Research Article

Evolutionary Multimodal Optimization Based on Bi-Population and Multi-Mutation Differential Evolution

Wei Li^{1,2,*}, Yaochi Fan¹, Qingzheng Xu³

¹School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China

²Shaanxi Key Laboratory for Network Computing and Security Technology, Xi'an 710048, China

³College of Information and Communication, National University of Defense Technology, Xi'an 710106, China

ARTICLE INFO

Article History

Received 28 Jan 2020

Accepted 23 Aug 2020

Keywords

Differential evolution
Multi-mutation strategy
Fitness Euclidean-distance ratio
Multimodal optimization problems

ABSTRACT

The most critical issue of multimodal evolutionary algorithms (EAs) is to find multiple distinct global optimal solutions in a run. EAs have been considered as suitable tools for multimodal optimization because of their population-based structure. However, EAs tend to converge toward one of the optimal solutions due to the difficulty of population diversity preservation. In this paper, we propose a bi-population and multi-mutation differential evolution (BMDE) algorithm for multimodal optimization problems. The novelties and contribution of BMDE include the following three aspects: First, bi-population evolution strategy is employed to perform multimodal optimization in parallel. The difference between inferior solutions and the current population can be considered as a promising direction toward the optimum. Second, multi-mutation strategy is introduced to balance exploration and exploitation in generating offspring. Third, the update strategy is applied to individuals with high similarity, which can improve the population diversity. Experimental results on CEC2013 benchmark problems show that the proposed BMDE algorithm is better than or at least comparable to the state-of-the-art multimodal algorithms in terms of the quantity and quality of the optimal solutions.

© 2020 The Authors. Published by Atlantis Press B.V.

This is an open access article distributed under the CC BY-NC 4.0 license (<http://creativecommons.org/licenses/by-nc/4.0/>).

1. INTRODUCTION

In the area of optimization, there has been a growing interest in applying optimization algorithms to solve large-scale optimization problems, multimodal optimization problems (MMOPs), multiobjective optimization problems (MOPs), constrained optimization problems, etc. [1–6]. Different from unimodal optimization, multimodal optimization seeks to find multiple distinct global optimal solutions instead of one global optimal solution. When multiple optimal solutions are involved, the classic evolutionary algorithms (EAs) are faced with the problem of maintaining all the optimal solutions in a single run. It is difficult to realize because the evolutionary strategies of EAs make the entire population converge to a single position [7]. For the purpose of locating multiple optima, a variety of niching methods and multiobjective optimization methods incorporated into EAs have been widely developed. The related techniques [8–16] include classification, clearing, clustering, crowding, fitness sharing, multiobjective optimization, neighborhood strategies, restricted tournament selection (RTS), speciation, etc. These techniques have successfully enabled EAs to solve MMOPs. Nevertheless, some critical issues still remain to be resolved. First, some radius-based

niching methods introduce new parameters that directly depend on the problem landscapes. The performance of algorithm often deteriorates when the selected parameters do not match the problem landscapes well. Second, some niching techniques employ sub-populations. However, the sub-populations may suffer from the genetic drift or they may be wasted to discover the same solution for some problems with complex landscapes. Third, when an offspring and its neighbor both sit on different peaks, either one of two peaks will be lost because only the winner can survive.

Diversification and intensification are two major issues in multimodal optimization [17]. The purpose of diversification is to ensure sufficient diversity in the population so that individuals can find multiple global optima. On the other hand, intensification allows individuals to congregate around potential local optima. Consequently, each optimum region is fully exploited by individuals. As a popular EA, differential evolution (DE) has shown to be suitable for finding one global optimal solution. However, it is inappropriate for finding multiple distinct global optimal solutions [2]. The one-by-one selection used in DE does not consider selecting individuals according to different peaks, which has disadvantage for diversity preservation. Moreover, it is a dilemma to choose an appropriate mutation scheme that favors both diversification and intensification. To solve these drawbacks, we propose a novel multimodal optimization algorithm (BMDE). Specifically, bi-population evolution strategy, multi-mutation strategy, and update strategy are

* Corresponding author. Email: liwei@xaut.edu.cn

proposed to help BMDE to accomplish diversification and intensification for locating multiple optimal solutions. The novelties and advantages of this paper are summarized as follows:

1. **Bi-population Evolution Strategy:** From the optimization perspective, the parent and its offspring compete to decide which one will survive at each generation. Losers will be discarded. However, historical data is usually beneficial to improve the convergence performance. In particle swarm optimization (PSO) [18], the previous best solutions of each particle are used to direct the movement of the current population. In addition, research shows that the difference between the inferior solutions and the current population can be considered as a promising direction toward the optimum [19]. Motivated by this consideration, we are interested in a set of individuals who fail in competition and consider their difference from the current population. More precisely, this paper employs two population to perform multimodal optimization in parallel. One population is employed to save the individuals who win in the competition, denoted as evolution population. The other population is employed to save the individuals who fail in the competition, denoted as inferior population. The evolution population may bear stronger exploitation capability. The inferior population may help to maintain the population diversity, which can avoid losing the potential global optima found by the population. In addition, inferior population can prevent the loss of potential optima due to replacement error.
2. **Multi-mutation strategy:** Generally, the performance of DE often deteriorates with the inappropriate choice of mutation strategy. Therefore, many mutation strategies have been designed for different optimization problems. In this paper, we introduce a multi-mutation strategy, which includes two effective mutation strategies with stronger exploration capability. Consequently, it is more suitable for solving multimodal problems.
3. **Update strategy:** As evolution proceeds, the population will move toward the peaks (global optima). If the number of individuals around a peak is too small, the population may not be able to find highly accurate solutions due to their poor capability of diversity preservation. Conversely, if the number of individuals around a peak is too large, many individuals will do duplication of labor, thus wasting the calculation cost. Moreover, the best individual will take over the population's resources and flood the next generation with its offspring. To address this problem, the update strategy is employed to eliminate the individual with high similarity and generate new individuals.
4. To assess the performance of the proposed algorithm, extensive experiments are conducted on CEC2013 multimodal benchmark problems [20], in comparison with thirteen state-of-the-art algorithms [2,13,21–25] for multimodal optimization. The experimental results show that the proposed method is promising for solving MMOPs.

The rest of this paper is organized as follows: Section 2 reviews the multimodal optimization formulation. In Section 3, we introduce the basic framework of DE algorithm and review some background knowledge of multimodal optimization techniques. In Section 4,

the details of the proposed algorithm are described. Experiments are presented in Section 5. Section 6 gives the conclusion and future work.

2. MULTIMODAL OPTIMIZATION FORMULATION

An optimization problem which have multiple global and local optima is known as MMOP. Without loss of generality, MMOP can be mathematically expressed as follows:

$$\begin{aligned} & \text{maximize } f(\mathbf{x}) \\ & \text{s.t. } \mathbf{x} \in S \end{aligned} \quad (1)$$

where $f(\mathbf{x})$ is the objective function. $\mathbf{x} = (x_1, \dots, x_i, \dots, x_D)$ is the decision vector. x_i is the i th decision variable. D is the dimension of the optimization problem. The decision space S is presented as

$$S = \prod_{i=1}^D [x_i^{\min}, x_i^{\max}] \quad (2)$$

where $x_i^{\min}$ and $x_i^{\max}$ denote the lower bound and upper bound for each decision variable x_i , respectively. In the case of a multi-modal problem, we seek a set of global optimal solutions $\mathbf{x}^*$ that maximize the objective function $f(\mathbf{x})$.

3. RELATED WORK

A. Differential Evolution

DE [26] is a very competitive optimizer for optimization problems. The key steps of DE algorithm are initialization, mutation, crossover, and selection, which are briefly introduced below.

1. **Initialization:** For an optimization problem of dimension D , a population $\mathbf{x}$ of NP real-valued vectors (or individuals) is typically initialized at random in accordance with a uniform distribution in the search space S . The j th decision variable of the i th individual at generation g can be initialized as follows:

$$x_{i,j,g} = L_j + rand \times (U_j - L_j), i = 1, \dots, NP, j = 1, \dots, D \quad (3)$$

where L_j and U_j are the lower and upper bounds of j th dimension, respectively. *rand* represents a uniformly distributed random number in the range of (0,1).

2. **Mutation:** In each generation, a mutant vector is formed for each individual based on scaled difference individuals. The frequently used mutation operators are listed below.
“DE/rand/1”

$$\mathbf{v}_{i,G} = \mathbf{x}_{r_1,G} + F (\mathbf{x}_{r_2,G} - \mathbf{x}_{r_3,G}) \quad (4)$$

“DE/best/1”

$$\mathbf{v}_{i,G} = \mathbf{x}_{\text{best},G} + F (\mathbf{x}_{r_1,G} - \mathbf{x}_{r_2,G}) \quad (5)$$

“DE/rand/2”

$$\mathbf{v}_{i,G} = \mathbf{x}_{r_1,G} + F(\mathbf{x}_{r_2,G} - \mathbf{x}_{r_3,G}) + F(\mathbf{x}_{r_4,G} - \mathbf{x}_{r_5,G}) \quad (6)$$

“DE/best/2”

$$\mathbf{v}_{i,G} = \mathbf{x}_{\text{best},G} + F(\mathbf{x}_{r_1,G} - \mathbf{x}_{r_2,G}) + F(\mathbf{x}_{r_3,G} - \mathbf{x}_{r_4,G}) \quad (7)$$

“DE/current-to-best/1”

$$\mathbf{v}_{i,G} = \mathbf{x}_{i,G} + F(\mathbf{x}_{\text{best},G} - \mathbf{x}_{i,G}) + F(\mathbf{x}_{r_1,G} - \mathbf{x}_{r_2,G}) \quad (8)$$

“DE/current-to-rand/1”

$$\mathbf{v}_{i,G} = \mathbf{x}_{i,G} + K(\mathbf{x}_{r_1,G} - \mathbf{x}_{i,G}) + F'(\mathbf{x}_{r_2,G} - \mathbf{x}_{r_3,G}) \quad (9)$$

where $i = 1, \dots, NP$, $r_1, r_2, r_3 \in \{1, \dots, NP\}$ are randomly selected and satisfy $r_1 \neq r_2 \neq r_3 \neq i$, $F \in [0, 1]$, F is control parameter. In “DE/current-to-rand/1,” K is the combination coefficient, which should be chosen with a uniform random distribution from $[0, 1]$ and $F' = K \cdot F$.

3. *Crossover*: DE employs a binomial crossover operation on the target vector $\mathbf{x}_i$ and the mutant vector $\mathbf{v}_i$ to form a trial vector $\mathbf{u}_i$ as follows:

$$u_{i,G}^j = \begin{cases} v_{i,G}^j & \text{if } \text{rand}_j(0, 1) \leq Cr \text{ or } j = j_{\text{rand}} \\ x_{i,G}^j & \text{otherwise} \end{cases} \quad j = 1, 2, \dots, D \quad (10)$$

where j_{rand} is randomly selected from $[1, D]$ to ensure that the trial vector has at least one dimension differ from the target vector. Cr is the crossover rate lying in the range of $[0, 1]$.

