Fig. 3 Flow chart of the active balance equation calculation

The ranges of transformer ratios are 0.9–1.1p.u., and each reactive compensator capacity is 30 MVA. The ranges of the bus voltage amplitude are 0.94–1.06p.u. The total active power load is 12.508p.u. with 100MVA base power. The topology and other data of the IEEE 57-bus system can be seen in Reference [21].

5.2 Single-objective OPF problem

In this case, the comparison between various methods is carried out to verify the performance of the IMOBA for solving single-objective OPF problem. Each objective function is deliberated individually. IEEE 57-bus test system is implemented with the IMOBA and original MOBA. During the simulation, 10 independent trails are executed and the best solution is found for single-objective OPF problem. In order to show the competitiveness of the proposed approach, IMOBA is compared with other algorithms reported in literature [22, 23, 25]. These algorithms involve artificial bee colony algorithm (ABC), improved ABC (IABC), cat swarm optimization (CSO), teaching-learning-based optimization algorithm (TLBO), modified TLBO method (MTLBO), genetic algorithm (GA), particle swarm optimization (PSO), differential evolution (DE) and imperialist competitive algorithm (ICA).

Table 1 Comparison of the results for minimization of the total fuel cost with different methods

Methods	The total fuel cost ($/h)			Standard deviation
	Min	Mean	Max	
IMOBA	41673	41720	41795	35.1
MOBA	41716	41808	41889	60.6
ABC	41781	41840	41927	38.2
IABC	41684	41698	41711	7.8
CSO	41696	41708	41719	13.7
TLBO	41689	41693	41698	12.7
GA	41712	41720	41734	18.2
PSO	41695	41714	41718	13.0
DE	41710	41715	41720	14.9
ICA	41710	41713	41716	11.6
MTLBO	41638	41651	41663	9.9
GSA	41696	–	–	–

– denotes "not available in literature"

Table 2 Comparison of the results for minimization of the total emission with different methods

Methods	The total emission (ton/h)			Standard deviation
	Min	Mean	Max	
IMOBA	1.0656	1.0718	1.1358	0.0212
MOBA	1.0726	1.0948	1.1428	0.0241
ABC	1.2048	1.3262	1.6351	0.1202
IABC	1.0484	1.1382	1.3709	0.0789
TLBO	1.0781	1.0802	1.9716	0.018
MTLBO	1.0772	1.0781	1.9152	0.011

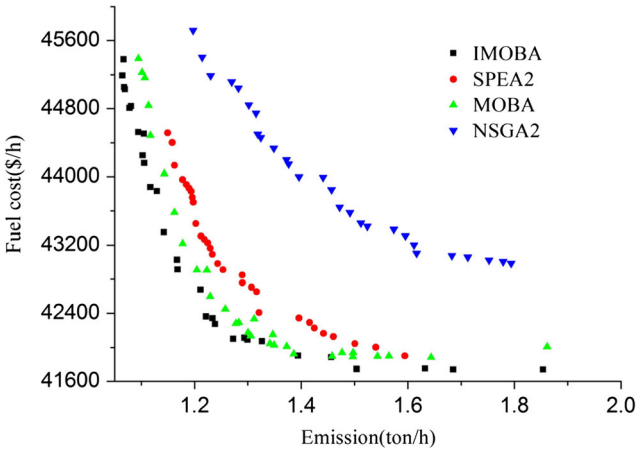

Fig. 5 Comparison of Pareto front for different methods

The best minimization of fuel cost and emission solutions obtained by the IMOBA in 10 trials is given in Tables 1 and 2. The results indicate that the application of IMOBA leads to 41673 $/h fuel cost and 1.0656 ton/h emission. By examining the results in Tables 1 and 2, it turns out that the best fuel cost and emission calculated by IMOBA are less in comparison with reported results in literatures and other algorithms. The critical performance indexes for all algorithms such as minimization fuel cost (Min), mean fuel cost (Mean), maximum fuel cost (Max), and the corresponding standard deviation for 10 independent runs are also shown in Tables 1 and 2. To illustrate the convergence characteristics of IMOBA and original MOBA, the minimum objective values of single fuel cost and single emission over 200 iterations are plotted in Fig. 4. As shown in Fig. 4, IMOBA has faster convergence rate than the original MOBA.

In a word, it is clear that the proposed IMOBA has better performance for solving the OPF problem in comparison with other algorithms reported in literatures. The superiority of the proposed IMOBA is obvious and obtains better results.

