Electrical Power and Energy Systems 60 (2014) 203–220

Contents lists available at ScienceDirect

Electrical Power and Energy Systems

journal homepage: www.elsevier.com/locate/ijepes

Multi-hive bee foraging algorithm for multi-objective optimal power flow considering the cost, loss, and emission

Hanning Chen [1], Ma Lian Bo [1], Yunlong Zhu *

CAS Key Laboratory of Networked Control System, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China

ARTICLE INFO

Article history:
Received 14 May 2013
Received in revised form 15 February 2014
Accepted 18 February 2014

Keywords:
Bee foraging algorithm
Cooperative coevolution
Multi-objective optimization
Optimal power flow

ABSTRACT

This paper proposes a multi-hive multi-objective bee algorithm (M^2OBA) for optimal power flow (OPF) in power systems. The proposed M^2OBA extend original artificial bee colony (ABC) algorithm to multi-objective and cooperative mode by combining external archive, comprehensive learning, greedy selection, crowding distance, and cooperative search strategy. Our algorithm uses the concept of Pareto dominance and comprehensive learning mechanism to determine the flight direction of a bee and maintains nondominated solution vectors in external archive based on greedy selection and crowing distance strategies. With cooperative search approaches, the single population ABC has been extended to interacting multi-hive model by constructing colony-level interaction topology and information exchange strategies. With six mathematical benchmark functions, M^2OBA is proved to have significantly better performance than three successful multi-objective optimizers, namely the fast non-dominated sorting genetic algorithm (NSGA-II), the multi-objective particle swarm optimizer (MOPSO), and the multi-objective ABC (MOABC), for solving complex multi-objective optimization problems. M^2OBA is then used for solving the real-world OPF problem that considers the cost, loss, and emission impacts as the objective functions. The 30-bus IEEE test system is presented to illustrate the application of the proposed algorithm. The simulation results, which are also compared to NSGA-II, MOPSO, and MOABC, are presented to illustrate the effectiveness and robustness of the proposed method.

© 2014 Elsevier Ltd. All rights reserved.

Introduction

In many real-world optimization applications, the decision maker (DM) is always faced with the multiple noncommensurable and often competing objectives [1–3]. The solutions to these multi-objective optimization (MO) problems often result from both the optimization and decision making process. Such as in real-world power system operation and planning, many system operators need to make decisions with respect to different objectives [4]. In power system control, optimal power flow (OPF) is one of most important MO issues, which in order to find the optimal adjustments of the control variables to minimize the selected objective function while satisfying various physical and operational constraints imposed by equipment and network limitations [5–7].

Many mathematical models and conventional techniques, such as gradient-based optimization algorithms, linear programming, interior point method, and Newton method, have been applied to solve the OPF problem [8,9]. However, these methods suffer from severe limitations in handling non-linear, discrete–continuous functions, and complex constraints [10]. In order to overcome the limitations of classical optimization techniques, a wide variety of the heuristic methods have been proposed to solve OPF, such as genetic algorithm (GA) [11,12], simulated annealing (SA) [13,14], tabu search (TS) [15], differential evolution (DE) algorithm [16,17], harmony search (HS) algorithm [18], and biogeography based optimization (BBO) [19,20]. The reported results are promising and encouraging for further research in this field [7]. However, all the mentioned mathematical techniques have some drawbacks such as being trapped in local optima and they are only suitable for considering a specific objective function in the OPF problem [21].

Swarm intelligence (SI) is an innovative artificial intelligence technique for solving complex optimization problems. This discipline is inspired by the collective behaviors of social animals such as fish schools, bird flocks, ant colonies, and bee swarms. In recent years, many algorithmic SI methods were designed to deal with practical problems [22–25]. Among them, the artificial bee colony (ABC) algorithm is a powerful search technique that drew inspiration from the biological foraging behaviors observed in bee colony

* Corresponding author. Address: Shenyang Institute of Automation, Chinese Academy of Sciences, Faculty Office VII, Nanta Street 114#, Shenhe District, Shenyang 110016, China.

E-mail address: perfect_chn@hotmail.com (Y. Zhu).

[1] These authors contributed equally to this work.

http://dx.doi.org/10.1016/j.ijepes.2014.02.017
0142-0615/© 2014 Elsevier Ltd. All rights reserved.

[26,27]. ABC is a population-based optimization tool, which has been implemented and applied easily to solve multi-variable single-objective problems. The main strength of ABC is its fast convergence, which compares favorably with evolutionary algorithms (EAs) and other global optimization algorithms [27]. Although ABC is relatively new, its relative simplicity and population-based feature have made it a high competitor in solving the MO problems. Several existing multi-objective ABC (MOABC) algorithms can be found in [28,29]. However, compare to the huge in-depth studies of other *EA* and SI algorithms, such as nondominated sorting genetic algorithm II (NSGAII) [30], strength Pareto evolutionary algorithm (SPEA2) [31], and multi-objective particle swarm optimization (MOPSO) [32], how to improve the diversity of swarm or overcome the local convergence of MOABC is still a challenging to the MO researchers.

The purpose of this paper is to develop a multi-hives bee colony foraging algorithm for solving the multi-objective OPF (MOOPF) problem. This proposed multi-hive multi-objective bee algorithm called M^{2}OBA extends the single population ABC to interacting multi-hive model by combing multi-objective handling strategies and hierarchical interaction topologies. The advantages of the proposed multi-swarm/population model can be listed as: (1) it can improve the population diversity; (2) it can fasten the convergence speed; and (3) it is easy to cooperate in hybrid with another search technique/strategy. By incorporating this new degree of complexity, M^{2}OBA can accommodate a considerable potential for solving more complex MO problem. Here we provide some initial insights into this potential by evaluating M^{2}OBA on a set of mathematical benchmark functions that including four two-objective problems and two three-objective cases, which have been widely employed by other researchers to evaluate their MO algorithms [33–36].

In this paper, the OPF problem is formulated as a multi-objective optimization model which consists of three important terms, namely the total fuel cost, the total emission, and the total real power loss. In addition, a fuzzy-based mechanism is employed to extract the best compromise solution. The 30-bus IEEE test system is presented to illustrate the efficiency of the proposed M^{2}OBA model on MOOPF problem. The simulation results, which are compared to three successful multi-objective optimizers, namely the NSGA-II, MOPSO, and MOABC, are presented to illustrate the effectiveness and robustness of the proposed method.

The rest of the paper is organized as follows. 'Principle of multi-objective optimization' gives relevant MO optimization concepts. In 'Optimal power flow problem formulation', the mathematical formulation of the multi-objective OPF problem is presented. 'Multi-hive multi-objective bee algorithm' gives the detailed description of the proposed M^{2}OBA. In 'Benchmark test', it will be shown that M^{2}OBA outperforms the other three MO optimizers on six widely-used benchmark functions. In 'Multi-objective optimal power flow based on M^{2}OBA', implementation of the M^{2}OBA for the OPF problem is presented. Simulation results and comparison on OPF with other approaches are given in 'Simulation results'. Finally, 'Conclusions' outlines the conclusions.

Principle of multi-objective optimization

A multi-objective optimization problem can be stated as follows: A general multi-objective optimization problem consists of a number of objectives to be optimized simultaneously and is associated with a number of equality and inequality constraints [37]. It can be formulated as follows:

$$\begin{aligned}&\underset{x}{\text{Minimize}}\ \ F(x) = (f_1(x),\ldots,f_m(x))\\&\text{Subject to}:\ \begin{cases} g_i(x) = 0, & i = 1,2,\cdots,p\\ h_j(x) \leqslant 0, & j = 1,2,\cdots,q\end{cases}\end{aligned}\tag{1}$$

where $x = (x_1, \ldots, x_n)$ is called decision (variable) vector, $X \subset R^n$ is the decision (variable) space, R^m is the objective space, F: $X \to R^m$ consists of m ($m \geqslant 2$) real-valued objective functions, and p and q is the number of equality constraints and inequality constraints respectively.

For Eq. (1), let $a = (a_1, \ldots, a_m)$, $b = (b_1,\ldots,b_m) \in R^m$ be two vectors, a is said to dominate b if $a_i \leqslant b_i$ for all $i = 1, \ldots, m$, and $a \neq b$. A point $x^* \in X$ is called Pareto optimal if there is no $x \in X$ such that $F(x)$ dominates $F(x^*)$. Pareto optimal solutions are also called efficient, nondominated, and noninferior solutions. The set of all the Pareto optimal solutions, denoted by *PS*, is called the Pareto set (Fig. 1). The set of all the Pareto objectives vectors, denoted by $PF = \{F(x) \in R^m | x \in PS\}$, is called the Pareto front [38] (Fig. 2).

Due to the inherent conflicting nature among the different objectives to be optimized, it is still a challenging to solve the MO optimization problems (MOPs). Fortunately, caused by the pioneering effort of [39], many well-known evolutionary algorithm (EA) and swarm intelligence (SI) techniques for the MOPs have been proposed with remarkable success [40–42].

Optimal power flow problem formulation

The purpose of the multi-objective OPF (MOOPF) problem is to determine the optimal control variables for minimizing a number of objective functions subject to a set of equality and inequality constraints. The MOOPF problem can be formulated as:

$$\begin{aligned}&\underset{x}{\text{Minimize}}\ f_i(x,u)\ i = 1,\ldots,N_{obj}\\&\text{Subject to}:\ \begin{cases} g(x,u) = 0\\ h(x,u) \leqslant 0\end{cases}\end{aligned}\tag{2}$$

where f_i is the ith objective function, N_{obj} is the number of objective functions, g is a set of constrain equations, and h is a set of formulated constrain in equations. x is the vector of dependent variables such as the slack bus power P_{G1}, the load bus voltage V_L, generator reactive power outputs Q_G, and the apparent power flow S_k. x can be expressed as:

$$x = [P_{G_1}, V_{L_1}, \ldots, V_{L_{PQ}}, Q_{G_1}, \ldots, Q_{G_{N_G}}, S_{L_1}, \ldots, S_{L_{N_L}}]\tag{3}$$

where N_{PQ}, N_G and N_L are the number of load buses, the number of generator buses, and the number of transmission lines, respectively.

Here u is a set of the control variables such as the generator real power output P_G expect at the slack bus P_{G2}, the generator voltages V_G, the transformer tap setting T, and the reactive power generations of var source Q_C. Therefore, u can be expressed as:

$$u = [P_{G_2}, \ldots, P_{G_{N_G}}, V_{G_1}, \ldots, V_{G_{N_G}}, T_1, \ldots, T_{N_T}, Q_{C_1}, \ldots, Q_{C_{N_C}}]\tag{4}$$

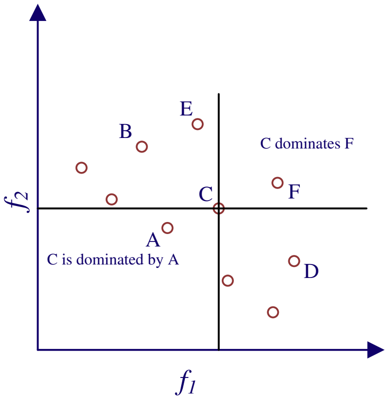


Fig. 1. The possible relations of solutions in objective space.

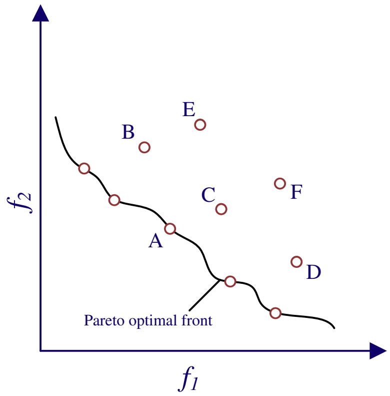


Fig. 2. Illustrative example of Pareto optimality in objective space.

where N_T and N_C are the number of regulating transformers and the number of var compensators, respectively.

In this paper, the MOOPF problem is to minimize three competing objective functions, namely the fuel cost, emission, and real power loss, while satisfying several equality and inequality constraints. Generally the problem is formulated as follows.

Objective functions

Minimization of fuel cost

The generators cost curves are represented by quadratic functions with sine components. The superimposed sine components represent the rippling effects produced by the steam admission valve openings [43]. The total $/h fuel cost of generating units considering the valve loading effects can be modeled as:

$$f_{\text{cost}} = \sum_{i=1}^{N_G} a_i + b_i P_{G_i} + c_i P_{G_i}^2 + \left| d_i \times \sin\left(e_i \times \left(P_{G_i}^{\min} - P_{G_i}\right)\right)\right| \tag{5}$$

where a_i, b_i, c_i, d_i, and e_i are the cost coefficients of the ith generator, and P_{G_i} is the real power output of thermal unit i.

Minimization of emission

The total ton/h emission of the atmospheric pollutants such as sulpher oxides SO_x and nitrogen oxides NO_x caused by fossil-fueled thermal units can be expressed as [44]:

$$f_{emission} = \sum_{i=1}^{N_G} \alpha_i + \beta_i P_{G_i} + \gamma_i P_{G_i}^2 \tag{6}$$

where α_i, β_i, and γ_i are emission coefficients of the ith generating unit.

Minimization of transmission loss

The power flow solution gives all bus voltage magnitudes and angles. Then, the total *MW* active power loss in a transmission network can be described as follows:

$$f_{loss} = \sum_{k=1}^{N_L} g_k \left(V_i^2 + V_j^2 - 2V_i V_j \cos(\delta_i - \delta_j)\right) \tag{7}$$

where g_k is the conductance of kth branch, V_i, V_j, δ_i and δ_j are the voltage magnitudes and phase angles of terminal buses of branch k.