4. *Selection*: The selection operator is employed to decide which one is to survival in the next generation through a one-to-one competition between the target vector and its trial vector. Take the maximization optimization problem as an example, the selection process can be outlined as

$$\mathbf{x}_{i,G+1} = \begin{cases} \mathbf{u}_{i,G}, & \text{if } f(\mathbf{u}_{i,G}) \geq f(\mathbf{x}_{i,G}) \\ \mathbf{x}_{i,G}, & \text{otherwise} \end{cases} \quad (11)$$

where $f(\cdot)$ is the objective function to be optimized (maximized).

As mentioned before, DE or other EAs have been employed for multimodal optimization. However, it is difficult for the algorithms to locate all the global optima in a run. On the one hand, the algorithm should maintain sufficient diversity to ensure the individuals to spread out widely within the search space. On the other hand, the individuals gather around potential local optima to fully exploit each optimum region. Therefore, it is greatly desirable for DE or other EAs to make a balance between exploration (diversity) and exploitation (convergence).

B. Niching Techniques for Multimodal Optimization

Niching is a concept derived from biology and refers to a living environment. The species evolve in different living environments. In terms of MMOP, niching refers local regions around an optimum. A brief review of representative niching techniques is given in the following.

The crowding strategy proposed by De Jong is one of the simplest niching techniques for MMOPs. The advantage of this strategy is that it can maintain the diversity of the whole population. However, the potential optimal solution will be replaced if the offspring is a superior solution. This phenomenon is called replacement error. To overcome this problem, deterministic crowding and probabilistic crowding, two improvement strategies to the original crowding, are proposed [11,27]. The deterministic crowding can effectively reduce replacement errors, while probabilistic crowding can prevent the loss of niches with lower fitness or loss of local optima. Nevertheless, the disadvantage of probabilistic crowding is slow convergence and poor fine searching ability. The restricted tournament selection (RTS) employs Euclidean or Hamming distance to find the nearest member within the w (window size) individuals. Similar to crowding, RTS can ensure the population diversity by the competition between similar individuals. However, this strategy suffers from the replacement error.

Both the fitness sharing strategy and speciation employ the grouping method. More precisely, the population is divided into different sub-populations according to the similarity of the individuals. The advantage of fitness sharing strategy is the stable niches maintained. However, the niche radius σ_{share} and r_s used in sharing and speciation, respectively, are difficult to specify for lacking of prior knowledge of the problems. Moreover, the computational complexity of sharing is much expensive.

In order to improve the niching techniques mentioned above, the neighborhood base CDE (NCDE) and the neighborhood based SDE (NSDE) are proposed in [2]. However, a new parameter m , which is the neighborhood size, is introduced in NCDE and NSDE. In order to relieve the influence of the parameter, Biswas *et al.* [28] introduced an improved local information sharing mechanism in niching DE, where the neighborhood size is dynamically changed in a nonlinear way. Shir *et al.* [25] proposed a niching variant of the covariance matrix adaptation evolution strategy (CMA-ES), where a technique called dynamic peak identification (DPI) is used to split the population into species. The number of niches, that is, the number of peaks of optimization problem, is predefined in DPI. However, it is practically difficult to know in advance the number of peaks that exist within the landscape of optimization problem.

C. Novel Operators for Multimodal Optimization

Wang *et al.* [29] introduced a dual-strategy mutation scheme in DE to balance exploration and exploitation in generating offspring. Moreover, in order to select suitable individuals from different clusters, an adaptive probabilistic selection mechanism based on affinity propagation clustering is proposed. In order to eliminate the requirement of prior knowledge to specify certain niching parameters, Qu *et al.* [14] proposed a distance-based locally informed particle swarm (LIPS) optimizer. LIPS employs the Euclidean-distance-based neighborhood information to guide the search process. Therefore, LIPS is able to form different stable niches and locate the desired peaks with high accuracy. However, the neighborhood size that influences the performance of the proposed LIPS is difficult to exactly specify. A larger neighborhood size is suitable for convergence while a smaller neighborhood size is suitable for maintaining the diversity of the population. To achieve a balance between local exploitation and global exploration with speciation niching technique, Hui *et al.* [30] proposed an ensemble and

arithmetic recombination-based speciation differential evolution algorithm (EARSDE), in which the arithmetic recombination is used to enhance exploration and the neighborhood mutation is employed to improve exploitation of individual peaks. Haghighi *et al.* [3] proposed an improved niching strategy which employs a nearest neighbor scheme and hill valley algorithm to control the interaction between GSA agents. Fieldsend *et al.* [31] proposed the niching migratory multi-swarm optimizer (NMMSO) which dynamically manages many particle swarms. In NMMSO, multiple swarms which have strong local search continuously merge and split in the search landscape. More precisely, the swarms will split if new regions are identified. In addition, they merge to prevent duplication of labor. Li [22] proposed RPSO (including $r2PSO$ and $r3PSO$) by employing ring topology which using individual particles' local memories to form a stable network. The network can retain the best particles which explore the search space more broadly.

D. Multiobjective Optimization Techniques for Multimodal Optimization

Similar to MOPs, a MMOP involves multiple optimal solutions. Therefore, a few attempts have been made to solve MMOPs by taking advantage of multiobjective optimization in maintaining good population diversity. For instance, Wang *et al.* [13] proposed a novel transformation technique MOMMOP which transforms an MMOP into an MOP with two conflicting objectives. Moreover, MOMMOP finds a trade-off between exploitation and exploration by balancing the decision variable and the objective function. Basak *et al.* [32] brought up a novel bi-objective formulation of the MMOP, in which the mean distance-based selection mechanism is chosen as the second objective to prevent the entire population from converging to a single optimum. Yao *et al.* [17] proposed a bi-objective multipopulation generic algorithm which employs a novel bi-objective mechanism and a multipopulation scheme to realize exploration. Therefore, the sub-populations of BMPGA are evolved toward two objectives separately and simultaneously instead of toward a single fitness objective. Li [33] introduced a multiobjective optimization method into the matrix adaptation evolution strategy (MA-ES) to solve MMOPs.

4. BMDE FOR MMOP

A. Framework of BMDE

The main framework of the proposed BMDE is summarized in Algorithm 1, from which we can see that BMDE is composed of three main strategies: (1) bi-population strategy; (2) multi-mutation strategy; and (3) update strategy. In bi-population evolution strategy, one population consists of the individuals with higher fitness value is employed for exploitation, while the other population which consists of the individuals with lower fitness value is used for exploration. Then, a multi-mutation strategy is employed to improve the solution accuracy and find more optimal solutions. Finally, update strategy is performed to generate new individuals to replace those with high similarity. The following sections will detail the three main strategies in Algorithm 1 successively.

Algorithm 1: Framework of BMDE

```

1: Initialize the population and set the parameters
2: Evaluate the fitness of the individuals in the population
3: while termination criteria is not satisfied
4:   Multi-mutation strategy (Algorithm 2)
5:   Bi-population evolution strategy (Algorithm 3)
6:   Update strategy (Algorithm 4)
7: end while

```

B. Multi-Mutation Strategy

DE has been proved to be an efficient algorithm in the past two decades due to its powerful global optimization properties on a wide range of problems, and its simplicity [34]. Here, we exploit the DE algorithm for multimodal optimization. As mentioned before, there are several widely used mutation strategies of DE. Different mutation strategies have different characteristics. DE/rand/1 and DE/rand/2 have a high diversity to avoid premature convergence. However, their convergence rate is poor. DE/best/1 and DE/best/2 perform better in convergence rate, however, they are easy to be trapped in local minima. As for DE/current-to-best/1 or DE/current-to-rand/1, it can be employed as a mixed strategy that combines a current individual and a best individual (or a random individual) as the origin individual and the terminal individual, respectively to form a composite base vector [30]. DE/current-to-best/1 and DE/current-to-rand/1 can escape from local minima with better diversity than DE/best/1 and DE/best/2.

As shown previously, niching is an effective technique to locate multiple optima in parallel. However, most niching methods have difficulties that need to be overcome [22], such as reliance on prior knowledge of some niching parameters, difficulty in maintaining found solutions in a run, higher computational complexity, etc. Inspired from the aforesaid observations in DE, we employ a multi-mutation strategy that utilizes two different mutation strategies instead of utilizing niching technique. One mutation strategy is DE/rand/1 which has a high diversity to avoid premature convergence. The other mutation strategy is developed based on DE/current-to-rand/1 and fitness Euclidean-distance ratio [21] which encourages the survival of fitter and closer individuals, denoted as DE_FER. The mutation strategy DE_FER can be calculated as follows:

$$\mathbf{v}_{i,G} = \mathbf{x}_{i,G} + \text{rand}_{i,G} \times (\mathbf{x}_{\text{fer},G} - \mathbf{x}_{i,G}) \quad (12)$$

where $\text{rand}_{i,G}$ of i th individual is a uniform random distribution from $[0, 1]$. In order to search for multiple peaks, instead of using a single global best at generation G , each individual is attracted toward $\mathbf{x}_{\text{fer},G}$, a fittest-and-closest neighborhood individual, which is achieved via computing its fitness and Euclidean-distance ratio. The fitness and Euclidean-distance ratio of $\mathbf{x}_{i,G}$ is described as follows [21]:

$$\text{FER} = \eta \frac{f(\mathbf{x}_{i,G}) - f(\mathbf{x}_{j,G})}{\|\mathbf{x}_{i,G} - \mathbf{x}_{j,G}\|_2} \quad j = 1, 2, \dots, NP \quad (13)$$

where NP is the population size. $f(\bullet)$ is the objective (fitness) function to be optimized (maximized). $\|\bullet\|_2$ is Euclidean-distance of two

individuals. η is a scaling factor to avoid fitness and Euclidean distance dominating each other.

$$\eta = \frac{\|U - L\|_2}{f(\mathbf{x}_{best}) - f(\mathbf{x}_{worst})} \quad (14)$$

where L and U denote the lower bound and upper bound of the search space of D dimension, respectively. $\mathbf{x}_{best}$ and $\mathbf{x}_{worst}$ are the best-fit individual and the worst-fit individual in the current generation, respectively.

The index (fer) corresponding to the maximum value of FER is defined by

$$fer = \arg \max_{1 \leq j \leq NP} FER_j \quad (15)$$

The multi-mutation strategy is described in Algorithm 2. NP is the population size. $\delta = \frac{1}{D+mod(D,2)}$, D is the dimension, and $mod(\bullet)$ is modular operation.

Algorithm 2: Multi-mutation strategy

```

1: for each individual  $\mathbf{x}_i$  ( $1 \leq i \leq NP$ )
2:   if  $rand < \delta$ 
3:     Apply DE/rand/1 to generate the offspring
4:   else
5:     Apply DE_FER to generate the offspring
6:   end if
7: end for

```

Multi-mutation strategy is able to keep a balance between the exploration and exploitation, which make it suitable for multimodal problem. DE/rand/1 uses the rand vector selected from the population to generate donor vectors. The scheme promotes exploration since all the vectors/individuals are attracted toward different position in the search space through iterations, thereby ensuring the population diversity. DE_FER uses the local best vector, i.e., a fittest-and-closest neighborhood individual. The local best vector can guide other vectors/individuals to exploit, so as to enhance local search ability.

C. Bi-population Evolution Strategy

Diversity and convergence are two targets that need to be achieved for dealing with MMOPs efficiently. As evolution progresses, the individuals with higher fitness tend to attract other inferior individuals, thereby inducing hill climbing behavior [34]. On the one hand, this behavior leads to loss of some peaks because of selection pressure and genetic drifts. On the other hand, as the research in [19] pointed out, historical data is another source that can be used to improve the algorithm performance. Hence, we are interested in a set of recently explored inferior individuals which may be used as a promising direction toward the optimum.