5.3 Multi-objective OPF problem

In this case, the proposed IMOBA is employed to solve the MOPF problem where the total fuel cost and the total emissions are deliberated as objective functions. In order to demonstrate the effectiveness of IMOBA, the results are compared with those of SPEA2, NSGA2 and MOBA. The Pareto front of IMOBA is compared with the above three algorithms shown in Fig. 5, from which we can clearly find that the Pareto front of IMOBA gives better distributed Pareto solutions. The Pareto front of the IMOBA totally lies inside the concave portion of other Pareto fronts. The corresponding compromise solution of IMOBA, SPEA2, NSGA2, MOBA and other algorithms reported in literature [22, 25, 27] are listed in Table 3,

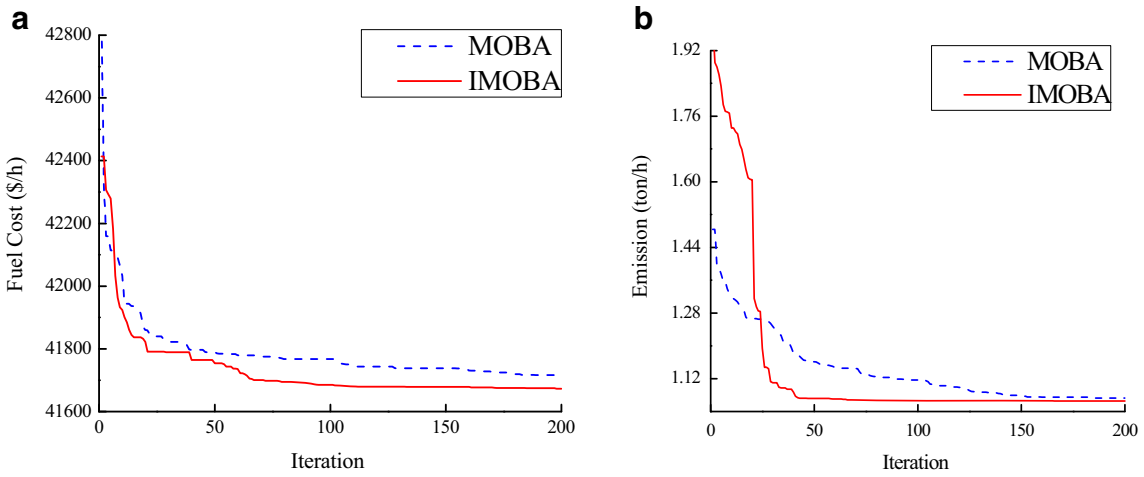

Fig. 4 Convergence characteristics of the IMOBA and MOBA algorithm. **a** The total fuel cost; **b** the total emissionss

Table 3 Comparison of compromise solution for different algorithms

Methods	Emission(ton/h)	Fuel cost($/h)
IMOBA	1.2387	42271
MOBA	1.2577	42449
NSGA2	1.2979	43568
SPEA2	1.4054	42320
ISPEA2	1.2904	42445
TLBO	1.2810	42510
MTLBO	1.2780	42490
ABC	1.3396	44741
IABC	1.2643	44152

which indicates obviously that the compromise solution of IMOBA dominates other algorithms. Therefore, the compromise solution of IMOBA is the best solution among all algorithms.

The index of C-Metric is presented to further check the calculation performance and superiority of IMOBA for solving the MOPF problem. The C-Metric is mainly applied to evaluate the degree of which one multi-objective optimization algorithm dominates another one. Suppose A_1 and A_2 are the Pareto sets of two different multi-objective algorithms, the C-Metric of A_1 and A_2 is defined as the ratio of the total number of solutions in A_2 which dominates solutions in A_1 and the number of A_2. The formula of C-Metric is expressed as [40].

$$C(A_1, A_2) = \frac{|a_2 \in A_2, \exists a_1 \in A_1 : a_1 < a_2|}{|A_2|} \tag{47}$$

Table 4 shows the C-Metric values of IMOBA and other algorithms, which indicates that there is no solutions in SPEA2, NSGA2, and MOBA dominate solutions in IMOBA. Therefore, the IMOBA obtains superior Pareto set.

To study the robustness of IMOBA for the IEEE 57-bus system, IMOBA is employed to solve the MOPF problem for ten independent trials in succession. Then it makes statistics for compromise solution of the Pareto set. The ultimate Pareto set is determined by choosing the best compromise solution in ten trials. Figure 6 shows the distribution of compromise solutions for ten repetitions. It is clear that the IMOBA has good robustness for solving the MOPF problem.