Constraints

Equality constraints

The equality constraints are the nonlinear power flow equations which are formulated as below:

$$0 = P_{G_i} - P_{D_i} - V_i \sum_{j=1}^{N_B} V_j (G_{ij} \cos\theta_{ij} + B_{ij} \sin\theta_{ij}) \tag{8}$$

$$0 = Q_{G_i} - Q_{D_i} - V_i \sum_{j=1}^{N_B} V_j (G_{ij} \sin\theta_{ij} - B_{ij} \cos\theta_{ij}) \tag{9}$$

where P_{G_i} is the injected active power at bus i, P_{D_i} is the demanded active power at bus i, Q_{G_i} is the injected reactive power at bus i, Q_{D_i} is the demanded reactive power at bus i, G_{ij} is the transfer conductance between bus i and j, B_{ij} is the transfer susceptance between bus i and j, θ_{ij} is the voltage angle difference between bus i and j and N_B is the total number of buses.

The mechanism of handling the equality constraint related to the equality of generation level with load level plus loss is that whenever each output of generator is set to its maximum or minimum level the related velocity of the control vector for the next iteration is declined. In this regard, a negative value is added to the current velocity in order to change the direction of aforementioned element that is output power of the generator.

Inequality constraints

These constraints are the set of continuous and discrete constraints that represent the system operational and security limits as follows:

$$\begin{aligned}
&P_{G_i}^{\min} \le P_{G_i} \le P_{G_i}^{\max} \quad i \in N_G \\
&Q_{G_i}^{\min} \le Q_{G_i} \le Q_{G_i}^{\max} \quad i \in N_G \\
&V_{G_i}^{\min} \le V_{G_i} \le V_{G_i}^{\max} \quad i \in N_G \\
&Q_{C_i}^{\min} \le Q_{C_i} \le Q_{C_i}^{\max} i \quad \in N_C \\
&T_i^{\min} \le T_i \le T_i^{\max} \quad i \in N_T \\
&V_{L_i}^{\min} \le V_{L_i} \le V_{L_i}^{\max} \quad i \in N_{PQ} \\
&S_{L_i} \le S_{L_i}^{\max} \quad i \in N_L
\end{aligned} \tag{10}$$

In this work, the penalty factor method is utilized for handling the inequality constraints. In this regard, each control vector which violates constraints will be fined by these penalty factors. Therefore, in the next step, this control vector will be deleted automatically.

Multi-hive multi-objective bee algorithm

The motivation

From Fig. 3 [27], we can understand the basic behavior characteristics of bee colony foraging behaviors better. Assume that there are two discovered food sources: A and B. At the very beginning, a potential bee forager will start as unemployed bee. That bee will have no knowledge about the food sources around the nest.

There are two possible options for such a bee:

i. It can be a scout and starts searching around the nest spontaneously for a food due to some internal motivation or possible external clue (‘*S*’ in Fig. 3).
ii. It can be a recruit after watching the waggle dances and starts searching for a food source (‘*R*’ in Fig. 3).

After finding the food source, the bee utilizes its own capability to memorize the location and then immediately starts exploiting it. Hence, the bee will become an “employed forager”. The foraging bee takes a load of nectar from the source and returns to the hive, unloading the nectar to a food store. After unloading the food, the bee has the following options:

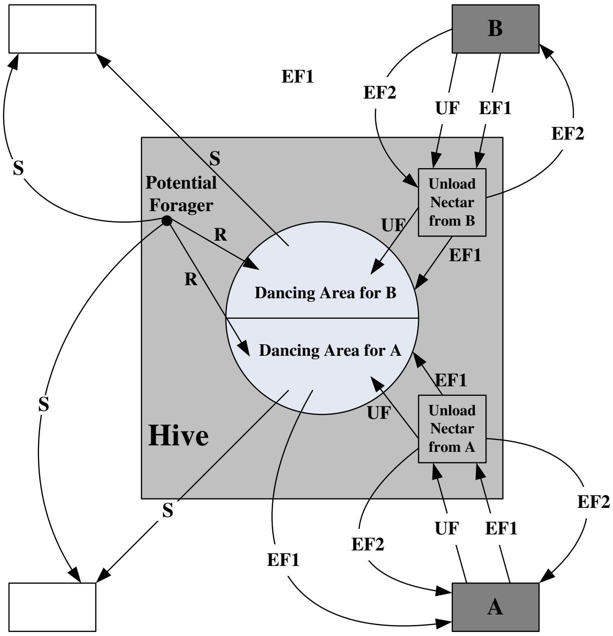


Fig. 3. Behavior of honeybee foraging for nectar.

iii. It might become an uncommitted follower after abandoning the food source (*UF*).
iv. It might dance and then recruit nest mates before returning to the same food source (*EF*1).
v. It might continue to forage at the food source without recruiting after bees (*EF*2).

As described in ‘Introduction’, the main objective of this paper is to develop multi-objective optimization approaches based on above bee colony foraging behaviors that improve the performance without compromising the computational efficiency of the ABC algorithm. Building on the success of preliminary work on bee foraging algorithms and multi-objective techniques, this work aims to demonstrate convincingly that the multi-objective strategies and multi-population cooperative mechanism are both effective strategies and can be utilized to help scaling up the performance of bee foraging algorithms for solving complex MO problems with high dimensionality.

In the proposed model, several multi-objective strategies, namely the external archive, comprehensive learning, greedy selection, and crowding distance strategies are employed to evaluate the fitness of the food source positions and select nondominated solutions. That is, all solutions in the external archive are regarded as food source positions and all bees are regarded as onlooker bees. Different from the original ABC algorithm, the proposed M^{2}OBA no longer exists employed bees and scout bees. At the same time, M^{2}OBA implements a hierarchical interaction topology that consists of two levels (i.e. individual level and colony level), in which information exchanges take place permanently. That is, each artificial bee of the multi-hive model searches the food source based on the information integration of its colony members and its cooperative partners from other hives.

In the initialization phase, the ABC algorithm generates a randomly distributed initial food source positions of *SN* solutions, where *SN* denotes the size of employed bees or onlooker bees. Each solution $x_i(i = 1, 2, \ldots, SN)$ is a *D*-dimensional vector. Here, *D* is the number of optimization parameters. And then evaluate each nectar amount fit_i. In ABC model, nectar amount is the solution value of benchmark function or real-world problem.

The single hive algorithm

The generic steps of the single colony/hive algorithm are shown in Table 1. The detail of all the key steps in Table 1 is elaborated in the following sections.

External archive

As opposed to single-objective optimization, multi-objective *EA* and SI techniques usually maintain a nondominated solutions set. In multi-objective optimization, for the absence of preference information, none of the solutions can be said to be better than the others. Therefore, in the proposed algorithm, we use an external archive to keep a historical record of the nondominated vectors found along the search process [31,45].

In the initialization phase, the bee hive generates a randomly distributed initial food source positions of *SN* solutions, namely the size of onlooker bees. Each solution $x_i(i = 1, 2, \ldots, SN)$ is a *D*-dimensional vector. Here, *D* is the number of optimization parameters. After initializing the solutions and calculating the value of every solution, they are sorted based on nondomination. We compare each solution with every other solution in the population to find which one is nondominated solution. We then put all nondominated solutions into external archive *EA*. The external archive will be updated at each generation.

Comprehensive learning strategy

In order to ensure the diversity of population, the comprehensive learning strategy [46] is embedded in M^{2}OBA algorithm. In our model, all solutions in the external archive are regarded as food source positions and all bees are regarded as onlooker bees. In each generation, each onlooker randomly chooses a food source from external archive, and goes to the food source area, and then chooses a new food source.

For each onlooker bee x_i, it randomly chooses *m* dimensions and learns from a nondominated solution which is randomly selected from *EA*. The new solution is produced by using the following expression:

$$v_{i,f(m)} = x_{i,f(m)} + \phi(m)(EA_{k,f(m)} - x_{i,f(m)}) \tag{11}$$

where $k \in (1, 2, \ldots, p)$ is randomly chosen index, and *p* is the number of solutions in the *EA*. $f(m)$ is the first *m* integers of a random permutation of the integers 1:*n*, and $f(m)$ defines which x_i's dimensions should learn from EA_k. Here $\phi(m)$ produce *m* random numbers which are all between [0, 2]. And the *m* random numbers

Table 1
Pseudocode for single hive algorithm.

```
cycle = 1
Initialize the food source positions (solutions)$x_i$, $i = 1 \ldots SN$
Evaluate the nectar amount (fitness $fiti$) of food sources
The initialized solutions are sorted based on nondomination
Store nondominated solutions in the external archive $EA$
repeat
  Onlooker Bees' Phase
    For each onlooker bee
      Randomly chooses a solution from $EA$
      Produce new solution $vi$
      Calculate the value $fit_i$
      Apply greedy selection mechanism in Table 3 to decide which solution
    enters $EA$
    EndFor
  The solutions in the $EA$ are sorted based on nondomination
  Keep the nondomination solutions of them staying in the $EA$
  If the number of nondominated solutions exceeds the allocated the size of
    $EA$
    Use crowding distance to remove the crowded members
  cycle = cycle + 1.
until cycle = Maximum Cycle Number
```

correspond to the *m* dimensions above. This mechanism makes the potential search space around EA_k. The potential search spaces on one dimension are plotted as a line in Fig. 4.

The remaining each dimension in x_i learns from the other nondominated solutions by:

$$v_{ij} = x_{ij} + \phi_{ij}(EA_{lj} - x_{ij}) \tag{12}$$

where $l \neq k$, $j \in (1, 2, \ldots, p)$ and $j \notin f(m)$. That is, each bee x_i randomly chooses a dimension and learns from a nondominated solution which is randomly selected from *EA*.

Greedy selection mechanism

In our algorithm, each individual will find a new solution in each generation. If the new solution dominates the original individual, then the new solution is allowed to enter the external archive. On the other hand, if the new solution is dominated by the original individual, then it is denied access to the external archive. If the new solution and the original individual do not dominate each other, then we randomly choose one of them to enter the external archive. That is, after producing new solutions in each generation, the greedy selection mechanism is applied to decide which solution enters *EA*. The selection mechanism is shown in Table 2.

Crowding distance

In the course of evolution, more and more new solutions enter the external archive. Considering that each new solution will be compared with every nondominated solution in the external archive to decide whether this new solution should stay in the archive, and the computational time is directly proportional to the number of comparisons, the size of external archive must be limited. In nondominated sorting phase of each generation, the solutions in the *EA* are sorted based on nondomination and the nondomination solutions are maintained in the *EA*. If the number of nondominated solutions exceeds the size of *EA*, the crowding distance is used [30] to remove the crowded members.

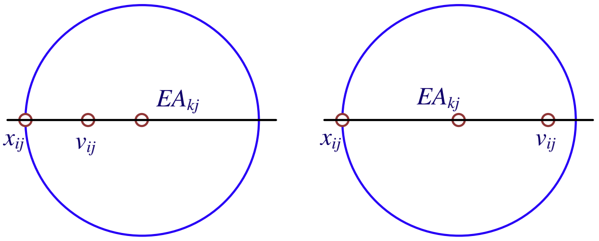


Fig. 4. Possible search regions per dimensions: $0 < \phi_j < 1$ (left); $1 < \phi_j < 2$ (right).

Table 2
Greedy selection mechanism.

```
1: If $v_i$ dominates $x_i$
   Put $v_i$ into EA
2: Else if $x_i$ dominates $v_i$
   Do nothing
3: Else if $x_i$ and $v_i$ are not dominated by each other
   Put vi into EA
   Produce a random number r drawn from a uniform distribution on the unit
    interval
   If $r < 0.5$
      Then the original solution is replaced by the new solution as new food
    source position.
      I.e. $x_i$ is replaced by $v_i$.
   Else
      Do nothing
   EndIf
4: EndIf
```

Crowding distance is used to estimate the density of solutions in the external archive. Usually, the perimeter of the cuboid formed by the vertices (i.e. the nearest neighbors) is called crowding distance. In Fig. 5, the crowding distance of the *i*th solution is the average side length of the cuboid (i.e. the dashed box in Fig. 5).

The crowding distance computation requires sorting the population in the external archive according to each objective function value in ascending order of magnitude. Then for each objective function, the boundary solutions (solutions with smallest and largest function values) are assigned an infinite distance value. All other intermediate solutions are assigned a distance value equal to the absolute normalized difference in the function values of two adjacent solutions. This calculation is continued with other objective functions. The overall crowding-distance value is calculated as the sum of individual distance values corresponding to each objective and each objective function is normalized before calculating the crowding distance.

Multiple hive cooperative mechanism

Original bee foraging algorithms use the analogy of a single-species population and the suitable definition of the artificial bee foraging behaviors reflect the social evolution in the population. However, the situation in nature is much more complex than what this simple metaphor seems to suggest. Indeed, in biological populations there is a continuous interplay between individuals of the same colony (or species), and also interaction and cooperation of various kinds with other colonies (or species) [47].

Recent advances in *EA* and SI show that the introduction of ecological models and cooperative architectures are effective methods to broaden the use of traditional *EA* and SI algorithms [48–52]. In particular, multiple population cooperative techniques provide an effective means of handling large and complex problems via a divide-and-conquer strategy. Hence, this work combines the hierarchical multi-hive interaction topology and an according information transfer strategy into the proposed multi-objective optimizer. With the hierarchical interaction topology, a suitable diversity in the whole population can be maintained. At the same time, the information transfer strategy significantly improves the convergence speed and robustness of the proposed method.