Motivated by this observation, instead of employing a single population in conventional DE, we use bi-population to achieve a good balance between locating different peaks by exploring and improving the solution precision by exploiting. Here, we denote FP as the evolution population to save the individuals who win in the competition and denote SP as the inferior population to save the individuals who fail in the competition. First, SP is initiated to be empty.

For each offspring $\mathbf{u}_i$, we calculate the Euclidean distance of $\mathbf{u}_i$ to the individuals in the current population. Then, the minimum Euclidean distance is calculated as follows:

$$dist_{j'} = \arg \min_{1 \leq j \leq NP} \frac{\|\mathbf{u}_{i,G} - \mathbf{x}_{j,G}\|_2}{\|U - L\|_2} \quad (16)$$

where j' is the index of the nearest individual to the offspring $\mathbf{u}_i$. L and U denote the lower bound and upper bound of the search space, respectively.

Compare the fitness of the offspring $\mathbf{u}_i$ and the individual $\mathbf{x}_{j'}$, if $\mathbf{u}_i$ is more fit than $\mathbf{x}_{j'}$, the superior individual $\mathbf{u}_i$ will replace $\mathbf{x}_{j'}$ in the population FP. Meanwhile, if the Euclidean distance between two individuals is greater than σ , the worse individual $\mathbf{x}_{j'}$ will be saved in the population SP. If the size of SP (denotes as $|SP|$) is larger than k times the population size NP , the $(|SP| - k \times NP)$ worst-ranking individuals are deleted from the population. The bi-population evolution strategy is described in Algorithm 3. σ is set to 0.01, which is based on a large number of experiments.

Algorithm 3: Bi-population Evolution Strategy

```

1: for each offspring  $\mathbf{u}_i$  ( $1 \leq i \leq NP$ )
2:   Find the individual  $j'$  which has minimum Euclidean distance to  $\mathbf{u}_i$ 
3:   if  $f(\mathbf{u}_i) > f(\mathbf{x}_{j'})$ 
4:     if  $dist_{j'} > \sigma$ 
5:        $SP = SP \cup \mathbf{x}_{j'}$ 
6:     end if
7:      $\mathbf{x}_{j'} = \mathbf{u}_i$ 
8:   end if
9: end for
10: if  $|SP| > k \times NP$ , Remove the worst individuals so that  $|SP| = k \times NP$ 

```

Bi-population evolution strategy employs two populations to perform multimodal optimization in parallel. On the one hand, the difference between inferior individuals and the current individuals can be considered as a promising direction toward the optimum. Therefore, inferior population can be employed to improve the population diversity and explore more peaks. On the other hand, evolution population that consists of superior individuals have good fitness, which can exploit the region around the superior individuals to accelerate the convergence speed.

D. Update Strategy

During the evolution process, the individuals are attracted toward the superior individuals. After several generations of evolution, many individuals may have high similarity because of selection pressure. These individuals may explore the same peak. Obviously, this is a waste of resources. In addition, if the number of individuals surrounded a peak is too small, this behavior may lead to loss of the peak. As a result, it is difficult for the algorithm to find as many distinct global peaks as possible. Therefore, the proposed algorithm eliminates the individuals which have high similarity and generates new individuals to enhance the population diversity.

The location (U) of updated individual is calculated as follows:

$$U = \{ \|\mathbf{x}_{i,G} - \mathbf{x}_{j,G}\|_2 < \sigma, 1 \leq j \leq NP, j \neq i \} \quad (17)$$

The setting of σ is the same as that of σ in Algorithm 3.

For all updated individuals ($u \in U$), the mutation strategy can be calculated as follows:

$$\mathbf{v}_{u,G} = \mathbf{x}_{r_1,G} + F \left(\mathbf{x}_{r_2,G} - \tilde{\mathbf{x}}_{r_3,G} \right) \quad (18)$$

where $\mathbf{x}_{r_1,G}$, $\mathbf{x}_{r_2,G}$ are randomly selected from the current population and satisfy $r_1 \neq r_2$, while $\tilde{\mathbf{x}}_{r_3,G}$ is randomly selected from the inferior population SP.

The pseudocode of the population updating is shown in Algorithm 4.

Algorithm 4: Update the population

```

1: for each individual  $\mathbf{x}_i$  ( $1 \leq i \leq NP$ )
2:   Locate the individual to be updated according to Eq. (17)
3:   for each updated individual  $\mathbf{x}_u$  ( $u \in U$ )
4:     if  $f(\mathbf{x}_u) < f(\mathbf{x}_i)$ 
5:       Perform mutation on  $\mathbf{x}_u$  according to Eq. (18)
6:     endif
7:   endfor
8: endfor
9: Output: population  $\mathbf{x}$ 

```

E. Complexity Analysis

The computational cost of BMDE mainly comes from the multi-mutation strategy, bi-population evolution strategy, and update strategy. For multi-mutation strategy, a loop over NP (population size) is conducted, containing a loop over D (dimension). The mutation and crossover operations are performed at the component level for each individual. Then, the runtime complexity is $O(NP \cdot D)$ at each iteration. For bi-population evolution strategy, first, we need to compare the objective function values of the offspring $\mathbf{u}_i$ and its nearest individual $\mathbf{x}_j$ in order to save better individual. Next, if the Euclidean distance between $\mathbf{u}_i$ and $\mathbf{x}_j$ is greater than the given threshold, the worse individual $\mathbf{x}_j$ will be saved in the inferior population. Hence in the worst possible case, the runtime complexity is $O(2 \cdot NP)$ at each iteration. For update strategy, in the worst possible case, if there are m ($m < NP$) individuals to be updated, the runtime complexity is $O(m \cdot NP \cdot D)$ at each iteration. The total time complexity of BMDE at one iteration can be estimated as follows:

$$T(NP) = NP \cdot D + 2 \cdot NP + m \cdot NP \cdot D = (D + 2 + m \cdot D) \cdot NP \quad (19)$$

Therefore, the time complexity of the proposed algorithm is $O(NP \cdot D)$.

5. COMPARATIVE STUDIES OF EXPERIMENTS

A. Benchmark Functions

In this section, 20 widely used benchmark functions from CEC2013 [20] test suite are employed to verify the effectiveness of the proposed algorithm against other state-of-the-art algorithms. These functions can be divided into two groups. This first group

includes F_1 – F_{10} which are simple and low-dimensional multimodal problems. F_1 – F_5 have a small number of global optima, and F_6 – F_{10} have a large number of global optima. The second group includes F_{11} – F_{20} which are composition multimodal problems composed of several basic problems with different characteristics. These benchmark functions show some characteristics like uneven landscape, multiple global optima and local optima, unequal spacing among optima, etc. The properties of these functions are given in Table 1. The meaning of each column is as follows. r denotes the niche radius. D denotes the dimension of the test function. The fourth column denotes the number of global optimal solutions for each test function. In order to ensure a fair comparison between the compared algorithms, the algorithm will terminate when the number of function evaluations reaches the specified threshold. *MaxFES* denotes the values of the maximum number of function evaluations (MaxFES) for each compared algorithm. The last column represents the population size for each function to be optimized.

Some functions from the CEC2013 are drawn in the following. Five-Uneven-Peak Trap (F_1) has 2 global optima and 3 local optima, as shown in Figure 1 (a). Six-Hump Camel Back function (F_5) has 2 global optima and 2 local optima, as shown in Figure 1 (b). Figure 1 (c) shows an example of the Shubert 2D function (F_6), where there are 18 global optima in 9 pairs. Figure 1 (d) shows the 2D version of CF_2 (F_{12}). Composition Function 2 (CF_2) is constructed based on eight basic functions, therefore it has eight global optima. The basic functions used in CF_2 include Rastrigin's function, Weierstrass function, Griewank's function, and Sphere function.

B. Algorithms Compared

In this paper, 13 algorithms are selected to compare with the proposed algorithm. These algorithms can be divided into four categories. CDE, FERPSO, $r2$ PSO, $r3$ PSO, $r2$ PSO-*lhc*, $r3$ PSO-*lhc*, NCDE, SCMA-ES, and MA-ESN belong to the category of niching without niching parameters. SDE and NSDE belong to the category of speciation. NShDE belongs to the category of fitness

Table 1 | Parameter setting for test functions.

Fun.	r	D	Number of Global Optima	MaxFES	Population Size
F_1	0.01	1	2	5E+04	80
F_2	0.01	1	5	5E+04	80
F_3	0.01	1	1	5E+04	80
F_4	0.01	2	4	5E+04	80
F_5	0.5	2	2	5E+04	80
F_6	0.5	2	18	2E+05	100
F_7	0.2	2	36	2E+05	300
F_8	0.5	3	81	4E+05	300
F_9	0.2	3	216	4E+05	300
F_{10}	0.01	2	12	2E+05	100
F_{11}	0.01	2	6	2E+05	200
F_{12}	0.01	2	8	2E+05	200
F_{13}	0.01	2	6	2E+05	200
F_{14}	0.01	3	6	4E+05	200
F_{15}	0.01	3	8	4E+05	200
F_{16}	0.01	5	6	4E+05	200
F_{17}	0.01	5	8	4E+05	200
F_{18}	0.01	10	6	4E+05	400
F_{19}	0.01	10	8	4E+05	200
F_{20}	0.01	20	8	4E+05	400

MaxFES, maximum number of function evaluations.

Figure 1 | Four functions from the Congress on Evolutionary Computation 2013.

sharing. MOMMOP belongs to the category of using multiobjective optimization method for MMOPs. The state-of-the-art multimodal algorithms used for comparison with BMDE are shown as follows.

1. CDE [11]: Crowding DE.
2. SDE [2]: Species DE.
3. FERPSO [21]: fitness-Euclidean-distance ration PSO.
4. *r*2PSO [22]: *lbest* PSO that interacts to its immediate right member by using a ring topology.
5. *r*3PSO [22]: *lbest* PSO that interacts to its immediate left and right member by using a ring topology.
6. *r*2PSO-*lhc* [22]: *r*2PSO model without overlapping neighborhoods.
7. *r*3PSO-*lhc* [22]: *r*3PSO model without overlapping neighborhoods.
8. NSDE [2]: neighborhood based SDE.
9. NShDE [2]: neighborhood based sharing DE.
10. NCDE [2]: neighborhood based CDE.
11. MOMMOP [13]: multiobjective optimization for MMOPs
12. SCMA-ES [23]: self-adaptive niching CMA-ES
13. MA-ESN [24–25]: matrix adaptation evolution strategy with dynamic niching
14. BMDE: the proposed algorithm.

C. Experimental Platform and Parameter Setting

To obtain an unbiased comparison, all the experiments are carried out on the same machine with an Intel Core i7-3770 3.40 GHz CPU and 4 GB memory.

All the algorithms use the same termination criterion, i.e., the *MaxFES*. The settings of *MaxFES* for the test functions are shown in Table 1. Each algorithm is run 25 times for each test function. The population size of BMDE is described in Table 1. The population size of other algorithms agree well with the original papers. Other parameter settings of each algorithm are provided in Table 2.