Table 4 C-Metric of IMOBA with other algorithms

Method	C-Metric
C(IMOBA,SPEA2)	100%
C(IMOBA, MOBA)	100%
C(IMOBA,NSGA2)	100%

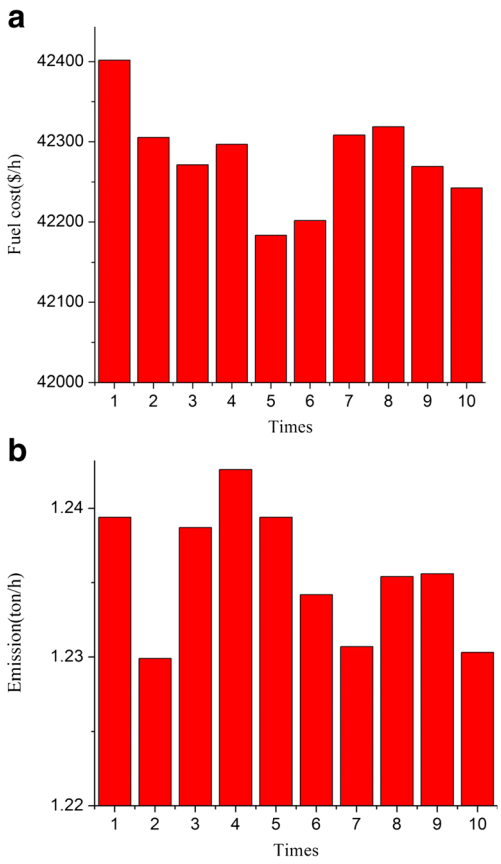

Fig. 6 Compromise solutions of objective functions distribution for 10 trials: **a** The total fuel cost; **b** The total emissions

6 Conclusions

An improved multi-objective optimization bat algorithm (IMOBA) is proposed to solve two-objective OPF problem. The proposed method is effectively implemented on the IEEE 57-bus system to deal with the OPF problem. In order to demonstrate the effectiveness and superiority of the proposed IMOBA method, the results are compared with other optimization algorithms reported in the literatures and the original MOBA algorithm. From the comparison results, we can conclude that the IMOBA has good performance when solving the OPF problem. The Pareto front distributes well and the diversity characteristic is satisfactory. According to the analysis, the IMOBA has good robustness and better distribution of Pareto front for solving the OPF problem.

Acknowledgments This work was supported by National Natural Science Foundation of China (No. 51379080, No. 41571514), the Fundamental Research Funds for the Central Universities (No. 2017KFYXJJ204) and Hubei Provincial Collaborative Innovation Center for New Energy Microgrid in China Three Gorges University (2015KJX09). Yanbin Yuan and Xiaotao Wu are the Co-first authors of this paper. These authors contributed equally to this work.