Interaction topology

For the multi-hive cooperative mode of M^2OBA, the whole population is divided into several colonies in different hives, and each colony performs a canonical single hive algorithmic paradigm described in 'The single hive algorithm'. In this multiple colony

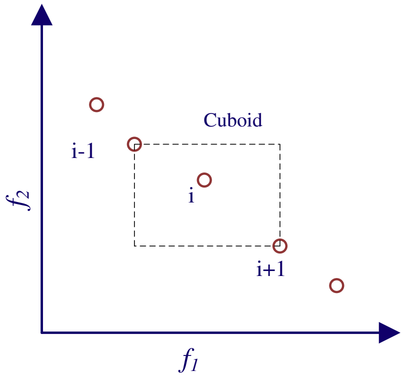


Fig. 5. Crowding-distance. Points are all nondominated solutions in the external archive.

model, the interaction of bees occurred in a two-level hierarchical topology. Fig. 6 illustrates two hierarchical interaction topologies, which employed the ring and star topologies in colony level respectively and both used the star topology in their individual level. It should be noticed that according to the own property of the proposed algorithm, the topology in the individual level of M^{2}OBA is always using the stars.

In Fig. 6(a), different bee hives are arranged in a unidirectional ring. That is, each colony can accept the sending list from an adjacent colony and the individuals of its own replacement list will be replaced with the individuals of the received list. The latter topology in Fig. 6(b) is a complete star graph, where each bee hive broadcasts its sending list to the remaining hives. This complete topology ensures a global synchronization of the best position identified in the multi-hive. The dissemination of this global incumbent is slower in the ring model, since it has to jump gradually from hive to hive.

Obviously, many patterns of connection can be used in different levels of our model. The most common ones are grid, two-dimensional and three-dimensional Von Neumann, small world, and hypercubes [47].

Information transfer strategy

After some predefined generations of optimization, according to the hierarchical interaction topology, each colony will select some individuals with superior information for information transfer. The selected individuals comprise a list and the list will be sent to another colony. On the other hand, each colony prepares a replacement list comprised of individuals which will be replaced by individuals coming from other colonies. A selection strategy is developed based on the comprehensive consideration of location of food source near each beehive, nondomination rank, and crowding-distance instead of their objective values. The constant value for the size of the sending list is predefined, and the sending list will be filled in each colony. The sending list is prepared on the basis of the following rules.

- *Rule*-1: The first consideration in the selection of individuals is nondomination rank. The individuals with the lower rank are preferred.
- *Rule*-2: If the number of individuals in the first rank is greater than the predefined value of the size of the sending list, the average hamming distance and the crowding distance between each pair of individuals in the colony will be calculated. The closest L individuals to the individual that has the largest average hamming distance from others will first enter the sending list and then the individuals with larger crowding distance will be selected. L is a value that depends on the size of the sending list K and the exchange factor δ $(0 < \delta < 1)$:

$$L = \delta \times \frac{K}{3} - 1 \tag{13}$$

- *Rule*-3: If the number of individuals in the first rank is less than the predefined value of the size of the sending list, the individuals in the first rank will first enter the sending list and then the remaining members of the sending list are chosen from subsequent nondominated fronts in the order of their ranking. If the individuals are in the same rank, we will prefer those with larger crowding distance. This procedure is continued until no more individuals can be accommodated in the sending list.

The replacement list that each swarm prepares is based on the nondomination rank and crowding distance in the colony. The replacement list is prepared on the basis of the following rules.

- *Rule*-4: The individuals in the last rank will be replaced first and then the remaining members of the sending replacement are chosen from previous nondominated fronts in the reverse order of their ranking.
- *Rule*-5: If the individuals are in the same rank, the individuals which are located in a lesser crowded region will be replaced first.

It should be noticed that if a hive received several sending lists from its neighbors, these sending lists can be combined in one, and the rules 1–3 should be applied to it to produce a final list for this hive. In this work, we employed the topology in Fig. 6(a). The flowchart of the M^{2}OBA is presented in Fig. 7.

Benchmark test

Test function

According to the No Free Lunch theorem, "for any algorithm, any elevated performance over one class of problems is exactly paid for in performance over another class" [53]. To fully evaluate the performance of the M^{2}OBA algorithm without a biased conclusion towards some chosen problems, we employed four 2-objective and two 3-objective benchmark functions. The formulas of these functions are presented below.

1. **ZDT1**: This is a 30-variables ($n = 30$) problem having a convex Pareto optimal set:

$$ZDT1:\begin{cases} \text{Minimize } f_1(x) = x_1 \\ \text{Minimize } f_2(x) = g(x)\left[1 - \sqrt{x_1/g(x)}\right] \\ \qquad g(x) = 1 + 9\left(\sum_{i=2}^{n} x_i\right) \Big/ (n-1) \end{cases} \tag{14}$$

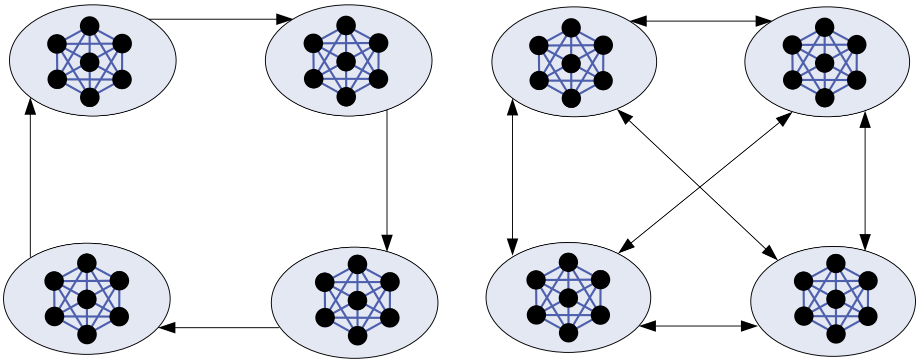

(a)
(b)

Fig. 6. Hierarchical topologies of the multi-hive population.

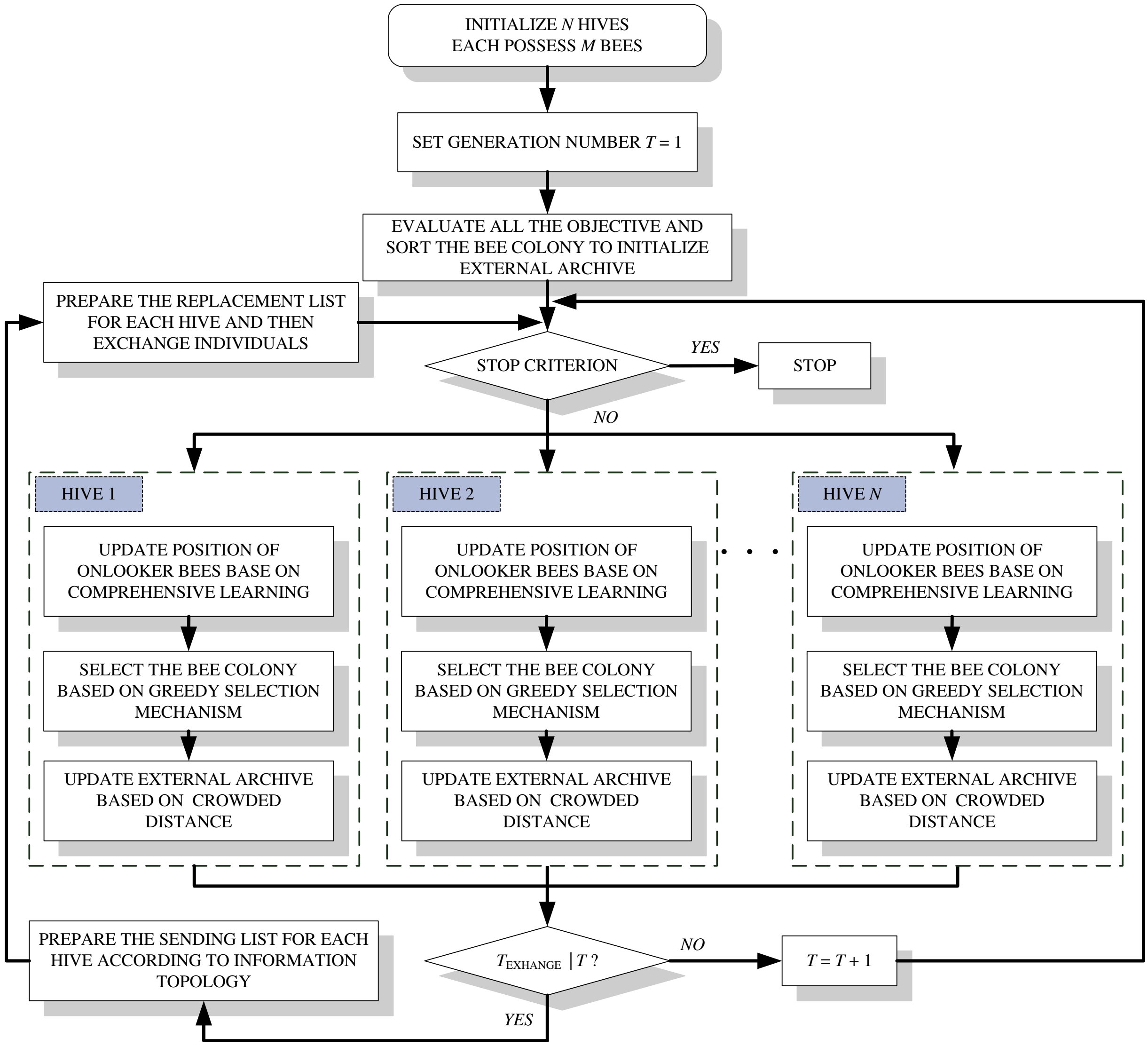

INITIALIZE N HIVES
EACH POSSESS M BEES
SET GENERATION NUMBER T = 1
EVALUATE ALL THE OBJECTIVE AND SORT THE BEE COLONY TO INITIALIZE EXTERNAL ARCHIVE
PREPARE THE REPLACEMENT LIST FOR EACH HIVE AND THEN EXCHANGE INDIVIDUALS
STOP CRITERION
YES
STOP
NO
HIVE 1
HIVE 2
HIVE N
UPDATE POSITION OF ONLOOKER BEES BASE ON COMPREHENSIVE LEARNING
SELECT THE BEE COLONY BASED ON GREEDY SELECTION MECHANISM
UPDATE EXTERNAL ARCHIVE BASED ON CROWDED DISTANCE
PREPARE THE SENDING LIST FOR EACH HIVE ACCORDING TO INFORMATION TOPOLOGY
T_EXHANGE | T ?
NO
T = T + 1
YES

Fig. 7. The flowchart of the M^{2}OBA.

where all variables lie in the range [0, 1]. The Pareto optimal region corresponds to $0 \le x_1^* \le 1$ and $x_1^* = 0$ for i = 2, 3, ..., 30.

2. **ZDT2**: This is also an n = 30 variable problem having a nonconvex Pareto optimal set:

$$ZDT2: \begin{cases} \text{Minimize } f_1(x) = x_1 \\ \text{Minimize } f_2(x) = g(x)[1 - (x_1/g(x))^2] \\ \qquad g(x) = 1 + 9\left(\sum_{i=2}^{n} x_i\right) \Big/ (n-1) \end{cases} \tag{15}$$

where all variables lie in the range [0, 1]. The Pareto optimal region corresponds to $0 \le x_1^* \le 1$ and $x_1^* = 0$ for i = 2, 3, ..., 30.

3. **ZDT3**: This is an n = 30 variable problem having a number of disconnected Pareto optimal fronts:

$$ZDT3: \begin{cases} \text{Minimize} f_1(x) = x_1 \\ \text{Minimize} f_2(x) = g(x)\left[1 - \sqrt{x_1/g(x)} - \frac{x_1}{g(x)}\sin(10\pi x_1)\right] \\ \qquad g(x) = 1 + 9\left(\sum_{i=2}^{n} x_i\right) \Big/ (n-1) \end{cases} \tag{16}$$

where all variables lie in the range [0, 1]. The Pareto optimal region corresponds to $x_1^* = 0$ for i = 2, 3, ..., 30, and hence not all points satisfying $0 \le x_1^* \le 1$ lie on the Pareto optimal front.

4. **ZDT6**: This is a 10-variable problem having a nonconvex Pareto optimal set. Moreover, the density of solutions across the Pareto optimal region is non-uniform and the density towards the Pareto optimal front is also thin:

$$ZDT6: \begin{cases} \text{Minimize } f_1(x) = 1 - \exp(-4x_1)\sin^6(6\pi x_1) \\ \text{Minimize } f_2(x) = g(x)[1 - (f_1(x)/g(x))^2] \\ \qquad g(x) = 1 + 9\left[\left(\sum_{i=2}^{n} x_i\right) \Big/ (n-1)\right]^{0.25} \end{cases} \tag{17}$$

where all variables lie in the range [0, 1]. The Pareto optimal region corresponds to $0 \le x_1^* \le 1$ and $x_1^* = 0$ for i = 2, 3, ..., 10.