Generally, there are five accuracy thresholds, $\epsilon = 1.0E-1$, $\epsilon = 1.0E-2$, $\epsilon = 1.0E-3$, $\epsilon = 1.0E-4$, and $\epsilon = 1.0E-5$. $\epsilon = 1.0E-1$ and $\epsilon = 1.0E-2$ are easily available. Therefore, the accuracy thresholds with $\epsilon = 1.0E-3$, $\epsilon = 1.0E-4$, and $\epsilon = 1.0E-5$ are selected in the experiments. In addition, two well-known criteria [11] called peak ratio (PR) and success rate (SR) are employed to measure the performance of different multimodal optimization algorithms for each function. PR denotes the average percentage of all known global optima found over all runs, while SR denotes the percentage of successfully detecting all global optima out of all runs for each function.

In view of statistics significance of the results, the Wilcoxon signed-rank test [35] at the 5% significance level is employed to compare BMDE with other compared algorithms. “ $\approx$ ”, “ $-$ ”, and “ $+$ ” are applied to express the performance of BMDE is similar to ($\approx$), worse than ($-$), and better than ($+$) that of the compared algorithm, respectively. Moreover, the Friedman’s test, which is implemented by using KEEL software [36], is used to determine the ranking of all compared algorithms.

Table 2 | Parameter setting of the algorithms.

Algorithm	Parameter Setting
CDE	Scaling factor $F = 0.9$, the probability of crossover $CR = 0.1$
SDE	Scaling factor $F = 0.9$, the probability of crossover $CR = 0.1$
FERPSO	Acceleration coefficient $c_1 = c_2 = 2.05$, inertia weight $w = 0.7298$
r2PSO	Acceleration coefficient $c_1 = c_2 = 2.05$, inertia weight $w = 0.7298$
r3PSO	Acceleration coefficient $c_1 = c_2 = 2.05$, inertia weight $w = 0.7298$
r2PSO-lhc	Acceleration coefficient $c_1 = c_2 = 2.05$, inertia weight $w = 0.7298$
r3PSO-lhc	Acceleration coefficient $c_1 = c_2 = 2.05$, inertia weight $w = 0.7298$
NSDE	Scaling factor $F = 0.9$, the probability of crossover $CR = 0.1$
NShDE	$F = 0.9$, $CR = 0.1$, archive size $Msize = 2 * NP$ (NP is the population size)
NCDE	Scaling factor $F = 0.9$, the probability of crossover $CR = 0.1$
MOMMOP	Scaling factor $F = 0.5$, the probability of crossover $CR = 0.7$
SCMA-ES	Candidate solutions $\lambda = 10$, $\mu_{eff} = 1$, step-size of the mutation $\sigma_0 = 0.25$
MA-ESN	Candidate solutions $\lambda = 10$, $\mu_{eff} = 1$, step-size of the mutation $\sigma_0 = 0.25$
BMDE	$F = 0.8$, $CR = 0.5$, archive size $Msize = 1.5 * NP$ (NP is the population size), $\sigma = 0.01$

BMDE, bi-population and multi-mutation differential evolution; NSDE, neighborhood-based SDE; CDE, crowding differential evolution; SDE, species differential evolution; FERPSO, fitness-Euclidean distance ration particle swarm optimization; PSO, particle swarm optimization; NShDE, neighborhood-based sharing DE; NCDE, neighborhood-based CDE; MOMMOP multiobjective optimization for MMOPs; SCMA-ES, self-adaptive niching CMA-ES; MA-ESN, matrix adaptation evolution strategy with dynamic niching.

D. Comparisons with State-of-the-Art Multimodal Algorithms

Tables 3–5 show the result obtained by BMDE, CDE, SDE, FERPSO, r2PSO, r3PSO, r2PSO-lhc, r3PSO-lhc, NSDE, NShDE, NCDE, MOMMOP, SCMA-ES, and MA-ESN on F_1 – F_{10} with respect to PR and SR at three accuracy levels, i.e., $\varepsilon = 1.0E-3$, $\varepsilon = 1.0E-4$, and $\varepsilon = 1.0E-5$. For the functions F_1 – F_{10} , as can be seen from Tables 3–5, BMDE can find all global solutions in a single run at three accuracy levels.

Table 3 shows that BMDE performs better than SDE, r2PSO, r2PSO-lhc on F_1 , F_4 , F_6 , F_7 , F_8 , F_9 and F_{10} . BMDE performs better than CDE on F_4 , F_6 , F_7 , F_8 , and F_9 . BMDE performs better than FERPSO and r3PSO-lhc on F_1 , F_6 , F_7 , F_8 , F_9 , and F_{10} . BMDE performs better than r3PSO on all problems except F_3 and F_5 . BMDE performs better than NSDE on F_4 , F_6 , F_7 , F_8 , F_9 and F_{10} . BMDE performs better than NShDE on F_6 , F_7 , F_8 , F_9 , and F_{10} . BMDE performs better than NCDE on F_6 , F_7 , F_8 , and F_9 . BMDE performs better than SCMA-ES and MA-ESN on all problems except F_1 . MOMMOP can find all global solutions in a single run at $\varepsilon = 1.0E-3$. BMDE outperforms CDE, SDE, FERPSO, r2PSO, r3PSO, r2PSO-lhc, r3PSO-lhc, NSDE, NShDE, NCDE, MOMMOP, SCMA-ES, and MA-ESN on 5, 7, 6, 6, 7, 6, 6, 5, 5, 2, 0, 9, 9 test problems, respectively.

Table 4 indicates that BMDE performs better than SDE, r2PSO, r2PSO-lhc, r3PSO-lhc on F_1 , F_4 , F_6 , F_7 , F_8 , F_9 , and F_{10} . BMDE

performs better than CDE on F_4 , F_6 , F_7 , F_8 , and F_9 . BMDE performs better than FERPSO on F_1 , F_6 , F_7 , F_8 , F_9 , and F_{10} . BMDE performs better than r3PSO on all problems except F_3 and F_5 . BMDE performs better than NSDE on F_4 , F_6 , F_7 , F_8 , F_9 , and F_{10} . BMDE performs better than NShDE and NCDE on F_6 , F_7 , F_8 , F_9 , and F_{10} . BMDE performs better than MOMMOP on F_4 and F_9 . BMDE performs better than SCMA-ES and MA-ESN on all problems except F_1 . BMDE outperforms CDE, SDE, FERPSO, r2PSO, r3PSO, r2PSO-lhc, r3PSO-lhc, NSDE, NShDE, NCDE, MOMMOP, SCMA-ES, and MA-ESN on 5, 7, 6, 7, 7, 7, 6, 6, 5, 3, 0, 9, 9 test problems, respectively.

Table 5 indicates that BMDE performs better than SDE, FERPSO, r2PSO, r2PSO-lhc, on F_1 , F_4 , F_6 , F_7 , F_8 , F_9 , and F_{10} . BMDE performs better than CDE on F_4 , F_6 , F_7 , F_8 , and F_9 . BMDE performs better than r3PSO on all problems except F_3 and F_5 . BMDE performs better than r3PSO-lhc on all problems except F_2 and F_3 . BMDE performs better than NSDE on F_4 , F_5 , F_6 , F_7 , F_8 , F_9 , and F_{10} . BMDE performs better than NShDE and NCDE on F_6 , F_7 , F_8 , F_9 , and F_{10} . BMDE performs better than MOMMOP on F_4 and F_9 . BMDE performs better than SCMA-ES and MA-ESN on all problems except F_1 . BMDE outperforms CDE, SDE, FERPSO, r2PSO, r3PSO, r2PSO-lhc, r3PSO-lhc, NSDE, NShDE, NCDE, MOMMOP, SCMA-ES, and MA-ESN on 5, 7, 6, 7, 7, 7, 7, 6, 5, 4, 1, 9, 9 test problems, respectively.

Tables 6–8 shows the result obtained by BMDE, CDE, SDE, FERPSO, r2PSO, r3PSO, r2PSO-lhc, r3PSO-lhc, NSDE, NShDE, NCDE, MOMMOP, SCMA-ES, and MA-ESN on F_{11} – F_{20} with respect to PR and SR at $\varepsilon = 1.0E-3$, $\varepsilon = 1.0E-4$, and $\varepsilon = 1.0E-5$, respectively. As can be seen in Tables 6–8, BMDE performs better than CDE, SDE, r2PSO, r3PSO, r2PSO-lhc, r3PSO-lhc, and NSDE on all problems at $\varepsilon = 1.0E-3$, $\varepsilon = 1.0E-4$, and $\varepsilon = 1.0E-5$, respectively. BMDE performs better than FERPSO on all problems except F_{12} at three accuracy levels. BMDE performs better than NShDE on F_{11} , F_{12} , F_{13} , F_{15} , F_{16} , F_{18} , and F_{19} at $\varepsilon = 1.0E-5$. BMDE performs better than NCDE on all problems except F_{20} at $\varepsilon = 1.0E-4$ and $\varepsilon = 1.0E-5$, respectively. BMDE performs better than MOMMOP on F_{11} , F_{13} , and F_{16} at $\varepsilon = 1.0E-4$ and $\varepsilon = 1.0E-5$, respectively. BMDE performs better than SCMA-ES and MA-ESN on all problems except F_{19} at three accuracy levels.

BMDE outperforms CDE, SDE, FERPSO, r2PSO, r3PSO, r2PSO-lhc, r3PSO-lhc, NSDE, NShDE, NCDE, MOMMOP, SCMA-ES, and MA-ESN on 10, 10, 8, 10, 10, 10, 10, 10, 4, 4, 1, 9, 9 test problems at $\varepsilon = 1.0E-3$, respectively. BMDE outperforms CDE, SDE, FERPSO, r2PSO, r3PSO, r2PSO-lhc, r3PSO-lhc, NSDE, NShDE, NCDE, MOMMOP, SCMA-ES, and MA-ESN on 10, 10, 8, 10, 10, 10, 10, 10, 4, 6, 2, 9, 9 test problems at $\varepsilon = 1.0E-4$, respectively. BMDE outperforms CDE, SDE, FERPSO, r2PSO, r3PSO, r2PSO-lhc, r3PSO-lhc, NSDE, NShDE, NCDE, MOMMOP, SCMA-ES, and MA-ESN on 10, 10, 8, 10, 10, 10, 10, 10, 4, 7, 2, 9, 9 test problems at $\varepsilon = 1.0E-5$, respectively.

Tables 3–8 show that the number of “+ (better)” obtained by the proposed algorithm is bigger than most compared algorithms except MOMMOP. It was experimental shown that BMDE ranks first among all algorithms when accuracy level $\varepsilon = 1.0E-5$, while BMDE ranks second among all algorithms when accuracy level $\varepsilon = 1.0E-3$ and $\varepsilon = 1.0E-4$. Apart from MOMMOP, the average excellent and good rate of BMDE for 20 functions is 73.75%

Table 3 Experimental results in PR and SR on problems F_1 – F_{10} at accuracy level $\varepsilon = 1.0\text{E}-3$.