References

1. Carpentier J (1962) Contribution to the economic dispatch problem (in French). Bull Soc Franc Elect 8:431–447
2. Zehar K, Sayah S (2008) Optimal power flow with environmental constraint using a fast successive linear programming algorithm: application to the algerian power system. Energy Convers Manag 49:3362–3366
3. Wei H, Sasaki H, Kubokawa J (2000) Large scale hydrothermal optimal power flow problems based on interior point nonlinear programming. IEEE Trans Power Syst 15(1):396–403
4. Kazemtabrizi B, Acha E (2014) An advanced STATCOM model for optimal power flows using Newton's method. IEEE Trans Power Syst 29(2):514–525
5. Sivasubramani S, Swarup KS (2011) Sequential quadratic programming based differential evolution algorithm for optimal power flow problem. IET Gener Transm Distrib 5(11):1149–1154
6. Sousa AA, Torres GL, Canizares CA (2011) Robust optimal power flow solution using trust region and interior-point methods. IEEE Trans Power Syst 26(2):487–499
7. Das SP, Achary NS, Padhy S (2016) Novel hybrid SVM-TLBO forecasting model incorporating dimensionality reduction techniques. Appl Intell 45(4):1148–1165
8. Yuan X, Ji B, Zhang S (2014) An improved artificial physical optimization algorithm for dynamic dispatch of generators with valve-point effects and wind power. Energy Convers Manag 82:92–105
9. Liu R, Fan J, Jiao L (2015) Integration of improved predictive model and adaptive differential evolution based dynamic multi-objective evolutionary optimization algorithm. Appl Intell 43:192–207
10. Ji B, Yuan X, Li X (2014) Application of quantum-inspired binary gravitational search algorithm for thermal unit commitment with wind power integration. Energy Convers Manag 87:589–598
11. Attia AF, Al-Turki YA, Abusorrah AM (2012) Optimal power flow using adapted genetic algorithm with adjusting population size. Electr Power Compon Syst 40(11):1285–1299
12. Yuan XH, Zhang YC, Yuan YB (2008) Improved self-adaptive chaotic genetic algorithm for hydrogeneration scheduling. J Water Resour Plan Manag-ASCE 134(4):319–325
13. Singh R, Mukherjee V, Ghoshal S (2016) Particle swarm optimization with an aging leader and challengers algorithm for the solution of optimal power flow problem. Appl Soft Comput 40:161–177
14. Ongsakul W, Tantimaporn T (2006) Optimal power flow by improved evolutionary programming. Electr Power Compon Syst 34:79–95
15. Panda A, Tripathy M (2015) Security constrained optimal power flow solution of wind-thermal generation system using modified bacteria foraging algorithm. Energy 93:816–827
16. Yuan X, Wang P, Yuan Y (2015) A new quantum inspired chaotic artificial bee colony algorithm for optimal power flow problem. Energy Convers Manag 100:1–9
17. Ghasemi M, Ghavidel S, Akbari E (2014) Solving non-linear, non-smooth and non-convex optimal power flow problems using chaotic invasive weed optimization algorithms based on chaos. Energy 73:340–353
18. Duman S, Guvenc U, Sonmez Y (2012) Optimal power flow using gravitational search algorithm. Energy Convers Manag 59:86–95
19. Yuan X, Ji B, Zhang S (2014) A new approach for unit commitment problem via binary gravitational search algorithm. Appl Soft Comput 22:249–260
20. Sayah S, Zehar K (2008) Modified differential evolution algorithm for optimal power flow with non-smooth cost functions. Energy Convers Manag 49(11):3036–3042
21. Niknam T, Jabbari M, Malekpour A (2011) A modified shuffle frog leaping algorithm for multi-objective optimal power flow. Energy 36:6420–6432
22. Shabanpour-Haghighi A, Seifi A, Niknam T (2014) A modified teaching–learning based optimization for multi-objective optimal power flow problem. Energy Convers Manag 77:597–607
23. Ghasemi M, Ghavidel S, Ghanbarian M (2014) Application of imperialist competitive algorithm with its modified techniques for multi-objective optimal power flow problem: a comparative study. Inf Sci 281:225–247
24. Roy P, Ghoshal S, Thakur S (2010) Multi-objective optimal power flow using biogeography-based optimization. Electr Power Compon Syst 38(12):1406–1426
25. He X, Wang W, Jiang J, Xu L (2015) An improved artificial bee colony algorithm and its application to multi-objective optimal power flow. Energies 8:2412–2437
26. Zhang J, Tang Q, Li P (2016) A modified MOEA/D approach to the solution of multi-objective optimal power flow problem. Appl Soft Comput 47:494–514
27. Yuan XH, Zhang BQ, Wang PT, Liang J, Yuan YB, Huang YH, Lei XH (2017) Multi-objective optimal power flow based on improved strength Pareto evolutionary algorithm. Energy 122:70–82
28. Yang X (2011) Bat algorithm for multi-objective optimisation. Int J Bio-Inspired Comput 3(5):267–274
29. Kang M, Kim J, Kim JM (2015) Reliable fault diagnosis for incipient low-speed bearings using fault feature analysis based on a binary bat algorithm. Inf Sci 294:423–438
30. Jaddi NS, Abdullah S, Hamdan AR (2015) Optimization of neural network model using modified bat-inspired algorithm. Appl Soft Comput 37:71–86
31. Adarsh BR, Raghunathan T, Jayabarathi T, Yang XS (2016) Economic dispatch using chaotic bat algorithm. Energy 96:666–675
32. Talafuse TP, Pohl EA (2016) A bat algorithm for the redundancy allocation problem. Eng Optim 48(5):900–910
33. Tharakeshwar TK, Seetharamu KN, Prasad BD (2017) Multi-objective optimization using bat algorithm for shell and tube heat exchangers. Appl Thermal Eng 110:1029–1038
34. Vedik B, Chandel AK (2017) Optimal PMU placement for power system observability using Taguchi binary bat algorithm. Measurement 95:8–20
35. Shang RH, Jiao LC, Liu F, Ma WP (2012) Novel immune clonal algorithm for MO problems. IEEE Trans Evol Comput 16(1):35–50
36. Jiao LC, Wang HD, Shang RH, Liu F (2013) A co-evolutionary multi-objective optimization algorithm based on direction vectors. Inf Sci 228:90–112
37. Shang RH, Luo S, Zhang WT, Stolkin R, Jiao LC (2016) A multiobjective evolutionary algorithm to find community structures based on affinity propagation. Physica A 453:203–227
38. Wang HD, Jiao LC, Xin Yao (2015) Two_Arch2: an improved two-archive algorithm for many-objective optimization. IEEE Trans Evol Comput 19(4):524–541
39. Chen Z, Yuan X, Ji B (2014) Design of a fractional order PID controller for hydraulic turbine regulating system using chaotic non-dominated sorting genetic algorithm II. Energy Convers Manag 84:390–404
40. Shang RH, Wang YY, Wang J, Jiao LC, Wang S, Qi LP (2014) A multi-population cooperative coevolutionary algorithm for multi-objective capacitated arc routing problem. Inf Sci 277:609–642