5. **DTLZ2**: This test problem has a spherical Pareto-optimal front:

$$DTLZ2: \begin{cases} \text{Minimize} f_1(x) = (1 + g(x_M))\cos(x_1\pi/2)\cdots\cos(x_{M-1}\pi/2) \\ \text{Minimize} f_2(x) = (1 + g(x_M))\cos(x_1\pi/2)\cdots\sin(x_{M-1}\pi/2) \\ \quad \vdots \quad \vdots \\ \text{Minimize} f_M(x) = (1 + g(x_M))\sin(x_1\pi/2) \\ \text{Subject to } 0 \leqslant x_i \leqslant 1, \quad \text{for } i = 1, \ldots, n \\ \text{Where } g(x_M) = \sum_{x_i \in x_M} (x_i - 0.5)^2 \end{cases} \tag{18}$$

where the Pareto-optimal solutions corresponds to $x_i^* = 0.5$ ($x_i^* \in x_M$) and all objective function values must satisfy the $\sum_{m=1}^{M}(f_m^*)^2 = 1$. As in the previous problem, it is recommended to use $k = |x_M| = 10$. The total number of variables is $n = M + k - 1$.

6. **DTLZ6**: This test problem has $2^M{-}1$ disconnected Pareto-optimal regions in the search space:

$$DTLZ6: \begin{cases} \text{Minimize } f_1(x) = x_1 \\ \text{Minimize } f_2(x) = x_2 \\ \quad\vdots \qquad \vdots \\ \text{Minimize } f_{M-1}(x) = x_{M-1} \\ \text{Minimize } f_M(x) = (1 + g(x_M))h(f_1, f_2, \ldots, f_{M-1}, g) \\ \text{Subject to } 0 \leqslant x_i \leqslant 1, \quad \text{for } i = 1, \ldots, n \\ \text{Where } g(x_M) = 1 + \frac{g}{|x_M|}\sum_{x_i \in x_M} x_i \\ \qquad h = M - \sum_{i=1}^{M-1}\left[\frac{f_i}{1+g}(1 + \sin(3\pi f_i))\right] \end{cases} \tag{19}$$

where the functional g requires $k = |x_M|$ decision variables and the total number of variables is $n = M + k - 1$.

Performance measures

In order to facilitate the quantitative assessment of the performance of a multi-objective optimization algorithm, two performance metrics are taken into consideration: (1) convergence metric Υ; (2) diversity metric Δ [30].

Convergence metric

This metric measure the extent of convergence to a known set of Pareto optimal solutions:

$$\Upsilon = \frac{\sum_{i=1}^{N} d_i}{N} \tag{20}$$

where N is the number of nondominated solutions obtained with an algorithm and d_i is the Euclidean distance between each of the nondominated solutions and the nearest member of the true Pareto optimal front. To calculate this metric, we find a set of $H = 10{,}000$ uniformly spaced solutions from the true Pareto optimal front in the objective space. For each solution obtained with an algorithm, we compute the minimum Euclidean distance of it from H chosen solutions on the Pareto optimal front. The average of these distances is used as the convergence metric Υ. Fig. 8 shows the calculation procedure of this metric.

Diversity metric

This metric measure the extent of spread achieved among the obtained solutions. Here, we are interested in getting a set of solutions that spans the entire Pareto optimal region. This metric is defined as:

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{N-1}|d_i - \bar{d}|}{d_f + d_l + (N-1)\bar{d}} \tag{21}$$

where d_i is the Euclidean distance between consecutive solutions in the obtained nondominated set of solutions and N is the number of nondominated solutions obtained by an algorithm. $\bar{d}$ is the average value of these distances. d_f and d_l are the Euclidean distances between the extreme solutions and the boundary solutions of the obtained nondominated set, as depicted in Fig. 9.

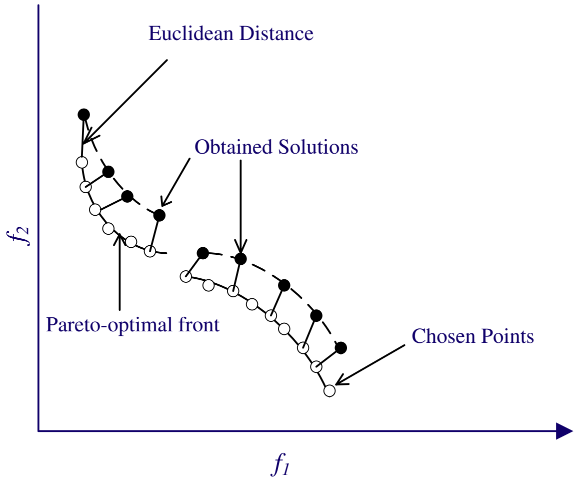


Fig. 8. Convergence metric Υ.

Experimental setting

To fully evaluate the performance of the proposed M^{2}OBA algorithm, three successful nature-inspired multi-objective optimization algorithms were used for comparison:

- the nondominated sorting genetic algorithm II (NSGA-II) [30];
- the multi-objective particle swarm optimization (MOPSO) [32];
- the multi-objective artificial bee colony algorithm (MOABC) [29].

In all experiments in this section, in order to compare the different algorithms with a fair time measure, the number of function evaluations (FEs) is used for the termination criterion. The other specific parameters of algorithms are given below:

NSGA-II settings

The original NSGA-II algorithm uses Simulated Binary Crossover (SBX) and Polynomial crossover. We use a population size of 100. Crossover probability $pc = 0.9$ and mutation probability is $pm = 1/n$, where n is the number of decision variables. The distribution indices for crossover and mutation operators as η_c μ = 20 and η_m μm = 20 respectively.

MOPSO settings

MOPSO used a population of 100 particles, a repository size of 100 particles, a mutation rate of 0.5, and 30 divisions for the

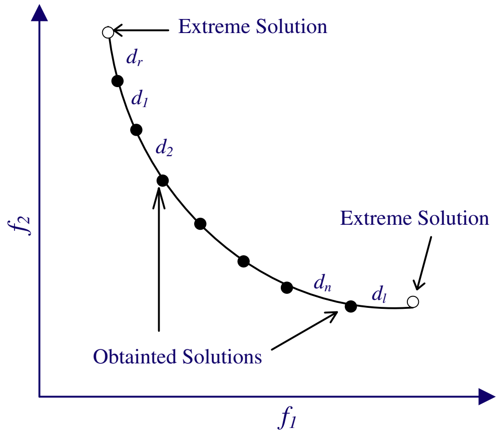


Fig. 9. Diversity metric Δ.

adaptive grid. The detailed implementation and parameters setting for this MOPSO version can be refer to [32].

MOABC settings

For the MOABC proposed in [29], a colony size of 50, archive size $A = 100$ was adopted.

M^2OBA settings

For the proposed M^2OBA, the number of hives N is set at 4, the colony size and archive size of each hive is $M = 20$ and $A = 40$, respectively.

Two objective functions

The experimental results found in 30 runs on each two objective benchmark, including the best, worst, average, median and standard deviation of the convergence metric and diversity metric values are proposed in Tables 3–6, respectively. Here all algorithms are terminated after 10,000 function evaluations. Figs. 10–13 show the optimal front obtained by three algorithms for each two objective function, respectively. The continuous lines represent the Pareto optimal front, while mark spots represent found nondominated solutions by the algorithms.

On ZDT1 function, when given 10,000 function evaluations for four algorithms, Table 3 shows that the performance of MOPSO in convergence metric is one order of magnitude better than that of NSGA-II and MOABC, but it is one order of magnitude worse than that of M^2OBA. For diversity metric, it can be observed that all algorithms have almost the same performance on ZDT1. Fig 10 shows that M^2OBA, MOPSO and MOABC can discover a well-distributed and diverse solution set for this problem. However, NSGA-II only finds a sparse distribution, and it cannot archive the true Pareto front for ZDT1.

On ZDT2 function, the results of the performance measures show that M^2OBA and MOABC have better convergence and diversity compared to the MOPSO and NSGA-II. From Table 4, it can be noticed that the performance of M^2OBA and MOABC in convergence metric and diversity metric are three orders of magnitude better than that of MOPSO and NSGA-II after 10,000 function evaluations. We can see from Table 4 that M^2OBA outperform MOABC one order of magnitude in terms of diversity metric. Fig. 11 shows that MOPSO and NSGA-II produces poor results on this test function and both of them cannot achieve the true Pareto front.

On ZDT3and ZDT6 functions, we can observe that the algorithms achieve similar performance ranking as on ZDT1and ZDT2.

Three objective functions

Tables 7 and 8 show the optimization results of M^2OBA, MOPSO, MOABC and NSGA-II algorithms for three objective DTLZ series problems. Figs. 14–17 show the true Pareto optimal front and the optimal front obtained by four algorithms for DTLZ2 and DTLZ6.

On DTLZ2 function, we can observe from Table 7 that the performances of all the algorithms in convergence metric have been competitively good over this problem. However, we can see that the performance of NSGA-II is one order of magnitude worse than that of the other three algorithms in convergence metric, while the performance of M^2OBA is one order of magnitude better than that of the other three algorithms in diversity metric. From Fig. 15, it can be seen that the front obtained from M^2OBA is found to be uniformly distributed on the true Pareto optimal front of DTLZ2.

DTLZ6 problem has $2^M - 1$ disconnected Pareto-optimal regions in the search space. This problem will test an algorithm's ability to maintain subpopulation in different Pareto-optimal regions. From Table 8, we can observe that the performance of M^2OBA is better than all the other algorithms in both convergence and diversity metrics. From Fig 17, we can observe that the M^2OBA is able to discover a well-distributed and diverse solution set for this problem. However, the NSGA-II cannot archive the true Pareto front for DTLZ6.

Multi-objective optimal power flow based on M^2OBA

In this section, the details of proposed approach to solve the multi-objective OPF problem are described.

Implementation of the M^2OBA algorithm for the OPF problem

To apply the multi-objective algorithms to solve the OPF problem, the following steps should be taken and repeated.

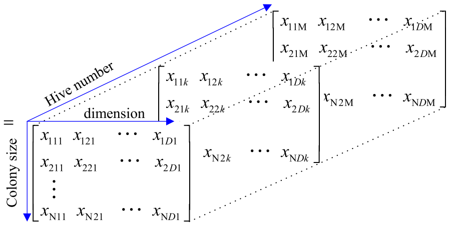
- *Step 1*: Input the parameters of power system, parameters of the M^2OBA algorithm, and lower and upper limits of each variable.
- *Step 2*: Transfer the constraint multi-objective problem to an unconstraint one as follows:

$$F(x) = \begin{bmatrix} F_1(x) \\ F_2(x) \\ F_3(x) \end{bmatrix} = \begin{bmatrix} f_{cost}(x) + Z_1\left(\sum_{j=1}^{N_{eq}} (g_i(x))^2\right) + Z_2\left(\sum_{j=1}^{N_{ueq}} (\text{Max}[0, -h_i(x)])^2\right) \\ f_{emission}(x) + Z_1\left(\sum_{j=1}^{N_{eq}} (g_i(x))^2\right) + Z_2\left(\sum_{j=1}^{N_{ueq}} (\text{Max}[0, -h_i(x)])^2\right) \\ f_{loss}(x) + Z_1\left(\sum_{j=1}^{N_{eq}} (g_i(x))^2\right) + Z_2\left(\sum_{j=1}^{N_{ueq}} (\text{Max}[0, -h_i(x)])^2\right) \end{bmatrix} \quad (22)$$

where N_{eq} and N_{ueq} are the number of equality and inequality constraints, respectively. $g_j(x)$ and $h_j(x)$ are the equality and inequality constraints, respectively, and Z_1 and Z_2 are the penalty factors.

- *Step 3*: Produce the initial M^2OBA population. $N \times M (N \geqslant 2, M \geqslant 2)$ individuals based on state variables should be randomly generated as follows:

Population =


Hive number
dimension
Colony size
x_{111} x_{121} ⋯ x_{1D1}
x_{211} x_{221} ⋯ x_{2D1}
⋮
x_{N11} x_{N21} ⋯ x_{ND1}
x_{11k} x_{12k} ⋯ x_{1Dk}
x_{21k} x_{22k} ⋯ x_{2Dk}
x_{N2k} ⋯ x_{NDk}
x_{11M} x_{12M} ⋯ x_{1DM}
x_{21M} x_{22M} ⋯ x_{2DM}
x_{N2M} ⋯ x_{NDM}

(23)

where x_{ijk} ($i = 1, 2, \ldots, M, j = 1, 2, \ldots, D, k = 1, 2, \ldots, N$) is the position of the jth state variable in the ith individual of the kth colony, N, D, and M are the number of initialized hives, control variables and bees in each hive, respectively.

- *Step 4*: Calculate the objective functions value for each bee in each hive, sort them based on nondomination, and store nondominated solutions in the external archive *EA* of each hive.
- *Step 5*: Update the position of each bee in each hive according to comprehensive learning mechanism. If any element of each bee breaks its limit then its position is fixed to the maximum minimum operating point.
- *Step 6*: Update each *EA* of each hive according to greedy selecting strategy, sort the *EA* based on nondomination, and select the nondomination solutions to stay in *EA*. If the number of nondominated solutions exceeds the allocated the size of *EA*, apply crowding distance to remove the crowded members.

Table 3
Comparison of performance on ZDT1.