Func	BMDE		CDE		SDE		FERPSO		r2PSO		r3PSO		r2PSO-lhc	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
F_1	1	1.000	1(≈)	1.000	0.78(+)	0.560	0(+)	0.000	0(+)	0.000	0(+)	0.000	0(+)	0.000
F_2	1	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	0.99(≈)	0.960	1(≈)	1.000
F_3	1	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000
F_4	1	1.000	0.75(+)	0.280	0.25(+)	0.000	1(≈)	1.000	0.97(≈)	0.880	0.96(+)	0.840	0.99(≈)	0.960
F_5	1	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000
F_6	1	1.000	0.95(+)	0.440	0.85(+)	0.080	0.88(+)	0.160	0.53(+)	0.000	0.63(+)	0.000	0.63(+)	0.000
F_7	1	1.000	0.93(+)	0.040	0.61(+)	0.000	0.41(+)	0.000	0.53(+)	0.000	0.50(+)	0.000	0.55(+)	0.000
F_8	1	1.000	0.49(+)	0.000	0.16(+)	0.000	0.02(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000
F_9	1	1.000	0.70(+)	0.000	0.23(+)	0.000	0.20(+)	0.000	0.17(+)	0.000	0.19(+)	0.000	0.19(+)	0.000
F_{10}	1	1.000	1(≈)	1.000	0.43(+)	0.000	0.34(+)	0.000	0.89(+)	0.280	0.91(+)	0.400	0.93(+)	0.400
+(BMDE is better)			5		7		6		6		7		6	
– (BMDE is worse)			0		0		0		0		0		0	
≈ (BMDE is similar)			5		3		4		4		3		4	

Func	r3PSO-lhc		NSDE		NShDE		NCDE		MOMMOP		SCMA-ES		MA-ESN	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
F_1	0(+)	0.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000
F_2	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	0.92(+)	0.720	0.88(+)	0.480
F_3	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	0.84(+)	0.840	0.72(+)	0.720
F_4	1(≈)	1.000	0.96(≈)	0.880	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	0.66(+)	0.040	0.64(+)	0.120
F_5	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	0.84(+)	0.720	0.84(+)	0.720
F_6	0.67(+)	0.000	0.01(+)	0.000	0.99(+)	0.840	0.99(≈)	0.960	1(≈)	1.000	0.75(+)	0.000	0.76(+)	0.000
F_7	0.53(+)	0.000	0.64(+)	0.000	0.98(+)	0.640	0.94(+)	0.000	1(≈)	1.000	0.79(+)	0.000	0.81(+)	0.000
F_8	0.00(+)	0.000	0.00(+)	0.000	0.89(+)	0.000	0.99(≈)	0.960	1(≈)	1.000	0.53(+)	0.000	0.52(+)	0.000
F_9	0.19(+)	0.000	0.26(+)	0.000	0.79(+)	0.000	0.66(+)	0.000	1(≈)	1.000	0.14(+)	0.000	0.14(+)	0.000
F_{10}	0.93(+)	0.360	0.98(+)	0.800	0.94(+)	0.360	1(≈)	1.000	1(≈)	1.000	0.84(+)	0.080	0.85(+)	0.080
+		6	5		5		2		0		9		9	
–		0	0		0		0		0		0		0	
≈		4	5		5		8		10		1		1	

BMDE, bi-population and multi-mutation differential evolution; NSDE, neighborhood-based SDE; CDE, crowding differential evolution; SDE, species differential evolution; FERPSO, fitness-Euclidean distance ration particle swarm optimization; PSO, particle swarm optimization; NShDE, neighborhood-based sharing DE; NCDE, neighborhood-based CDE; MOMMOP multiobjective optimization for MMOPs; SCMA-ES, self-adaptive niching CMA-ES; MA-ESN, matrix adaptation evolution strategy with dynamic niching; PR, peak ratio; SR, success rate.

$\left(\sum_{i=1}^{20} b_i / (20 \times 12)\right)$ when $\varepsilon = 1.0\text{E}-3$, the average excellent and good rate of BMDE is 76.25% $\left(\sum_{i=1}^{20} b_i / (20 \times 12)\right)$ when $\varepsilon = 1.0\text{E}-4$, and the average excellent and good rate of BMDE is 77.50% $\left(\sum_{i=1}^{20} b_i / (20 \times 12)\right)$ when $\varepsilon = 1.0\text{E}-5$.

In order to test the statistical significance of the 14 compared algorithms, the Wilcoxon's test at the 5% significance level, which is implemented by using KEEL software [36], is employed based on the PR values. Tables 9–11 summarizes the statistical test results at three accuracy levels. It can be seen from Tables 9–11 that BMDE provides higher R+ values than R– values compared with CDE, SDE, FERPSO, r2PSO, r3PSO, r2PSO-lhc, r3PSO-lhc, NSDE, NShDE, NCDE, SCMA-ES, and MA-ESN at three accuracy levels. Furthermore, the p values of CDE, SDE, FERPSO, r2PSO, r3PSO, r2PSO-lhc, r3PSO-lhc, NSDE, SCMA-ES, and MA-ESN are less than 0.05 at $\varepsilon = 1.0\text{E}-3$, which means that BMDE is significantly better than these competitors. The p values of NShDE, NCDE, and MOMMOP are greater than 0.05, which means that the performance of BMDE is not different from that of NShDE, NCDE, and MOMMOP. Similarly, the p values of all compared algorithms are less than 0.05 at $\varepsilon = 1.0\text{E}-4$ except NShDE and MOMMOP. When $\varepsilon = 1.0\text{E}-5$, the p values of all compared algorithms are less than 0.05 except MOMMOP. Experimental results show that BMDE performs well on accuracy of solution.

To further determine the ranking of the 14 compared algorithms, the Friedman's test, which is also implemented by using KEEL software, is conducted. As shown in Table 12, the overall ranking sequences for the test problems are MOMMOP, BMDE, NShDE, NCDE, FERPSO, CDE, NSDE, MA-ESN, SCMA-ES, r2PSO-lhc, r3PSO-lhc, SDE, r2PSO, r3PSO at $\varepsilon = 1.0\text{E}-3$. Table 13 shows that the overall ranking sequences for the test problems are MOMMOP, BMDE, NShDE, NCDE, FERPSO, MA-ESN, SCMA-ES, SDE, NSDE, CDE, r3PSO-lhc, r2PSO-lhc, r3PSO, r2PSO at $\varepsilon = 1.0\text{E}-4$. Table 14 shows that the overall ranking sequences for the test problems are BMDE, MOMMOP, NShDE, NCDE, FERPSO, SCMA-ES, MA-ESN, SDE, CDE, r3PSO-lhc, r3PSO, NSDE(r2PSO-lhc), r2PSO at $\varepsilon = 1.0\text{E}-5$. The average rank of NSDE is the same as that of r2PSO-lhc. In general, the ranking value of BMDE and MOMMOP are approximately equal. Therefore, it can be concluded that the evolution strategies used in BMDE are effective.

In order to intuitively display the number of global optimal solutions found by each algorithm, Figures 2–4 show the results of average number of peaks (ANP) obtained in 25 independent runs by each algorithm for function F_1 – F_{20} at $\varepsilon = 1.0\text{E}-3$, $\varepsilon = 1.0\text{E}-4$, and $\varepsilon = 1.0\text{E}-5$, respectively. ANP denotes the average number of peaks found by an algorithm over all runs. In addition, in order to make Figures 2–4 clearer, r2PSO-lhc, r3PSO-lhc, SCMA-ES are denoted as r2-lhc, r3-lhc, and SCMA for short, respectively.

Table 4 | Experimental results in PR and SR on problems F_1 – F_{10} at accuracy level $\varepsilon = 1.0\text{E}-4$.

Func	BMDE		CDE		SDE		FERPSO		r2PSO		r3PSO		r2PSO-lhc	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
F_1	1	1.000	1(≈)	1.000	0.78(+)	0.560	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000
F_2	1	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	0.99(≈)	0.960	1(≈)	1.000
F_3	1	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000
F_4	1	1.000	0.22(+)	0.040	0.25(+)	0.000	1(≈)	1.000	0.89(+)	0.600	0.92(+)	0.680	0.94(+)	0.800
F_5	1	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000
F_6	1	1.000	0.49(+)	0.000	0.83(+)	0.04	0.85(+)	0.04	0.35(+)	0.000	0.48(+)	0.000	0.50(+)	0.000
F_7	1	1.000	0.92(+)	0.000	0.61(+)	0.000	0.41(+)	0.000	0.49(+)	0.000	0.47(+)	0.000	0.51(+)	0.000
F_8	1	1.000	0.05(+)	0.000	0.16(+)	0.000	0.01(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000
F_9	1	1.000	0.69(+)	0.000	0.23(+)	0.000	0.18(+)	0.000	0.08(+)	0.000	0.09(+)	0.000	0.09(+)	0.000
F_{10}	1	1.000	1(≈)	1.000	0.43(+)	0.000	0.33(+)	0.000	0.80(+)	0.080	0.85(+)	0.240	0.84(+)	0.080
+(BMDE is better)			5		7		6		7		7		7	
– (BMDE is worse)			0		0		0		0		0		0	
≈ (BMDE is similar)			5		3		4		3		3		3	
Func	r3PSO-lhc		NSDE		NShDE		NCDE		MOMMOP		SCMA-ES		MA-ESN	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
F_1	0.00(+)	0.00	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000
F_2	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	0.80(+)	0.320	0.66(+)	0.040
F_3	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	0.80(+)	0.800	0.56(+)	0.560
F_4	0.98(≈)	0.920	0.91(+)	0.680	1(≈)	1.000	1(≈)	1.000	0.99(≈)	0.960	0.63(+)	0.000	0.64(+)	0.120
F_5	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	0.78(+)	0.640	0.80(+)	0.680
F_6	0.52(+)	0.000	0.00(+)	0.000	0.79(+)	0.040	0.98(+)	0.760	1(≈)	1.000	0.75(+)	0.000	0.76(+)	0.000
F_7	0.50(+)	0.000	0.49(+)	0.000	0.98(+)	0.640	0.94(+)	0.000	1(≈)	1.000	0.79(+)	0.000	0.80(+)	0.000
F_8	0.00(+)	0.000	0.00(+)	0.000	0.62(+)	0.000	0.98(≈)	0.920	1(≈)	1.000	0.53(+)	0.000	0.52(+)	0.000
F_9	0.09(+)	0.000	0.13(+)	0.000	0.78(+)	0.000	0.66(+)	0.000	0.99(≈)	0.920	0.14(+)	0.000	0.14(+)	0.000
F_{10}	0.88(+)	0.240	0.89(+)	0.280	0.94(+)	0.360	0.99(≈)	0.960	1(≈)	1.000	0.84(+)	0.080	0.85(+)	0.080
+	6		6		5		3		0		9		9	
–	0		0		0		0		0		0		0	
≈	4		4		5		7		10		1		1	

Table 5 | Experimental results in PR and SR on problems F_1 – F_{10} at accuracy level $\varepsilon = 1.0\text{E}-5$.

Func	BMDE		CDE		SDE		FERPSO		r2PSO		r3PSO		r2PSO-lhc	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
F_1	1	1.000	1(≈)	1.000	0.78(+)	0.560	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000
F_2	1	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	0.99(≈)	0.960	1(≈)	1.000
F_3	1	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000
F_4	1	1.000	0.02(+)	0.000	0.25(+)	0.000	0.99(≈)	0.960	0.70(+)	0.240	0.78(+)	0.360	0.86(+)	0.560
F_5	1	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000
F_6	1	1.000	0.10(+)	0.000	0.82(+)	0.04	0.79(+)	0.000	0.21(+)	0.000	0.37(+)	0.000	0.46(+)	0.000
F_7	1	1.000	0.80(+)	0.000	0.61(+)	0.000	0.39(+)	0.000	0.43(+)	0.000	0.44(+)	0.000	0.00(+)	0.000
F_8	1	1.000	0.00(+)	0.000	0.16(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000
F_9	1	1.000	0.58(+)	0.000	0.23(+)	0.000	0.14(+)	0.000	0.03(+)	0.000	0.03(+)	0.000	0.03(+)	0.000
F_{10}	1	1.000	1(≈)	1.000	0.43(+)	0.000	0.32(+)	0.000	0.65(+)	0.000	0.76(+)	0.120	0.74(+)	0.000
+(BMDE is better)			5		7		6		7		7		7	
– (BMDE is worse)			0		0		0		0		0		0	
≈ (BMDE is similar)			5		3		4		3		3		3	
Func	r3PSO-lhc		NSDE		NShDE		NCDE		MOMMOP		SCMA-ES		MA-ESN	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
F_1	0.00(+)	0.00	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000
F_2	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	0.65(+)	0.160	0.44(+)	0.000
F_3	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	0.72(+)	0.720	0.32(+)	0.320
F_4	0.85(+)	0.520	0.87(+)	0.560	1(≈)	1.000	1(≈)	1.000	0.97(≈)	0.880	0.62(+)	0.000	0.60(+)	0.080
F_5	0.98(≈)	0.960	0.98(≈)	0.960	1(≈)	1.000	1(≈)	1.000	1(≈)	1.000	0.74(+)	0.600	0.78(+)	0.640
F_6	0.41(+)	0.000	0.00(+)	0.000	0.33(+)	0.000	0.91(+)	0.200	1(≈)	1.000	0.75(+)	0.000	0.76(+)	0.000
F_7	0.46(+)	0.000	0.35(+)	0.000	0.98(+)	0.560	0.94(+)	0.000	1(≈)	1.000	0.79(+)	0.000	0.80(+)	0.000
F_8	0.00(+)	0.000	0.00(+)	0.000	0.28(+)	0.000	0.98(+)	0.480	1(≈)	1.000	0.53(+)	0.000	0.52(+)	0.000
F_9	0.03(+)	0.000	0.07(+)	0.000	0.77(+)	0.000	0.65(+)	0.000	0.98(+)	0.000	0.14(+)	0.000	0.14(+)	0.000
F_{10}	0.80(+)	0.040	0.75(+)	0.040	0.94(+)	0.360	0.99(≈)	0.960	1(≈)	1.000	0.84(+)	0.080	0.85(+)	0.080
+	7		6		5		4		1		9		9	
–	0		0		0		0		0		0		0	
≈	3		4		5		6		9		1		1	

Table 6 Experimental results in PR and SR on problems F_{11} – F_{20} at accuracy level $\varepsilon = 1.0\text{E}-3$.

Func	BMDE		CDE		SDE		FERPSO		r2PSO		r3PSO		r2PSO-lhc	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
F_{11}	0.90	0.400	0.16(+)	0.000	0.18(+)	0.000	0.66(+)	0.000	0.64(+)	0.000	0.66(+)	0.000	0.65(+)	0.000
F_{12}	0.62	0.000	0.01(+)	0.000	0.15(+)	0.000	0.67(–)	0.000	0.36(+)	0.000	0.38(+)	0.000	0.42(+)	0.000
F_{13}	0.68	0.000	0.12(+)	0.000	0.18(+)	0.000	0.66($\approx$)	0.000	0.64(+)	0.000	0.62(+)	0.000	0.66(+)	0.000
F_{14}	0.66	0.000	0.03(+)	0.000	0.14(+)	0.000	0.40(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.02(+)	0.000
F_{15}	0.43	0.000	0.04(+)	0.000	0.03(+)	0.000	0.23(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000
F_{16}	0.66	0.000	0.00(+)	0.000	0.00(+)	0.000	0.62(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000
F_{17}	0.25	0.000	0.00(+)	0.000	0.00(+)	0.000	0.05(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000
F_{18}	0.49	0.000	0.03(+)	0.000	0.00(+)	0.000	0.14(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000
F_{19}	0.12	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000
F_{20}	0.12	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000
+(BMDE is better)			10		10		8		10		10		10	
– (BMDE is worse)			0		0		1		0		0		0	
$\approx$ (BMDE is similar)			0		0		1		0		0		0	
Func	r3PSO-lhc		NSDE		NShDE		NCDE		MOMMOP		SCMA-ES		MA-ESN	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
F_{11}	0.64(+)	0.000	0.66(+)	0.000	0.66(+)	0.000	0.70(+)	0.000	0.98(–)	0.880	0.58(+)	0.000	0.59(+)	0.000
F_{12}	0.38(+)	0.000	0.03(+)	0.000	0.44(+)	0.000	0.33(+)	0.000	0.95(–)	0.600	0.35(+)	0.000	0.55(+)	0.000
F_{13}	0.64(+)	0.000	0.49(+)	0.000	0.66($\approx$)	0.000	0.66($\approx$)	0.000	0.66($\approx$)	0.000	0.58(+)	0.000	0.60(+)	0.000
F_{14}	0.02(+)	0.000	0.25(+)	0.000	0.66($\approx$)	0.000	0.66($\approx$)	0.000	0.66($\approx$)	0.000	0.24(+)	0.000	0.32(+)	0.000
F_{15}	0.00(+)	0.000	0.16(+)	0.000	0.36(+)	0.000	0.35(+)	0.000	0.63(–)	0.000	0.16(+)	0.000	0.16(+)	0.000
F_{16}	0.00(+)	0.000	0.02(+)	0.000	0.66($\approx$)	0.000	0.64($\approx$)	0.000	0.62(+)	0.000	0.20(+)	0.000	0.18(+)	0.000
F_{17}	0.00(+)	0.000	0.00(+)	0.000	0.27($\approx$)	0.000	0.24($\approx$)	0.000	0.54(–)	0.000	0.13(+)	0.000	0.12(+)	0.000
F_{18}	0.00(+)	0.000	0.00(+)	0.000	0.28(+)	0.000	0.30(+)	0.000	0.50($\approx$)	0.000	0.16(+)	0.000	0.16(+)	0.000
F_{19}	0.00(+)	0.000	0.00(+)	0.000	0.18(–)	0.000	0.16(–)	0.000	0.22(–)	0.000	0.12($\approx$)	0.000	0.12($\approx$)	0.000
F_{20}	0.00(+)	0.000	0.00(+)	0.000	0.25(–)	0.000	0.25(–)	0.000	0.12($\approx$)	0.000	0.00(+)	0.000	0.00(+)	0.000
+	10		10		4		4		1		9		9	
–	0		0		2		2		5		0		0	
$\approx$	0		0		4		4		4		1		1	

Table 7 Experimental results in PR and SR on problems F_{11} – F_{20} at accuracy level $\varepsilon = 1.0\text{E}-4$.

Func	BMDE		CDE		SDE		FERPSO		r2PSO		r3PSO		r2PSO-lhc	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
F_{11}	0.88	0.320	0.02(+)	0.000	0.18(+)	0.000	0.66(+)	0.000	0.57(+)	0.000	0.64(+)	0.000	0.61(+)	0.000
F_{12}	0.52	0.000	0.00(+)	0.000	0.18(+)	0.000	0.60(–)	0.000	0.29(+)	0.000	0.33(+)	0.000	0.36(+)	0.000
F_{13}	0.68	0.000	0.02(+)	0.000	0.15(+)	0.000	0.60($\approx$)	0.000	0.59(+)	0.000	0.60(+)	0.000	0.60(+)	0.000
F_{14}	0.66	0.000	0.00(+)	0.000	0.18(+)	0.000	0.39(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.01(+)	0.000
F_{15}	0.40	0.000	0.00(+)	0.000	0.14(+)	0.000	0.23(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000
F_{16}	0.66	0.000	0.00(+)	0.000	0.03(+)	0.000	0.60(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000
F_{17}	0.25	0.000	0.00(+)	0.000	0.00(+)	0.000	0.05(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000
F_{18}	0.42	0.000	0.00(+)	0.000	0.00(+)	0.000	0.13(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000
F_{19}	0.12	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000
F_{20}	0.12	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000
+(BMDE is better)			10		10		8		10		10		10	
– (BMDE is worse)			0		0		1		0		0		0	
$\approx$ (BMDE is similar)			0		0		1		0		0		0	
Func	r3PSO-lhc		NSDE		NShDE		NCDE		MOMMOP		SCMA-ES		MA-ESN	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
F_{11}	0.64(+)	0.000	0.66(+)	0.000	0.66(+)	0.000	0.68(+)	0.000	0.70(+)	0.000	0.58(+)	0.000	0.58(+)	0.000
F_{12}	0.32(+)	0.000	0.03(+)	0.000	0.38(+)	0.000	0.21(+)	0.000	0.92(–)	0.440	0.34(+)	0.000	0.41(+)	0.000
F_{13}	0.60(+)	0.000	0.46(+)	0.000	0.66($\approx$)	0.000	0.66($\approx$)	0.000	0.66($\approx$)	0.000	0.58(+)	0.000	0.60(+)	0.000
F_{14}	0.00(+)	0.000	0.16(+)	0.000	0.66($\approx$)	0.000	0.66($\approx$)	0.000	0.66($\approx$)	0.000	0.24(+)	0.000	0.32(+)	0.000
F_{15}	0.00(+)	0.000	0.14(+)	0.000	0.36(+)	0.000	0.35(+)	0.000	0.59(–)	0.000	0.16(+)	0.000	0.16(+)	0.000
F_{16}	0.00(+)	0.000	0.00(+)	0.000	0.66($\approx$)	0.000	0.63(+)	0.000	0.62(+)	0.000	0.20(+)	0.000	0.18(+)	0.000
F_{17}	0.00(+)	0.000	0.00(+)	0.000	0.26($\approx$)	0.000	0.23(+)	0.000	0.52(–)	0.000	0.13(+)	0.000	0.12(+)	0.000
F_{18}	0.00(+)	0.000	0.00(+)	0.000	0.24(+)	0.000	0.26(+)	0.000	0.50(–)	0.000	0.16(+)	0.000	0.16(+)	0.000
F_{19}	0.00(+)	0.000	0.00(+)	0.000	0.15(–)	0.000	0.12($\approx$)	0.000	0.22(–)	0.000	0.12($\approx$)	0.000	0.12($\approx$)	0.000
F_{20}	0.00(+)	0.000	0.00(+)	0.000	0.24(–)	0.000	0.25(–)	0.000	0.12($\approx$)	0.000	0.00(+)	0.000	0.00(+)	0.000
+	10		10		4		6		2		9		9	
–	0		0		2		1		5		0		0	
$\approx$	0		0		4		3		3		1		1	

Table 8 Experimental results in PR and SR on problems F_{11} – F_{20} at accuracy level $\varepsilon = 1.0\text{E}-5$.