ZDT1	M^{2}OBA	MOABC	NSGA-II	MOPSO
Converge metric				
Average	8.5264e−004	2.3149e−002	2.2278e−001	1.534e−003
Median	8.3455e−004	2.2455e−002	1.4410e−001	1.4512e−003
Best	7.5656e−004	1.8867e−002	7.2072e−002	8.4802e−004
Worst	1.0510e−003	2.7309e−002	8.7348e−001	2.2124e−003
Std	8.0445e−005	2.6057e−003	2.3806e−001	6.6747e−004
Diversity metric				
Average	6.1502e−001	3.3061e−001	5.9362e−001	6.9775e−001
Median	6.0783e−001	3.3108e−001	5.2694e−001	6.8873e−001
Best	5.3456e−001	2.8898e−001	4.5810e−001	6.7851e−001
Worst	6.7678e−001	3.7446e−001	9.1451e−001	7.2602e−001
Std	4.4416e−002	2.4324e−002	1.5649e−001	2.5062e−002

Table 4
Comparison of performance on ZDT2.

ZDT2	M^{2}OBA	MOABC	NSGA-II	MOPSO
Converge metric				
Average	8.2512e−004	1.0023e−003	2.9476e−001	1.620e−001
Median	8.3727e−004	9.5537e−004	1.8105e−001	8.5731e−002
Best	6.2466e−004	7.2183e−004	1.0668e−001	6.5277e−004
Worst	4.7223e−003	1.5690e−003	9.8218e−001	6.2114e−001
Std	1.4789e−001	2.5380e−004	2.6288e−001	6.6671e−001
Diversity metric				
Average	6.7809e−002	2.9352e−001	7.6302e−001	6.3802e−001
Median	6.7379e−002	2.9185e−001	7.2999e−001	6.3801e−001
Best	6.1118e−002	2.5003e−001	4.6617e−001	5.2805e−001
Worst	7.5581e−002	3.3111e−001	1.0542e+000	7.2302e−001
Std	1.4541e−002	2.2147e−002	2.3567e−001	7.7498e−002

Table 5
Comparison of performance on ZDT3.

ZDT3	M^{2}OBA	MOABC	NSGA-II	MOPSO
Converge metric				
Average	4.0779e−003	1.5551e−003	3.3642e+000	1.4321e−002
Median	4.0750e−003	1.5186e−003	3.9688e+000	3.5213e−003
Best	3.5541e−003	1.2901e−003	6.9186e−001	2.6342e−003
Worst	4.7440e−003	2.1061e−003	4.7179e+000	5.8521e−002
Std	3.2377e−004	2.2384e−004	1.4822e+000	2.4734e−002
Diversity metric				
Average	6.3788e−002	6.6057e−001	1.1219e+000	6.960e−001
Median	6.4409e−002	6.6325e−001	1.1187e+000	7.0444e−001
Best	5.5876e−002	6.3194e−001	1.0231e+000	5.0995e−001
Worst	6.7171e−002	6.8535e−001	1.2849e+000	8.7371e−001
Std	1.3904e−002	1.8462e−002	8.8914e−002	1.4262e−001

Table 6
Comparison of performance on ZDT6.

ZDT6	M^{2}OBA	MOABC	NSGA-II	MOPSO
Converge metric				
Average	5.0334e−004	3.9988e−003	3.3642e+000	5.052e−001
Median	1.0989e−005	2.8035e−005	3.9688e+000	8.2662e−004
Best	9.6425e−005	1.9642e−005	6.9186e−001	6.5963e−004
Worst	1.7650e−003	2.6604e−002	4.7179e+000	2.5230
Std	7.4049e−001	8.9540e−003	1.4822e+000	1.1280
Diversity metric				
Average	7.9631e−002	6.0121e−001	1.1219e+000	6.637e−001
Median	7.6843e−002	4.7318e−001	1.1187e+000	6.751e−001
Best	6.0809e−002	4.3742e−001	1.0231e+000	5.974e−001
Worst	1.1885e−001	1.2171e+000	1.2849e+000	7.174e−001
Std	1.9418e−002	2.7989e−001	8.8914e−002	4.55e−002

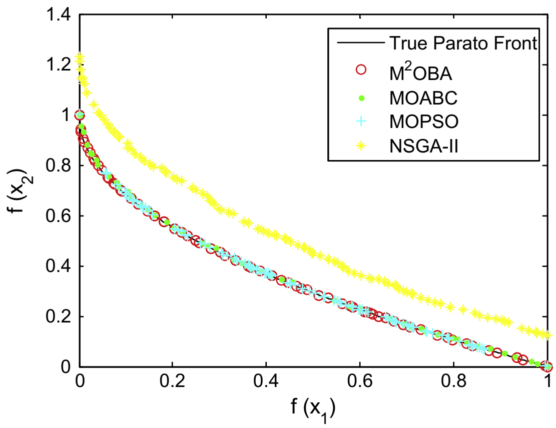


Fig. 10. Pareto fronts obtained by M^{2}OBA, MOPSO, MOABC, and NSGA-II on ZDT1.

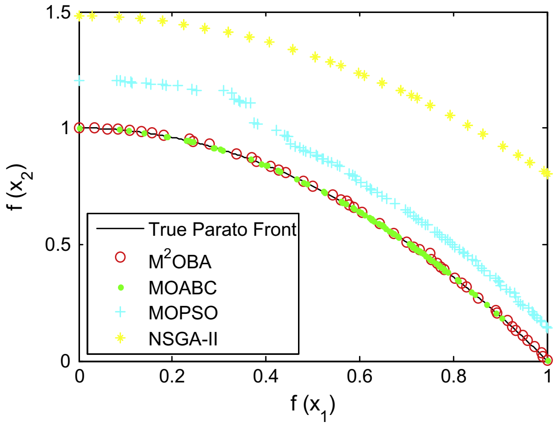


Fig. 11. Pareto fronts obtained by M^{2}OBA, MOPSO, MOABC, and NSGA-II on ZDT2.

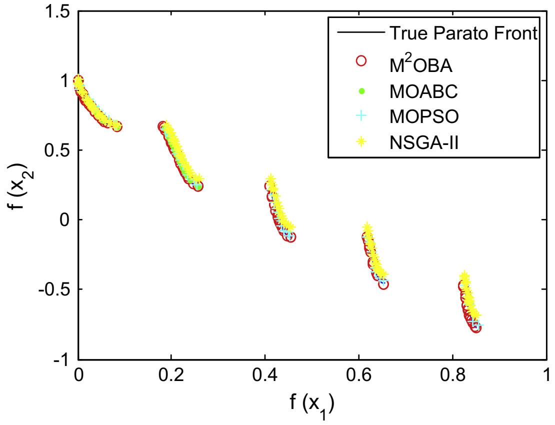


Fig. 12. Pareto fronts obtained by M^{2}OBA, MOPSO, MOABC, and NSGA-II on ZDT3.

- *Step 7*: If the current iteration number obtains the preordained maximum iteration number, the algorithm is stopped, otherwise go to step 4.

Best compromise solution based on fuzzy decision

Upon having the Pareto-optimal set of nondominated solution, the proposed approach presents one solution to the decision maker

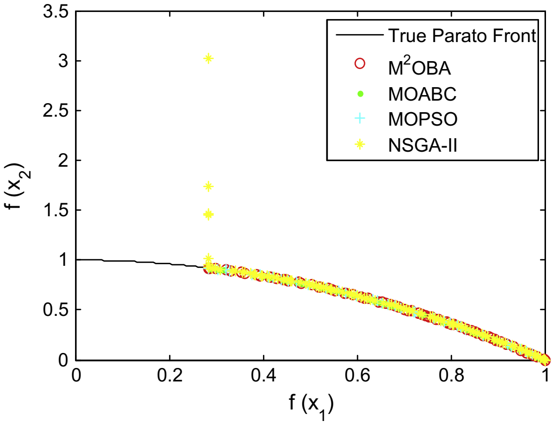


Fig. 13. Pareto fronts obtained by M^{2}OBA, MOPSO, MOABC, and NSGA-II on ZDT6.

Table 7
Comparison of performance on DTLZ2.

DTLZ2	M^{2}OBA	MOABC	NSGA-II	MOPSO
Converge metric				
Average	2.8571e−003	7.6836e−003	8.7363e−002	3.89231e−003
Median	2.8582e−003	6.7337e−003	8.9419e−002	3.81261e−003
Best	2.5627e−003	4.9233e−003	6.0319e−002	3.71302e−003
Worst	3.1086e−003	1.2708e−002	1.1310e−001	3.9421e−003
Std	1.5407e−004	2.4500e−003	1.6418e−002	8.1354e−005
Diversity metric				
Average	4.3011e−002	4.1177e−001	6.0903e−001	4.1162e−001
Median	4.2726e−002	4.1107e−001	5.0881e−001	4.0565e−001
Best	3.9502e−002	3.5942e−001	3.7879e−001	3.9533e−001
Worst	4.6634e−002	4.7267e−001	7.3358e−001	4.4691e−001
Std	2.2715e−003	3.1091e−002	1.8498e−002	2.0232e−002

Table 8
Comparison of performance on DTLZ6.

DDTLZ6	M^{2}OBA	MOABC	NSGA-II	MOPSO
Converge metric				
Average	1.5498e−002	2.9901e−002	6.6160e−001	1.6426e−002
Median	1.5033e−002	2.7952e−002	2.8640e−001	1.7221e−002
Best	1.2131e−002	1.9106e−002	1.6724e−001	8.3231e−002
Worst	2.0374e−002	3.7997e−002	3.1497e+000	2.9165e−002
Std	2.6337e−003	6.7314e−003	9.2366e−001	8.3412e−003
Diversity metric				
Average	5.2003e−002	5.4715e−001	5.7305e−001	5.4362e−001
Median	5.1893e−002	5.5144e−001	5.5839e−001	5.4445e−001
Best	4.6557e−002	5.0172e−001	4.5813e−001	5.184e−001
Worst	5.9150e−002	6.0581e−001	7.0776e−001	5.6638e−001
Std	3.7954e−003	2.9597e−002	7.2624e−002	1.8843e−002

in power system as the best compromise solution. In this work, a fuzzy-based mechanism is employed to extract the best compromise solution over the trade-off curve and assist the decision maker to adjust the generation levels efficiently [54]. Due to imprecise nature of the decision maker's judgment, each objective function of the *i*th solution is represented by a membership function μ_i defined as flow:

$$\mu_i = \begin{cases} 1 & F_i \le \min(F_i) \\ \frac{\max(F_i)-F_i}{\max(F_i)-\min(F_i)} & \min(F_i) \leqslant F_i \le \max(F_i), \\ 0 & F_i \ge \max(F_i) \end{cases} \tag{24}$$

where $\min(F_i)$ and $\max(F_i)$ are lower and upper bounds of *i*th objective function. The higher the values of the membership function are, the greater the solution satisfaction is.

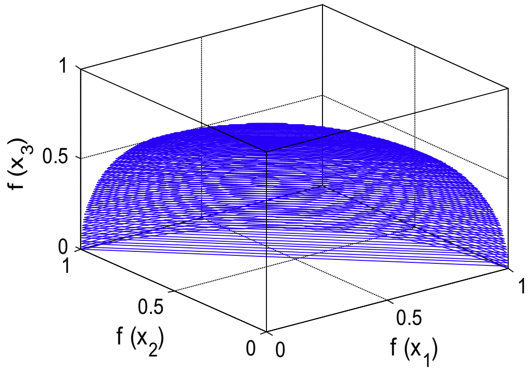


Fig. 14. The true Pareto-optimal front on DTLZ2.

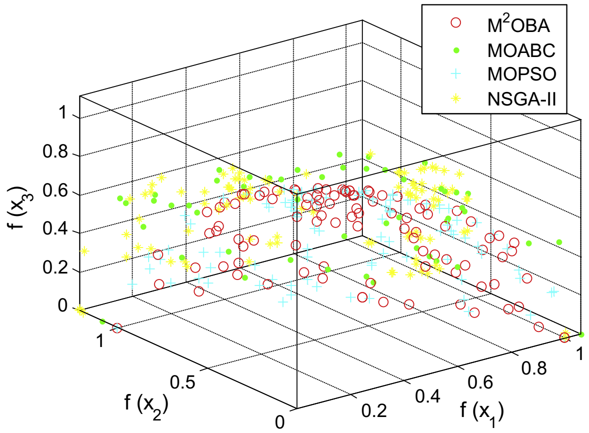


Fig. 15. Pareto fronts obtained by M^{2}OBA, MOPSO, MOABC, and NSGA-II on DTLZ2.

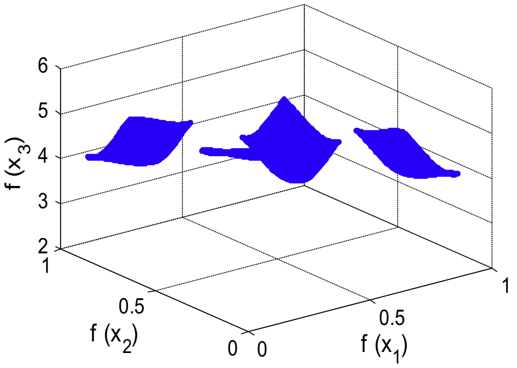


Fig. 16. The true Pareto-optimal front on DTLZ6.

For each nondominated solution, the normalized membership function μ^k is calculated as:

$$\mu^k = \frac{\sum_{i=1}^{N_{obj}} \mu_i^k}{\sum_{k=1}^{M}\sum_{i=1}^{N_{obj}} \mu_i^k}, \tag{25}$$

where M is the number of nondominated solutions, and N_{obj} is the number of object. The best compromise solution is the one having the maximum of μ^k.