Func	BMDE		CDE		SDE		FERPSO		r2PSO		r3PSO		r2PSO- <i>lhc</i>	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
F_{11}	0.86	0.240	0.00(+)	0.000	0.18(+)	0.000	0.66(+)	0.000	0.45(+)	0.000	0.58(+)	0.000	0.56(+)	0.000
F_{12}	0.44	0.000	0.00(+)	0.000	0.15(+)	0.000	0.64(–)	0.000	0.25(+)	0.000	0.29(+)	0.000	0.26(+)	0.000
F_{13}	0.68	0.000	0.00(+)	0.000	0.18(+)	0.000	0.65(≈)	0.000	0.43(+)	0.000	0.58(+)	0.000	0.50(+)	0.000
F_{14}	0.66	0.000	0.00(+)	0.000	0.14(+)	0.000	0.38(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000
F_{15}	0.40	0.000	0.00(+)	0.000	0.03(+)	0.000	0.23(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000
F_{16}	0.66	0.000	0.00(+)	0.000	0.00(+)	0.000	0.58(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000
F_{17}	0.25	0.000	0.00(+)	0.000	0.00(+)	0.000	0.05(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000
F_{18}	0.37	0.000	0.00(+)	0.000	0.00(+)	0.000	0.05(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000
F_{19}	0.12	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000
F_{20}	0.12	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000	0.00(+)	0.000
+(BMDE is better)			10		10		8		10		10		10	
–(BMDE is worse)			0		0		1		0		0		0	
≈(BMDE is similar)			0		0		1		0		0		0	
Func	r3PSO- <i>lhc</i>		NSDE		NShDE		NCDE		MOMMOP		SCMA-ES		MA-ESN	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
F_{11}	0.58(+)	0.000	0.65(+)	0.000	0.66(+)	0.000	0.68(+)	0.000	0.66(+)	0.000	0.57(+)	0.000	0.56(+)	0.000
F_{12}	0.28(+)	0.000	0.02(+)	0.000	0.34(+)	0.000	0.16(+)	0.000	0.81(–)	0.080	0.33(+)	0.000	0.27(+)	0.000
F_{13}	0.56(+)	0.000	0.44(+)	0.000	0.66(≈)	0.000	0.66(≈)	0.000	0.66(≈)	0.000	0.58(+)	0.000	0.59(+)	0.000
F_{14}	0.00(+)	0.000	0.12(+)	0.000	0.66(≈)	0.000	0.66(≈)	0.000	0.66(≈)	0.000	0.24(+)	0.000	0.32(+)	0.000
F_{15}	0.00(+)	0.000	0.08(+)	0.000	0.36(+)	0.000	0.35(+)	0.000	0.59(–)	0.000	0.16(+)	0.000	0.16(+)	0.000
F_{16}	0.00(+)	0.000	0.00(+)	0.000	0.66(≈)	0.000	0.60(+)	0.000	0.62(+)	0.000	0.20(+)	0.000	0.18(+)	0.000
F_{17}	0.00(+)	0.000	0.00(+)	0.000	0.26(≈)	0.000	0.22(+)	0.000	0.47(–)	0.000	0.13(+)	0.000	0.12(+)	0.000
F_{18}	0.00(+)	0.000	0.00(+)	0.000	0.20(+)	0.000	0.26(+)	0.000	0.50(–)	0.000	0.16(+)	0.000	0.16(+)	0.000
F_{19}	0.00(+)	0.000	0.00(+)	0.000	0.12(≈)	0.000	0.07(+)	0.000	0.22(–)	0.000	0.12(≈)	0.000	0.12(≈)	0.000
F_{20}	0.00(+)	0.000	0.00(+)	0.000	0.24(–)	0.000	0.25(–)	0.000	0.12(≈)	0.000	0(+)	0.000	0.00(+)	0.000
+		10	10		4		7		2		9		9	
–		0	0		1		1		5		0		0	
≈		0	0		5		2		3		1		1	

Table 9 Results obtained by the Wilcoxon test for algorithm BMDE at accuracy level $\varepsilon = 1.0\text{E}-3$.

VS	R ⁺	R [–]	Exact P-Value	Asymptotic P-Value
CDE	185.0	5.0	≥0.2	0.000234
SDE	188.5	1.5	≥0.2	0.000144
FERPSO	198.0	12.0	≥0.2	0.000375
r2PSO	188.5	1.5	≥0.2	0.000104
r3PSO	208.5	1.5	≥0.2	0.000076
r2PSO- <i>lhc</i>	188.5	1.5	≥0.2	0.000123
r3PSO- <i>lhc</i>	205.0	5.0	≥0.2	0.000152
NSDE	205.0	5.0	≥0.2	0.000164
NShDE	144.0	45.5	≥0.2	0.039016
NCDE	152.0	37.5	≥0.2	0.016654
MOMMOP	68.0	142.0	≥0.2	1
SCMA-ES	208.5	1.5	≥0.2	0.000065
MA-ESN	208.5	1.5	≥0.2	0.000096

NSDE, neighborhood-based SDE; CDE, crowding differential evolution; SDE, species differential evolution; FERPSO, fitness-Euclidean distance ration particle swarm optimization; PSO, particle swarm optimization; NShDE, neighborhood-based sharing DE; NCDE, neighborhood-based CDE; MOMMOP multiobjective optimization for MMOPs; SCMA-ES, self-adaptive niching CMA-ES; MA-ESN, matrix adaptation evolution strategy with dynamic niching.

Table 10 Average ranking of the algorithms (Friedman) at accuracy level $\varepsilon = 1.0\text{E}-3$.

Algorithm	Ranking
BMDE	3.05
CDE	8.475
SDE	9.85
FERPSO	7.275
r2PSO	10.175
r3PSO	10.2
r2PSO- <i>lhc</i>	9.25
r3PSO- <i>lhc</i>	9.4
NSDE	8.65
NShDE	3.875
NCDE	4.2
MOMMOP	2.8
SCMA-ES	9.075
MA-ESN	8.725

BMDE, bi-population and multi-mutation differential evolution; NSDE, neighborhood-based SDE; CDE, crowding differential evolution; SDE, species differential evolution; FERPSO, fitness-Euclidean distance ration particle swarm optimization; PSO, particle swarm optimization; NShDE, neighborhood-based sharing DE; NCDE, neighborhood-based CDE; MOMMOP multiobjective optimization for MMOPs; SCMA-ES, self-adaptive niching CMA-ES; MA-ESN, matrix adaptation evolution strategy with dynamic niching.

6. CONCLUSION

This paper presents an enhanced DE with bi-population and multi-mutation strategy for multimodal optimization problems. In the proposed algorithm, the advantages of different mutation strategies and multi-population are embedded into DE algorithm. Firstly, the proposed algorithm employs two different mutation strategies

to explore and exploit in parallel to find multiple optimal solutions. Furthermore, the differences between the inferior individuals and the current individuals are employed to provide a promising direction toward the optimum. Finally, individuals with high similarity are updated to improve population diversity. The experimental results suggest that BMDE can achieve a better and more

Table 11 | Results obtained by the Wilcoxon test for algorithm BMDE at accuracy level $\varepsilon = 1.0E-4$.

VS	R ⁺	R ⁻	Exact P-Value	Asymptotic P-Value
CDE	185.0	5.0	≥ 0.2	0.000187
SDE	188.5	1.5	≥ 0.2	0.000144
FERPSO	199.0	11.0	≥ 0.2	0.000393
r2PSO	188.5	1.5	≥ 0.2	0.000123
r3PSO	208.5	1.5	≥ 0.2	0.000082
r2PSO-lhc	188.5	1.5	≥ 0.2	0.000123
r3PSO-lhc	188.5	1.5	≥ 0.2	0.000096
NSDE	205.0	5.0	≥ 0.2	0.000152
NShDE	149.5	40.5	≥ 0.2	0.025364
NCDE	164.5	25.5	≥ 0.2	0.004043
MOMMOP	94.5	115.5	≥ 0.2	1
SCMA-ES	208.5	1.5	≥ 0.2	0.000089
MA-ESN	208.5	1.5	≥ 0.2	0.000096

BMDE, bi-population and multi-mutation differential evolution; NSDE, neighborhood-based SDE; CDE, crowding differential evolution; SDE, species differential evolution; FERPSO, fitness-Euclidean distance ration particle swarm optimization; PSO, particle swarm optimization; NShDE, neighborhood-based sharing DE; NCDE, neighborhood-based CDE; MOMMOP multiobjective optimization for MMOPs; SCMA-ES, self-adaptive niching CMA-ES; MA-ESN, matrix adaptation evolution strategy with dynamic niching.

Table 12 | Average ranking of the algorithms (Friedman) at accuracy level $\varepsilon = 1.0E-4$.

Algorithm	Ranking
BMDE	2.875
CDE	9.375
SDE	9.225
FERPSO	7.225
r2PSO	10.675
r3PSO	10.15
r2PSO-lhc	9.45
r3PSO-lhc	9.425
NSDE	9.35
NShDE	3.85
NCDE	4.2
MOMMOP	2.85
SCMA-ES	8.4
MA-ESN	7.95

BMDE, bi-population and multi-mutation differential evolution; NSDE, neighborhood-based SDE; CDE, crowding differential evolution; SDE, species differential evolution; FERPSO, fitness-Euclidean distance ration particle swarm optimization; PSO, particle swarm optimization; NShDE, neighborhood-based sharing DE; NCDE, neighborhood-based CDE; MOMMOP multiobjective optimization for MMOPs; SCMA-ES, self-adaptive niching CMA-ES; MA-ESN, matrix adaptation evolution strategy with dynamic niching.

Table 13 | Results obtained by the Wilcoxon test for algorithm BMDE at accuracy level $\varepsilon = 1.0E-5$.

VS	R ⁺	R ⁻	Exact P-Value	Asymptotic P-Value
CDE	185.0	5.0	≥ 0.2	0.000234
SDE	188.5	1.5	≥ 0.2	0.000133
FERPSO	178.0	12.0	≥ 0.2	0.000646
r2PSO	188.5	1.5	≥ 0.2	0.000123
r3PSO	208.5	1.5	≥ 0.2	0.000082
r2PSO-lhc	188.5	1.5	≥ 0.2	0.000096
r3PSO-lhc	208.5	1.5	≥ 0.2	0.000065
NSDE	208.5	1.5	≥ 0.2	0.000065
NShDE	168.0	42.0	≥ 0.2	0.017231
NCDE	182.5	27.5	≥ 0.2	0.003289
MOMMOP	95.5	114.5	≥ 0.2	1
SCMA-ES	208.5	1.5	≥ 0.2	0.000096
MA-ESN	208.5	1.5	≥ 0.2	0.000103

BMDE, bi-population and multi-mutation differential evolution; NSDE, neighborhood-based SDE; CDE, crowding differential evolution; SDE, species differential evolution; FERPSO, fitness-Euclidean distance ration particle swarm optimization; PSO, particle swarm optimization; NShDE, neighborhood-based sharing DE; NCDE, neighborhood-based CDE; MOMMOP multiobjective optimization for MMOPs; SCMA-ES, self-adaptive niching CMA-ES; MA-ESN, matrix adaptation evolution strategy with dynamic niching.

Table 14 | Average ranking of the algorithms (Friedman) at accuracy level $\varepsilon = 1.0E-5$.

Algorithm	Ranking
BMDE	2.725
CDE	9.575
SDE	9.25
FERPSO	7.175
r2PSO	10.45
r3PSO	9.75
r2PSO-lhc	9.975
r3PSO-lhc	9.675
NSDE	9.975
NShDE	4.075
NCDE	4.125
MOMMOP	2.825
SCMA-ES	7.625
MA-ESN	7.8

BMDE, bi-population and multi-mutation differential evolution; NSDE, neighborhood-based SDE; CDE, crowding differential evolution; SDE, species differential evolution; FERPSO, fitness-Euclidean distance ration particle swarm optimization; PSO, particle swarm optimization; NShDE, neighborhood-based sharing DE; NCDE, neighborhood-based CDE; MOMMOP multiobjective optimization for MMOPs; SCMA-ES, self-adaptive niching CMA-ES; MA-ESN, matrix adaptation evolution strategy with dynamic niching.