Simulation results

In order to verify the proposed approach, the IEEE 30-bus system that illustrated in Fig. 18 is used as the test systems for

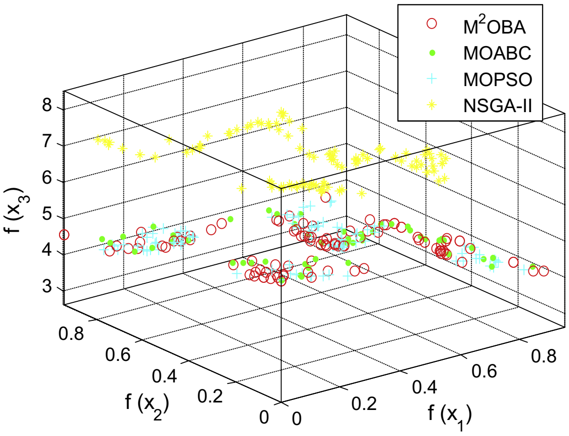


Fig. 17. Pareto fronts obtained by M^{2}OBA, MOPSO, MOABC, and NSGA-II on DTLZ6.

M^{2}OBA, MOPSO, MOABC, and NSGA-II. The IEEE 30-bus test case represents a portion of the American Electric Power System (in the Midwestern US) as of December, 1961. IEEE is the acronym for the Institute of Electrical and Electronics Engineers. The IEEE 30-bus system includes six generators, 41 transmission lines, 4 transformers with off-nominal tap ratio in the lines 6–9, 6–10, 4–12 and 27–28. The system data are given in [55]. The active power generation limits are listed in Table 9. The limits of generator buses and load buses are between 0.95–1.1 p.u, and 0.9–1.05 p.u, respectively. The lower and upper limits of transformer taps are 0.9 p.u. and 1.05 p.u., respectively, and the step size is 0.01 p.u. The parameter settings for these four algorithms are the same as in ‘Performance measures’.

Case I: two-objective OPF optimization

In this case, the OPF model is handled as a multi-objective optimization problem, where each two objective functions are optimized simultaneously. According to the cost–emission, loss–cost, loss–emission pairs, all obtained Pareto fronts by the M^{2}OBA, MOPSO, MOABC, and NSGA-II algorithms are shown in Figs. 19–21, respectively. Tables 10–15 show the Pareto-optimal solutions and best compromise solutions for each objective in the two-dimensional Pareto front.

Table 9
Characteristics of the generation units.

	G1	G2	G3	G4	G5	G6
Generator limits						
P_{Cmax} (MW)	150	150	150	150	150	150
P_{Cmin} (MW)	5	5	5	5	5	5
Cost coefficients						
a	10	10	20	10	20	10
b	200	150	180	100	180	150
c	100	120	40	60	40	100
Emission coefficients						
α	4.091	2.543	4.258	5.326	4.258	6.131
β	−5.554	−6.047	−5.094	−3.550	−5.094	−5.555
γ	6.490	5.638	4.586	3.380	4.586	5.151
ξ	2.0e−4	5.0e−4	1.0e−6	2.0e−3	1.0e−6	1.0e−5
λ	2.857	3.333	8.000	2.000	8.000	6.667

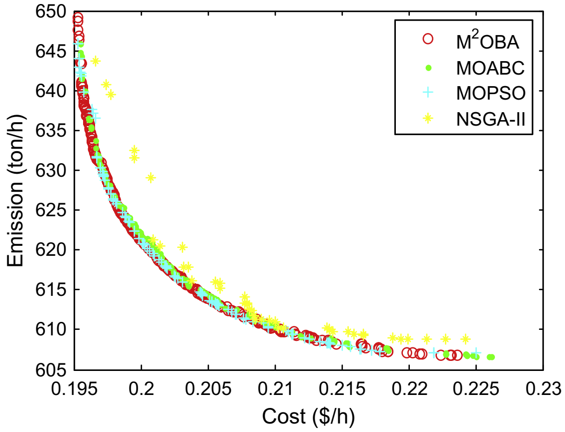


Fig. 19. Pareto fronts obtained by M^{2}OBA, MOPSO, MOABC, and NSGA-II for cost and emission (f_1–f_2).

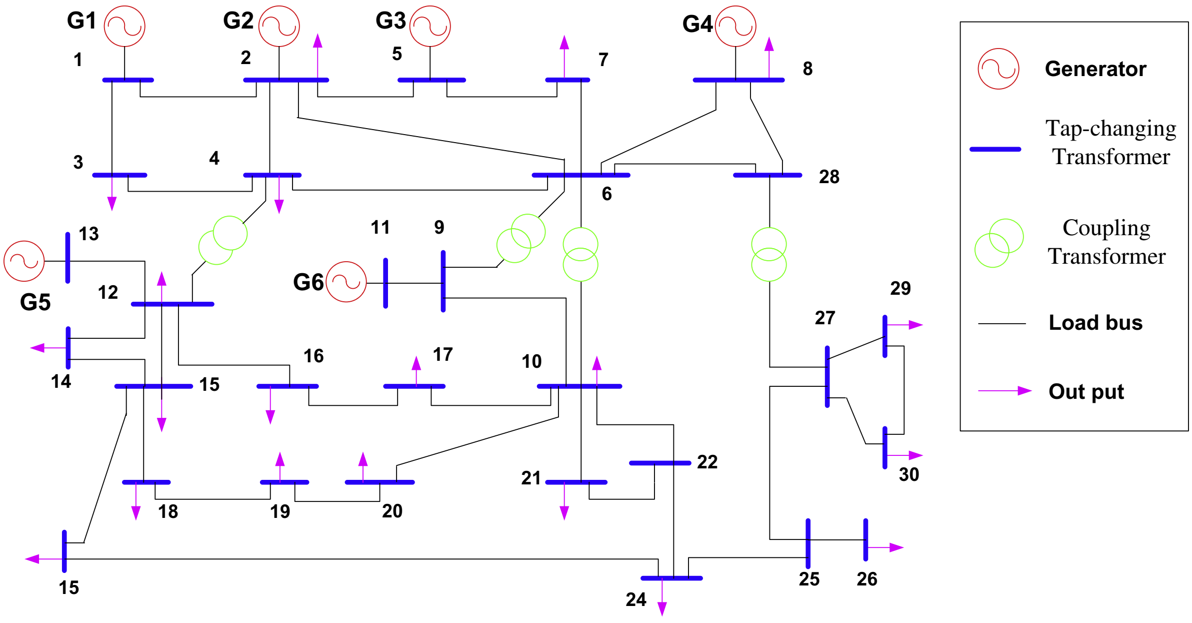


Fig. 18. IEEE 30-bus system.

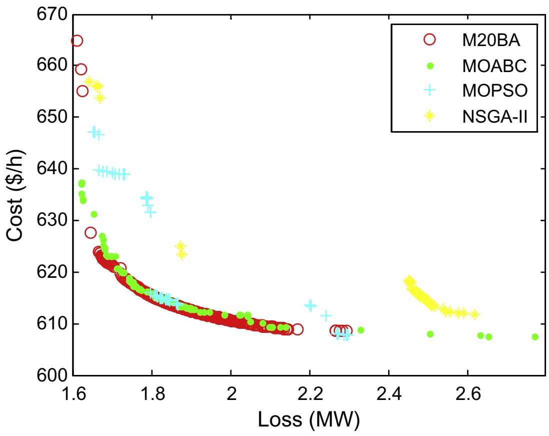


Fig. 20. Pareto fronts obtained by M^2OBA, MOPSO, MOABC, and NSGA-II for cost and loss (f_1–f_3).

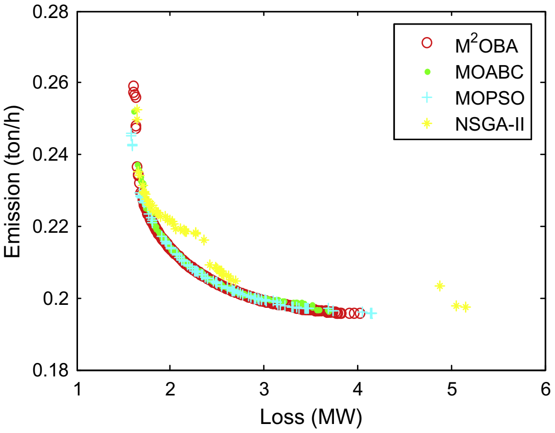


Fig. 21. Pareto fronts obtained by M^2OBA, MOPSO, MOABC, and NSGA-II for emission and loss (f_2–f_3).

For all the results on three two-objective OPF problems, it is clear that the proposed M^2OBA algorithm is able to obtain well-distributed Pareto-optimal fronts. It should be noted that in the proposed approach the trade-off among the competing objectives is obtained by emphasizing on non-dominated solutions and getting a well-distributed set of solutions, respectively.

From Table 10, we can see that M^2OBA get the best convergence solutions for fuel cost and emission objective functions. Fig. 19 shows the Pareto-optimal front for fuel cost and emission objective functions. From Fig. 19, it can be clearly observed that M^2OBA can discover a well-distributed and diverse solution set for this problem, while the other three algorithms only find some sparse distributions. Table 11 shows the best compromise solutions for fuel cost and emission objective functions using different algorithms. For M^2OBA, the best compromise solution is 616.86 $/h and 0.202 ton/h. The fuel cost and emission values in the best compromise solution that are very close to their best values in Table 10, then the truth of the aforementioned statement is clear in all Pareto fronts.

Table 11
The best compromise solutions for cost and emission (f_1–f_2) using different multi-objective algorithms.

	M^2OBA	MOABC	NSGA-II	MOPSO
PG1	24.4837	25.1875	36.1782	24.2855
PG2	39.0558	39.4798	55.3158	37.61433
PG3	56.9255	38.5551	49.5340	55.3510
PG4	71.1890	74.1921	46.0786	72.0443
PG5	50.5360	48.0866	56.8116	51.5658
PG6	44.6095	43.0481	44.1181	44.5503
f_1 Fuel cost ($/h)	616.8621	615.5057	643.7436	616.6645
f_2 Emission (ton/h)	0.2023	0.2035	0.1965	0.2024

For cost–loss objective functions and emission–loss objective functions, we can observe that the algorithms achieve similar performance ranking as for fuel cost–emission objective functions.

Case II: three-objective OPF optimization

In this case, three competing objectives are optimized simultaneously by the proposed algorithm and the obtained results are shown in Fig. 22. Table 16 shows the minimum values for each objective in the three-dimensional Pareto front.

It is clear that cost, emission and loss cannot be further improved without degrading the other two related optimized objectives. Fig. 22 clearly shows the relationships among all presented objective functions. Between the obtained Pareto-optimal solutions, it is necessary to choose one of them as a best compromise for implementation. As well as in the case I, the best compromise results are also presented in Table 17.

From Tables 16 and 17, we can observe that the performance of M^2OBA is better than all the other algorithms in both best and compromise solutions on fuel cost, emission and loss objectives. From Fig. 22, we can observe that the M^2OBA is able to discover a well-distributed and diverse solution set for this three-objective problem. However, the other three algorithms cannot archive the true Pareto front for the three-objective OPF.

From the results, it can once again be proved that the proposed method is giving well-distributed Pareto-optimal front for the three-objective OPF optimization. The results confirm that the M^2OBA algorithm is an impressive tool for solving the real-world multi-objective OPF problem where multiple Pareto-optimal solutions can be obtained in a single run.

Table 10
The best solutions for cost and emission (f_1–f_2) from the Pareto front based on four multi-objective algorithms.

	M^2OBA		MOABC		NSGA-II		MOPSO	
	Best f_1	Best f_2	Best f_1	Best f_2	Best f_1	Best f_2	Best f_1	Best f_2
PG1	12.33	41.24	10.34	6.71	36.17	41.08	26.06	55.72
PG2	30.08	45.39	30.14	41.46	55.31	48.60	33.32	46.15
PG3	55.43	55.21	57.45	61.66	49.53	51.19	61.58	55.92
PG4	105.20	41.36	105.49	100.81	46.07	43.09	101.88	40.22
PG5	45.26	52.81	45.36	41.51	56.81	51.95	45.17	55.89
PG6	37.86	51.59	37.35	33.95	44.11	51.77	37.84	49.87
f_1 (Fuel cost ($/h)	606.65	645.79	606.67	608.80	643.74	645.93	607.08	645.88
f_2 Emission (ton/h)	0.2250	0.1902	0.2261	0.2241	0.1965	0.1953	0.2249	0.1952

Table 12
The best solutions for cost and loss (f_1–f_3) from the Pareto front based on four multi-objective algorithms.

	M^2OBA		MOABC		NSGA-II		MOPSO	
	Best f_1	Best f_3	Best f_1	Best f_1	Best f_3	Best f_3	Best f_1	Best f_3
PG1	15.00	2.26	9.77	18.36	3.91	2.19	67.12	8.88
PG2	29.31	26.34	33.91	20.81	10.63	23.71	38.31	9.20
PG3	56.26	106.00	60.90	40.30	110.74	73.77	68.44	108.80
PG4	112.59	56.77	95.39	119.33	67.45	98.03	101.65	70.32
PG5	30.21	5.00	45.29	40.49	10.96	5.00	37.02	72.36
PG6	42.29	88.61	40.88	46.70	81.33	82.29	28.28	33.63
f_1 (Fuel cost ($/h)	608.52	664.86	607.4267	611.67	656.80	637.30	609.33	660.21
f_3 Loss (MW)	2.2924	1.6099	2.7710	2.6197	1.6407	1.6186	2.6137	2.2421

Table 13
The best compromise solutions for cost and loss (f_1–f_3) using different multi-objective algorithms.