Figure 2 | Box-plot of ANP found by BMDE, CDE, SDE, FERPSO, r2PSO, r3PSO, r2PSO-lhc, r3PSO-lhc, NSDE, NShDE, NCDE, MOMMOP, SCMA-ES, and MA-ESN on F_1 – F_{20} at accuracy level $\varepsilon = 1.0E-3$. ANP, average number of peaks; BMDE, bi-population and multi-mutation differential evolution; NSDE, neighborhood-based SDE; CDE, crowding differential evolution; SDE, species differential evolution; FERPSO, fitness-Euclidean distance ration particle swarm optimization; PSO, particle swarm optimization; NShDE, neighborhood-based sharing DE; NCDE, neighborhood-based CDE; MOMMOP multiobjective optimization for MMOPs; SCMA-ES, self-adaptive niching CMA-ES; MA-ESN, matrix adaptation evolution strategy with dynamic niching.

Figure 3 | Box-plot of ANP found by BMDE, CDE, SDE, FERPSO, r2PSO, r3PSO, r2PSO-lhc, r3PSO-lhc, NSDE, NShDE, NCDE, MOMMOP, SCMA-ES, and MA-ESN on F_1 – F_{20} at accuracy level $\varepsilon = 1.0\text{E}-4$. ANP, average number of peaks; BMDE, bi-population and multi-mutation differential evolution; NSDE, neighborhood-based SDE; CDE, crowding differential evolution; SDE, species differential evolution; FERPSO, fitness-Euclidean distance ratio particle swarm optimization; PSO, particle swarm optimization; NShDE, neighborhood-based sharing DE; NCDE, neighborhood-based CDE; MOMMOP multiobjective optimization for MMOPs; SCMA-ES, self-adaptive niching CMA-ES; MA-ESN, matrix adaptation evolution strategy with dynamic niching.

Figure 4 | Box-plot of ANP found by BMDE, CDE, SDE, FERPSO, r2PSO, r3PSO, r2PSO-lhc, r3PSO-lhc, NSDE, NShDE, NCDE, MOMMOP, SCMA-ES, and MA-ESN on F_1 – F_{20} at accuracy level $\varepsilon = 1.0\text{E}-5$. ANP, average number of peaks; BMDE, bi-population and multi-mutation differential evolution; NSDE, neighborhood-based SDE; CDE, crowding differential evolution; SDE, species differential evolution; FERPSO, fitness-Euclidean distance ratio particle swarm optimization; PSO, particle swarm optimization; NShDE, neighborhood-based sharing DE; NCDE, neighborhood-based CDE; MOMMOP multiobjective optimization for MMOPs; SCMA-ES, self-adaptive niching CMA-ES; MA-ESN, matrix adaptation evolution strategy with dynamic niching.

consistent performance than most multimodal optimization algorithms on CEC2013 test problems. Future work will improve the performance of BMDE to solve F_{11} – F_{20} effectively. In addition, we will extend the BMDE to solve the multiobjective optimization problems, constrained optimization problems, and real-world applications.

CONFLICT OF INTEREST

The authors have declared no conflicts of interest.

AUTHORS' CONTRIBUTIONS

All authors contributed to the work. All authors read and approved the final manuscript.

ACKNOWLEDGMENTS

This research is partly supported by the Doctoral Foundation of Xi'an University of Technology (112-451116017), National Natural Science Foundation of China under Project Code (61803301, 61773314), and the Scientific Research Foundation of the National University of Defense Technology (grant no. ZK18-03-43).

REFERENCES

- [1] B. Cao, J. Zhao, Y. Gu, Y. Ling, X. Ma, Applying graph-based differential grouping for multiobjective large-scale optimization, *Swarm Evol. Comput.* 53 (2020), 1–15.
- [2] B.Y. Qu, P.N. Suganthan, J.J. Liang, Differential Evolution with neighborhood mutation for multimodal optimization, *IEEE Trans. Evol. Comput.* 16 (2012), 601–614.
- [3] P. Haghbayan, H. Nezamabadi-Pour, S. Kamyab, A niche GSA method with nearest neighbor scheme for multimodal optimization, *Swarm Evol. Comput.* 35 (2017), 78–92.
- [4] R. Cheng, M. Li, K. Li, X. Yao, Evolutionary multiobjective optimization-based multimodal optimization: fitness landscape approximation and peak detection, *IEEE Trans. Evol. Comput.* 22 (2018), 692–706.
- [5] H. Garg, A hybrid GSA-GA algorithm for constrained optimization problems, *Inf. Sci.* 478 (2019), 499–523.
- [6] H. Garg, A hybrid PSO-GA algorithm for constrained optimization problems, *Appl. Math. Comput.* 274 (2016), 292–305.
- [7] Y.J. Gong, J. Zhang, Y. Zhou, Learning multimodal parameters: a bare-bones niching differential evolution approach, *IEEE Trans. Neural Netw. Learn. Syst.* 29 (2018), 2944–2959.
- [8] W. Dong, M. Zhou, Gaussian classifier-based evolutionary strategy for multimodal optimization, *IEEE Trans. Neural Netw. Learn. Syst.* 25 (2014), 1200–1216.
- [9] B.Y. Qu, J.J. Liang, P.N. Suganthan, Niching particle swarm optimization with local search for multi-modal optimization, *Inf. Sci.* 197 (2012), 131–143.
- [10] W. Gao, G.G. Yen, S. Liu, A cluster-based differential evolution with self-adaptive strategy for multimodal optimization, *IEEE Trans. Cybernetics.* 44 (2014), 1314–1327.
- [11] S. Kamyab, M. Eftekhari, Using a self-adaptive neighborhood scheme with crowding replacement memory in genetic algorithm for multimodal optimization, *Swarm Evol. Comput.* 12 (2013), 1–17.
- [12] P.S. Oliveto, D. Sudholt, C. Zarges, On the benefits and risks of using fitness sharing for multimodal optimisation, *Theor. Comput. Sci.* 773 (2019), 53–70.
- [13] Y. Wang, H.X. Li, G.G. Yen, W. Song, MOMMOP: multiobjective optimization for locating multiple optimal solutions of multimodal optimization problems, *IEEE Trans. Cybernetics.* 45 (2015), 830–843.
- [14] B.Y. Qu, P.N. Suganthan, S. Das, A distance-based locally informed particle swarm model for multimodal optimization, *IEEE Trans. Evol. Comput.* 17 (2013), 387–402.
- [15] W. Luo, X. Lin, T. Zhu, P. Xu, A clonal selection algorithm for dynamic multimodal function optimization, *Swarm Evol. Comput.* 50 (2019), 1–11.
- [16] B. Qu, C. Li, J. Liang, L. Yan, K. Yu, Y. Zhu, A self-organized speciation based multi-objective particle swarm optimizer for multimodal multi-objective problems, *Appl. Soft Comput.* 86 (2020), 1–13.
- [17] J. Yao, N. Kharm, P. Grogono, Bi-objective multipopulation genetic algorithm for multimodal function optimization, *IEEE Trans. Evol. Comput.* 14 (2010), 80–102.
- [18] J. Kennedy, K. Eberhart, Particle swarm optimization, in: *Proceedings of the IEEE International Conference on Neural Networks*, Perth, Australia, 27 November–1 December 1995, pp. 1942–1948.
- [19] J.Q. Zhang, A.C. Sanderson, JADE: JADE:adaptive differential evolution with optional external archive, *IEEE Trans. Evol. Comput.* 13 (2009), 945–958.
- [20] X. Li, A. Engelbrecht, M. Epitropakis, Benchmark functions for CEC 2013 special session and competition on niching methods for multimodal function optimization, Technical Report, Royal Melbourne Institute of Technology, 2013.
- [21] X. Li, A multimodal particle swarm optimizer based on fitness Euclidean-distance ratio, Genetic and Evolutionary Computation Conference, GECCO 2007, Proceedings, London, England, UK, July 7–11, ACM, 2007, pp. 78–85.
- [22] X. Li, Niching without niching parameters: particle swarm optimization using a ring topology, *IEEE Trans. Evol. Comput.* 14 (2010), 150–169.
- [23] O.M. Shir, M. Emmerich, T. Bäck, Adaptive Niche Radii and Niche shapes approaches for Niching with the CMA-ES, *Evol. Comput.* 18 (2010), 97–126.
- [24] H.G. Beyer, B. Sendhoff, Simplify your covariance matrix adaptation evolution strategy, *IEEE Trans. Evol. Comput.* 21 (2017), 746–759.
- [25] O.M. Shir, T. Back, Dynamic niching in evolution strategies with covariance matrix adaptation, *IEEE Congress Evol. Comput.* 3 (2005), 2584–2591.
- [26] R. Storn, K.V. Price, Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces, *J. Global Optimization.* 11 (1997), 341–359.
- [27] E.C. Osuna, D. Sudholt, Runtime analysis of crowding mechanisms for multimodal optimization, *IEEE Trans. Evol. Comput.* 24 (2020), 581–592.
- [28] S. Biswas, S. Kundu, S. Das, Inducing niching behavior in differential evolution through local information sharing, *IEEE Trans. Evol. Comput.* 19 (2015), 246–263.
- [29] Z.J. Wang, Z.H. Zhan, Y. Lin, W.J. Yu, H.Q. Yuan, T.L. Gu, S. Kwong, J. Zhang, Dual-strategy differential evolution with affinity propagation clustering for multimodal optimization problems, *IEEE Trans. Evol. Comput.* 22 (2018), 894–908.

- [30] S. Hui, P.N. Suganthan, Ensemble and arithmetic recombination-based speciation differential evolution for multimodal optimization, *IEEE Trans. Cybernetics*. 46 (2016), 64–74.
- [31] J.E. Fieldsend, Running up those hills: multi-modal search with the niching migratory multi-swarm optimizer, 2014 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2014.
- [32] A. Basak, S. Das, K.C. Tan, Multimodal optimization using a biobjective differential evolution algorithm enhanced with mean distance-based selection, *IEEE Trans. Evol. Comput.* 17 (2013), 666–685.
- [33] W. Li, Matrix adaptation evolution strategy with multi-objective optimization for multimodal optimization, *Algorithms*. 12 (2019), 1–23.
- [34] S. Das, P.N. Suganthan, Differential evolution: a survey of the state-of-the-art, *IEEE Trans. Evol. Comput.* 15 (2011), 4–31.
- [35] Y. Wang, Z.X. Cai, Q.F. Zhang, Differential evolution with composite trial vector generation strategies and control parameters, *IEEE Trans. Evol. Comp.* 15 (2011), 55–66.
- [36] J. Alcalá-Fdez, L. Sánchez, S. García, M.J. del Jesus, S. Ventura, J.M. Garrell, J. Otero, C. Romero, J. Bacardit, V.M. Rivas, J.C. Fernández, F. Herrera, KEEL: a software tool to assess evolutionary algorithms to data mining problems, *Soft Comput.* 13 (2009), 307–318.