	M^2OBA	MOABC	NSGA-II	MOPSO
PG1	4.4066	8.5482	3.1622	11.5906
PG2	24.5709	28.8145	12.4072	20.6855
PG3	80.6596	75.3171	87.9835	83.0087
PG4	106.0443	105.2485	92.0489	96.5241
PG5	7.7217	17.4874	27.8416	11.8301
PG6	61.6727	49.8543	61.8298	71.89055
f_1 Fuel cost ($/h)	1.6761	1.8702	1.873572	1.62575
f_3 Loss (MW)	622.5188	613.3222	623.2921	630.6447

Table 15
The best compromise solutions for emission and loss (f_2–f_3) using different multi-objective algorithms.

	M^2OBA	MOABC	NSGA-II	MOPSO
PG1	20.9113	69.3327	30.4071	22.4024
PG2	33.6887	22.0615	32.0927	25.0679
PG3	77.5503	31.1704	76.1658	102.4471
PG4	65.1039	66.8127	66.2653	63.925
PG5	17.4135	77.6187	33.3860	6.4725
PG6	70.7418	18.4182	47.7291	86.267
f_2 Emission (ton/h)	2.0098	2.0144	2.6463	1.5965
f_3 Loss (MW)	0.21348	0.2134	0.2051	0.2445

Statistical test

In this experiment, in order to further investigate the efficacy and robustness of the proposed M^2OBA on the real-world OPF problem, the analysis of variance (ANOVA) test was conducted to decide on the statistical significance of each of the four tested algorithms over the others. The graphical statistics analyses are done through box plot. A box plot is a graphical tool, which provides an excellent visual summary of many important aspects of a distribution. The box stretches from the lower hinge (defined as the 25th percentile) to the upper hinge (the 75th percentile) and therefore contains the middle half of the scores in the distribution. The median is shown as a line across the box. Therefore, one-fourth of the distribution is between this line and the top of the box and one-fourth of the distribution is between this line and the bottom of the box.

The graphical analysis results of the ANOVA test for all algorithms on both two and three-objective cases in 30 runs are illustrated in Figs. 23–26. Looking at these box plots, the general features of the distribution can be noticed.

The box plots for the best compromise solutions of fuel cost and emission (f_1–f_2) pair in Table 11 are shown in Fig. 23. From Fig. 23(a), we can observe that M^2OBA obtain the best variance distribution of compromise solutions, in which the fuel cost values are ranged from 615 to 620. Then Fig. 23(b) illustrates the M^2OBA achieve both best variance and mean value in all algorithms, which means that M^2OBA provides more robust performance for the f_1–f_2 pairs optimization than that of the other three algorithms.

The box plots for the results presented in Tables 13 and 15 are shown in Figs. 24 and 25, respectively. From the box plot representation in Figs. 24 and 25, it is clearly visible and proved that the M^2OBA p provides better results for each objective function in the two-objective OPF cases, which implies that M^2OBA provides more stable searching ability for all the two-objective OPF cases than that of the other three algorithms.

For the case of three-objective OPF optimization, Fig. 26 shows similar ANOVA results according to Table 17. That is, M^2OBA obtains best results in three single-objective functions, namely the cost, loss, and emission objectives, respectively.

From all the ANOVA results, it is very clear that the proposed M^2OBA yields optimal result in all four tested MO algorithms on the real-world OPF problem.

Computation time analysis

For real-world optimization problems, we need not only the high optimization accuracy and computation robustness, but also a faster solution speed. Hence, in this experiment, the computation time analysis was also carried out to validate the efficacy of the proposed M^2OBA.

Table 14
The best solutions for emission and loss (f_2–f_3) from the Pareto front based on four multi-objective algorithms.

	M^2OBA		MOABC		NSGA-II		MOPSO	
	Best f_2	Best f_3	Best f_2	Best f_2	Best f_3	Best f_3	Best f_2	Best f_3
PG1	45.36	9.11	43.96	33.86	11.95	63.28	32.92	11.48
PG2	44.83	5.00	41.84	59.34	6.50	144.46	44.71	76.36
PG3	56.45	90.97	57.31	50.04	69.67	108.61	62.17	23.07
PG4	40.30	71.16	40.65	34.62	95.30	61.97	42.82	63.32
PG5	46.60	5.00	42.59	63.52	6.41	5.17	50.07	82.53
PG6	53.86	103.75	34.62	47.13	95.19	34.62	52.75	58.56
f_2 Emission (ton/h)	0.1956	0.2592	0.2182	0.1974	0.2523	0.2732	0.1959	0.2823
f_3 Loss (MW)	4.0313	1.6026	4.6123	5.1511	1.6475	2.1242	3.8294	1.7634

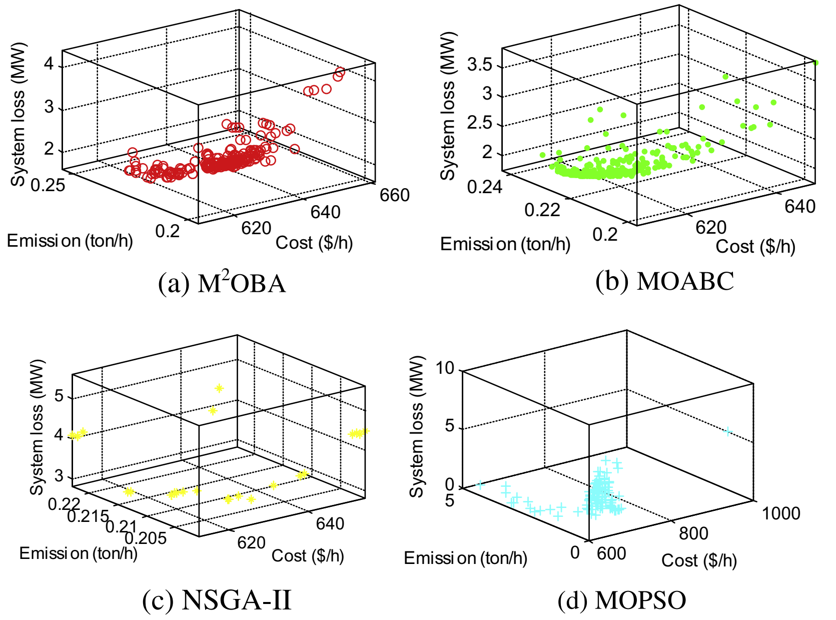


Fig. 22. Pareto fronts obtained by M^{2}OBA, MOPSO, MOABC, and NSGA-II for fuel cost, emission and loss (f_1–f_2–f_3).

Table 16
The best solutions for cost, emission and loss (f_1–f_2–f_3) based on four multi-objective algorithms.

	M^{2}OBA		MOABC		NSGA-II		MOPSO	
	Best f_2	Best f_3	Best f_2	Best f_2	Best f_3	Best f_3	Best f_2	Best f_3
PG1	18.82	45.99	5.05	34.59	2.76	20.63	37.29	52.06
PG2	18.99	44.79	10.47	50.80	33.19	28.78	47.0140	50.56
PG3	71.31	62.78	107.93	63.02	92.06	61.62	58.2576	31.54
PG4	99.15	37.05	70.31	37.32	74.44	103.09	45.5714	45.1823
PG5	44.36	53.13	5.00	46.29	14.42	30.61	50.4057	41.4252
PG6	33.40	44.04	86.22	55.18	68.24	40.99	48.8634	66.9433
f_1 Fuel cost ($/h)	609.85	650.88	661.03	648.48	631.68	608.79	640.0787	662.610
f_2 Emission (ton/h)	0.2243	0.1960	0.2547	0.1963	0.2288	0.2232	0.195650	4.32359
f_3 Loss (MW)	2.6526	4.4066	1.6051	3.8274	1.7422	2.3552	4.0067	0.2010

Table 17
The best compromise solutions for cost, emission and loss (f_1–f_2–f_3) using different multi-objective algorithms.

	M^{2}OBA	MOABC	NSGA-II	MOPSO
PG1	18.7228	18.4350	36.1782	21.27403
PG2	32.9912	28.1939	55.3158	14.7574
PG3	67.4162	73.5849	49.5340	89.6146
PG4	83.1738	86.8403	46.0786	87.0153
PG5	31.9749	26.1992	56.8116	7.1517
PG6	51.4947	52.2758	44.1181	65.3095
f_1 Fuel cost ($/h)	613.1135	613.9435	6.2302	631.3939
f_2 Emission (ton/h)	0.2118	0.2169	0.2022	0.2333
f_3 Loss (MW)	2.3739	2.12932	3.2935	1.7228

Since the time spent on each generation is different from one algorithm to another, the iterations taken by various algorithms do not mean the actual time they spent. For M^{2}OBA, make a hypothesis that the computation cost of one bee in M^{2}OBA is *Time-Cost_i*, the total computation cost for one generation is $N^{*}M^{*}Cost\_i$. However, due to the random and uncertainty of the heuristic algorithms used, it is very difficult to give a theoretic computation analysis for all algorithms involved. In order to straightforwardly evaluate the algorithmic time response, we presented in Fig. 27 the computing times (average in 30 runs) of all algorithms on each multi-objective OPF cases respectively.

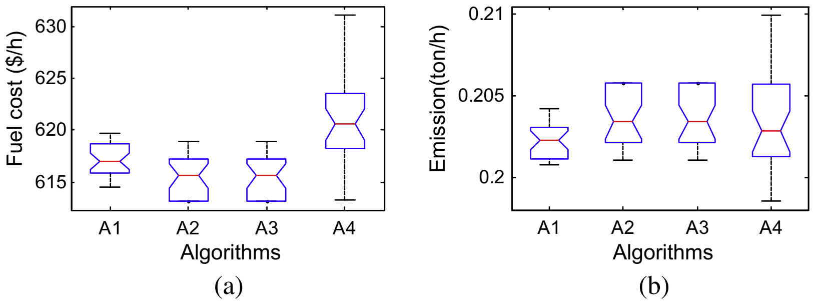


Fig. 23. The box plots of best compromise solutions obtained by M^{2}OBA (A1), MOABC (A2), MOPSO (A3) and NSGA-II (A4) for fuel cost and emission (f_1–f_2) pairs: (a) comprised solutions sets of fuel cost (f_1) and (b) comprised solutions sets of emission (f_2).

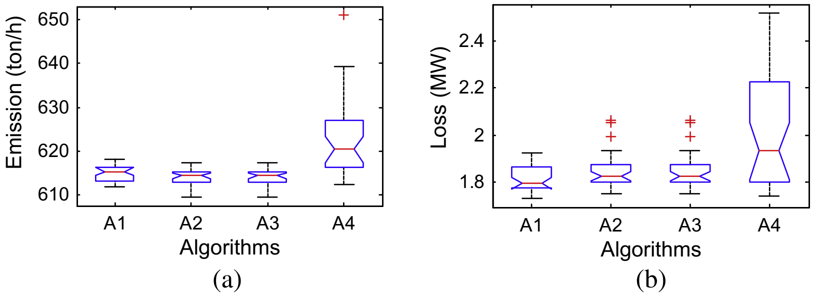


Fig. 24. The box plots of best compromise solutions obtained by M^2OBA (A1), MOABC (A2), MOPSO (A3) and NSGA-II (A4) for fuel cost and loss (f_1–f_3) pairs for 30 runs: (a) comprised solutions sets of fuel cost (f_1) and (b) comprised solutions sets of loss (f_3).

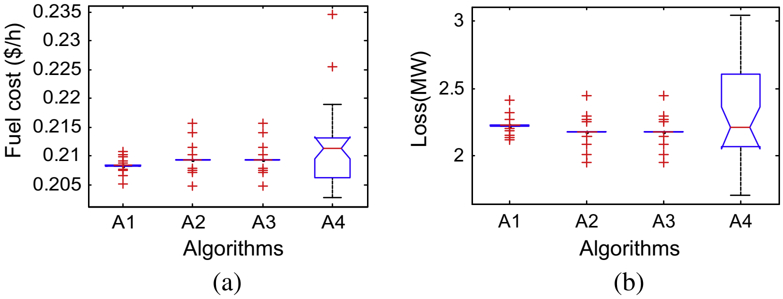


Fig. 25. The box plots of best compromise solutions obtained by M^2OBA (A1), MOABC (A2), MOPSO (A3) and NSGA-II (A4) for emission and loss (f_2–f_3) pairs for 30 runs: (a) comprised solutions sets of emission (f_2) and (b) comprised solutions sets of loss (f_3).

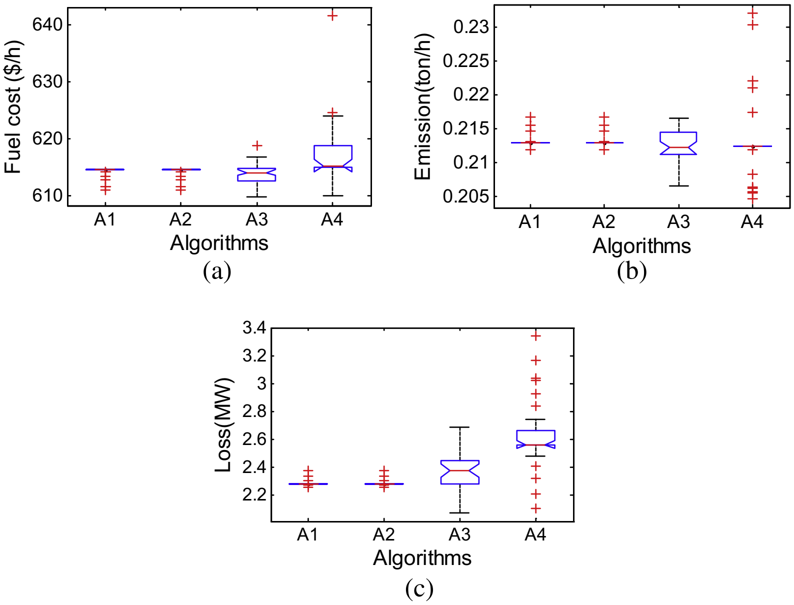


Fig. 26. The box plots of best compromise solutions obtained by M^2OBA (A1), MOABC (A2), MOPSO (A3) and NSGA-II (A4) for fuel cost, emission and loss (f_1–f_2–f_3) pairs for 30 runs: (a) comprised solutions sets of fuel cost (f_1), (b) comprised solutions sets of emission (f_2), (c) comprised solutions sets of loss (f_3).

From Fig. 27, it is visible that the M^2OBA takes the most computing time in all compared algorithms. This can be explained that the multi-hive strategy employed by M^2OBA enhanced the local search ability at cost of increasing the computation amount. Although the computing time spent on a run of optimization procedure for the other three algorithms was less than that for M^2OBA, M^2OBA took smaller iterations to achieve more robust and precise OPF solutions. Therefore, the absolute time required for a run of M^2OBA can be saved by reducing the maximum iterations.

Conclusions

In this paper, different multi-objectives, which consider the cost, loss, and emission impacts, for OPF problem were formed. These multi-objectives have been solved by the proposed M^2OBA, MOPSO, MOABC, and NSGA-II methods.

In the proposed M^2OBA model, we use Pareto concept, external archive, greedy selection and crowding distance strategies, and comprehensive learning approach to make the algorithm converge

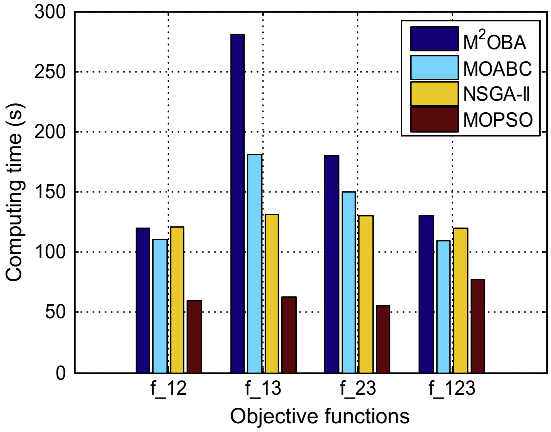


Fig. 27. Computing times comparison of M^{2}OBA, MOABC, MOPSO and NSGA-II for different objective functions pairs (f_12: fuel cost (f_1)–emission (f_2) f_13: fuel cost (f_1)–loss (f_3), f_23: emission (f_2)–loss (f_3), f_123: fuel cost (f_1)–emission (f_2)–loss(f_3)).

to the true Pareto optimal front, and use multiple colony cooperative coevolution to ensure the diversity of the whole population. The M^{2}OBA algorithm is conceptually simple and easy to implement and has considerable potential for solving complex multi-objective optimization problems. With a set of 6 mathematical benchmark functions (including both two and three-objective cases), M^{2}OBA is proved to have better performance than the MOPSO, MOABC, and NSGA-II.

Additionally, a fuzzy membership approach has been used to identify the best compromise solution for the proposed multi-objective OPF optimization model. The simulation studies, which conduct on 30-bus IEEE test system, also show that the M^{2}OBA obtains statistical better than the MOPSO, MOABC, and NSGA-II methods in terms of optimization accuracy and convergence robust.

Acknowledgement

This research is partially supported by National Natural Science Foundation of China und Grant 61105067, 61203161, 61174164 and 51205389.

References

[1] Coello Coello CA, Van Veldhuizen DA, Lamont GB. Evolutionary algorithms for solving multi-objective problems. Norwell, MA: Kluwer; 2002.
[2] Tan KC, Khor EF, Lee TH. Multiobjective evolutionary algorithms and applications. Berlin, Germany: Springer-Verlag; 2005.
[3] Yen GG, Lu H. Dynamic multiobjective evolutionary algorithm: adaptive cell-based rank and density estimation. IEEE Trans Evol Comput 2003;7:253–74.
[4] Naresh Babu AV, Ramana T, Sivanagaraju S. Analysis of optimal power flow problem based on two stage initialization algorithm. Int J Electr Power Energy Syst 2014;55:91–9.
[5] Surender Reddy S, Bijwe PR, Abhyankar AR. Faster evolutionary algorithm based optimal power flow using incremental variables. Int J Electr Power Energy Syst 2014;54:198–210.
[6] Panda Ambarish, Tripathy M. Optimal power flow solution of wind integrated power system using modified bacteria foraging algorithm. Int J Electr Power Energy Syst 2014;54:306–14.
[7] Rezaei Adaryani M, Karami A. Artificial bee colony algorithm for solving multi-objective optimal power flow problem. Int J Electr Power Energy Syst 2013;53:219–30.
[8] Momoh JA, El-Hawary ME, Adapa R. A review of selected optimal power flow literature to 1993. Part I: nonlinear and quadratic programming approaches. IEEE Trans Power Syst 1999;14:96–104.
[9] Momoh JA, El-Hawary ME, Adapa R. A review of selected optimal power flow literature to 1993. Part II: Newton, linear programming and interior point methods. IEEE Trans Power Syst 1999;14:105–11.
[10] Khorsandi A, Alimardani A, Vahidi B, Hosseinian SH. Hybrid shuffled frog leaping algorithm and Nelder–Mead simplex search for optimal reactive power dispatch. IEE Proc-Gener Transm Distrib 2011;5:249–56.
[11] Lai LL, Ma JT, Yokoyama R, Zhao M. Improved genetic algorithm for optimal power flow under both normal and contingent operation states. Electr Power Energy Syst 1997;19:287–92.
[12] Bakirtzis AG, Biskas PN, Zoumas CE, Petridis V. Optimal power flow by enhanced genetic algorithm. IEEE Trans Power Syst 2002;17:229–36.
[13] Roa-Sepulveda CA, Pavez-Lazo BJ. A solution to the optimal power flow using simulated annealing. Electr Power Energy Syst 2003;25:47–57.
[14] Sousa T, Soares J, Vale ZA, Morais H, Faria P. Simulated annealing metaheuristic to solve the optimal power flow. In: 2011 IEEE power and energy society general meeting; 2011. p. 1–8.
[15] Abido MA. Optimal power flow using tabu search algorithm. Electr Power Compon Syst 2002;30:469–83.
[16] Sayah S, Zehar K. Modified differential evolution algorithm for optimal power flow with non-smooth cost functions. Energy Convers Manage 2008;49:3036–42.
[17] Nayak MR, Krishnanand KR, Rout PK. Modified differential evolution optimization algorithm for multi-constraint optimal power flow. In: 2011 international conference on energy, automation, and signal (ICEAS); 2011. p. 1–7.
[18] Sivasubramani S, Swarup KS. Multi-objective harmony search algorithm for optimal power flow problem. Electr Power Energy Syst 2011;33:745–52.
[19] Roy PK, Ghoshal SP, Thakur SS. Biogeography based optimization for multiconstraint optimal power flow with emission and non-smooth cost function. Expert Syst Appl 2010;37:8221–8.
[20] Rarick R, Simon D, Villaseca FE, Vyakaranam B. Biogeography-based optimization and the solution of the power flow problem. In: IEEE international conference on systems, man and cybernetics, 2009 (SMC 2009); 2009. p. 1003–8.
[21] AlRashidi MR, El-Hawary ME. Hybrid particle swarm optimization approach for solving the discrete OPF problem considering the valve loading effects. IEEE Trans Power Syst 2007;22(4):2030–8.
[22] Gao ZW, Pang HL, Wang DW, Chen HN. RFID network scheduling based on symbiotic particle swarm optimization. Inf Control 2012;41(5):564–70.
[23] Chen HN, Zhu YL, Hu KY. Cooperative bacterial foraging optimization. Discrete Dynamic in Nature and Society; 2009.
[24] Qin L, Chen L, Zhou RG, Gu Q, Wu Y. A multicast routing algorithm based on ant colony system. Inf Control 2006;35(5):545–50.
[25] Chen JY, Yang DY. Hybrid intrusion detection method based on light-weighted artificial immune computation. Inf Control 2012;41(5):529–36.
[26] Gao W, Liu S. Improved artificial bee colony algorithm for global optimization. Inf Process Lett 2011;111:871–82.
[27] Karaboga D, Akay B. A comparative study of artificial bee colony algorithm. Appl Math Comput 2009;214:108–32.
[28] Omkar SN, Senthilnath J, Khandelwal Rahul, Narayana Naik G, Gopalakrishnan S. Artificial bee colony (ABC) for multi-objective design optimization of composite structures. Appl Soft Comput 2011;11(1):489–99.
[29] Hedayatzadeh R, Hasanizadeh B, Akbari R, Ziarati K. A multi-objective artificial bee colony for optimizing multi-objective problems. In: 3rd international conference on advanced computer theory and engineering (ICACTE), Chengdu, China; 2010. p. 277–81.
[30] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 2002;6(2):182–97.
[31] Zitzler E, Laumanns M, Thiele L. SPEA2: improving the strength pareto evolutionary algorithm. In: Proc EUROGEN 2001: evolutionary methods design optimization control appl ind problems, Athens, Greece, 2002. p. 95–100.
[32] Coello Coello Carlos A, Pulido GT, Lechuga Maximino Salazar. Handling multiple objectives with particle swarm optimization. IEEE Trans Evol Comput 2004;8(3):256–79.
[33] Zhang Q, Liu W, Li H. The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. In: Proc of congress on evolutionary computation (CEC 2009), Norway; 2009. p. 203–8.
[34] Zhao SZ, Suganthan PN. Two-lbests based multi-objective particle swarm optimizer. Eng Optimiz 2011;43(1):1–17.
[35] Qu BY, Suganthan PN. Multi-objective evolutionary algorithms based on the summation of normalized objectives and diversified selection. Inf Sci 2010;180(17):3170–81.
[36] Qu BY, Suganthan PN. Constrained multi-objective optimization algorithm with ensemble of constraint handling methods. Eng Optimiz 2011;43(4): 403–16.
[37] Hwang CL, Masud AS. Multiple objective decision making – methods and applications. Berlin, Germany: Springer-Verlag; 1979.
[38] Jensen MT. Reducing the run-time complexity of multiobjective EAs: the NSGA-II and other algorithms. IEEE Trans Evol Comput 2003;7:503–15.
[39] Schaffer JD. Multiple objective optimization with vector evaluated genetic algorithms. In: Proceedings of the 1st international conference on genetic algorithms. NJ, USA: L. Erlbaum Associates; 1985. p. 93–100.
[40] Moore J, Chapman R. Application of particle swarm to multiobjective optimization. Ph.D. thesis, Auburn University; 1999.
[41] Leong WF, Yen GG. PSO-based multiobjective optimization with dynamic population size and adaptive local archives. IEEE Trans Syst, Man, Cybern, Part B: Cybern 2008;38(5):1270–93.
[42] Liu DS, Tan KC, Goh CK, Ho WK. A multiobjective memetic algorithm based on particle swarm optimization. IEEE Trans Syst Man Cybern B Cybern 2007;137(1):42–50.
[43] Walters DC, Sheble GB. Genetic algorithm solution of economic dispatch with valve point loading. IEEE Trans Power Syst 1993;8:1325–32.
[44] Alrashidi MR, El-Hawary ME. Hybrid particle swarm optimization approach for solving the discrete OPF problem considering the valve loading effects. IEEE Trans Power Syst 2007;22:2030–8.

[45] Knowles JD, Corne DW. Approximating the nondominated front using the pareto archived evolution strategy. Evol Comput 2000;8(2):149–72.

[46] Liang JJ, Qin AK, Suganthan PN, Baskar S. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Trans Evol Comput 2006;10(3):281–95.

[47] Tomassini M. Spatially structured evolutionary algorithms: artificial evolution in space and time. Springer; 2005.

[48] Niu B, Zhu YL, He XX, Zeng XP, Wu H. MCPSO: a multi-swarm cooperative particle swarm optimizer. Appl Math Comput 2007;185(2):1050–62.

[49] Chen HN, Zhu YL, Hu KY. Multi-colony bacteria foraging optimization with cell-to-cell communication for RFID network planning. Appl Soft Comput 2010;10(2):539–47.

[50] Mohammed E, Mohamed K. A taxonomy of cooperative search algorithms. Lect. Notes Comput. Sci. 2005;3636:32–41.

[51] Ghaffarian R. CCGA packages for space applications. Microelectron Reliab 2006;46(12):2006–24.

[52] Bergh F, Engelbrecht AP. A cooperative approach to particle swarm optimization. IEEE Tans Evolut Comput 2004;8(3):225–39.

[53] Wolpert DH, Macready WG. No free lunch theorems for search. IEEE Trans Evol Comput 1997;1(1):67–82.

[54] Dhillon JS, Parti SC, Kothari DP. Stochastic economic emission load dispatch. Electr Power Syst Res 1993;26:186–97.

[55] Alsac O, Stott B. Optimal load flow with steady-state security. IEEE Trans Power Appar Syst 1974;93(3):745–51.