Research Article

Enhancing the Performance of Biogeography-Based Optimization Using Multitopology and Quantitative Orthogonal Learning

Siao Wen, Jinfu Chen, Yinhong Li, Dongyuan Shi, and Xianzhong Duan

State Key Laboratory of Advanced Electromagnetic Engineering and Technology, School of Electrical and Electronic Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

Correspondence should be addressed to Jinfu Chen; chenjinfu@mail.hust.edu.cn

Received 18 April 2017; Revised 16 July 2017; Accepted 7 August 2017; Published 13 September 2017

Academic Editor: Thomas Hanne

Copyright © 2017 Siao Wen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Two defects of biogeography-based optimization (BBO) are found out by analyzing the characteristics of its dominant migration operator. One is that, due to global topology and direct-copying migration strategy, information in several good-quality habitats tends to be copied to the whole habitats rapidly, which would lead to premature convergence. The other is that the generated solutions by migration process are distributed only in some specific regions so that many other areas where competitive solutions may exist cannot be investigated. To remedy the former, a new migration operator precisely developed by modifying topology and copy mode is introduced to BBO. Additionally, diversity mechanism is proposed. To remedy the latter defect, quantitative orthogonal learning process accomplished based on space quantizing and orthogonal design is proposed. It aims to investigate the feasible region thoroughly so that more competitive solutions can be obtained. The effectiveness of the proposed approaches is verified on a set of benchmark functions with diverse characteristics. The experimental results reveal that the proposed method has merits regarding solution quality, convergence performance, and so on, compared with basic BBO, five BBO variant algorithms, seven orthogonal learning-based algorithms, and other non-OL-based evolutionary algorithms. The effects of each improved component are also analyzed.

1. Introduction

Optimization problems have widely existed in any areas of life (economic, engineering, medicine, business, urban planning, etc.) and they play a very vital role in both academic research field and industrial production. They tend to be more complicated with the unceasing progress of science and technology. To handle such challenging optimization problems, much effort has been devoted by researchers and different techniques have been proposed during the last several decades. The proposed methods can be categorized into two main groups: derivative-based algorithms and artificial intelligence methods. Derivative-based algorithms require objective functions to be smooth and differentiable. Due to this property, they are restricted to be applied to many complex optimization problems. Oppositely, artificial intelligence methods do not require certain properties to be satisfied. In

artificial intelligence methods, evolutionary algorithms (EAs) are popular and have been successfully applied to real-world problems. EAs mimic various social behaviors existing in nature to solve the optimization problems. Some popular EAs include genetic algorithm (GA) [1, 2], particle swarm optimization (PSO) [3, 4], artificial immune system (AIS) [5], differentiable evolution (DE) [6, 7], ant colony optimization (ACO) [8], artificial bee colony (ABC) [9, 10], and simulated annealing algorithm (SA) [11].

As an effective EA, biogeography-based optimization (BBO) [12] has shown excellent performance after being tested on various benchmark functions [13–15], such as simple structure, fewer parameters than many other EAs, and strong robustness against the variations of its algorithm-dependent parameters [12]. Due to its excellent performance, it has been successfully applied to a variety of real-world problems [16–20]. However, BBO has also been shown

to have certain weaknesses. Although BBO has excellent exploitation ability, it tends to fall into local optima. That is to say, BBO lacks exploration ability. As we know, for an optimization algorithm, exploration and exploitation are two important concepts in any optimization algorithms. Exploration is the ability to search for areas far from the current individuals in the search space while exploitation is the capacity to investigate the vicinities nearby the current solution. Obviously, they are two entirely different objectives, so developing an algorithm good on both is a challenging work. To remedy it, much effort has been made, and some variants of BBO are proposed. One of the most popular research hotspots focuses on modifying the migration process. Li et al. [21] proposed a perturb migration operator to create new individuals to update the target individual. Ma and Simon [22] proposed a blended migration operator that can make two parental habitats contribute different constant weighted features to a new feature of an offspring. Xiong et al. [23] proposed a polyphyletic migration operator to utilize as many as four habitats' features to construct a new solution vector, which can generate new features from more promising areas in the search space. Li and Yin [24] introduced a multiparent migration operator which adopts three consecutive individuals to generate three offspring according to their fitness value. From the experimental results of [21–24], it can be seen that a well-designed migration operator indeed contributes to improving the performance of BBO. Besides migration operator modification, some other researchers focus on hybridizing BBO with some operators in other EAs so as to make up for BBO's defects. BBO is hybridized with harmony search (HS) to obtain HSBBO in study [25], which aims to bring the exploration of HS into BBO. Study [26] combines DE with BBO to develop DBBO algorithm. Study [27] hybridizes opposition-based learning with BBO to obtain oppositional BBO (OBBO). All these algorithms have achieved excellent performance.

In this paper, both migration process modification and hybridization strategy are focused on. Two defects of BBO are found out by analyzing the characteristics of the migration operator, which is the main operator of BBO. One is that, due to global topology and direct-copying migration strategy, information in several good-quality habitats tends to be copied to the whole habitats rapidly, which would lead to premature convergence. The other is that the solutions generated by migration operator are distributed only in some specific regions so that other areas where competitive solutions may exist cannot be investigated. To remedy the former, a new migration operator precisely developed by taking topology modification and copy-mode improvement as two cut-in points is proposed. Diversity mechanism is also proposed for remedying it. To remedy the latter defect, quantitative orthogonal learning (QOL) process accomplished based on space quantizing and orthogonal design is adopted. It aims to investigate the solution space thoroughly so that more competitive solutions can be obtained.

The remainder of this paper is organized as follows. Section 2 briefly introduces the basic principle of BBO. The new proposed algorithm is described in Section 3. The experimental results conducted on various benchmark

functions are shown in Section 4 to verify the effectiveness of the algorithm. In Section 5, the effects of different improved components are analyzed. Section 6 is devoted to conclusions.

2. The Proposed Methodology

BBO is inspired by the equilibrium theory of island biogeography. Each individual in population is called a "habitat." The goodness of a habitat (i.e., candidate solution) is measured by the habitat suitability index (HSI) [12]. Habitats with higher HSI tend to have more species than those with lower HSI. In BBO, the immigration rate λ and emigration rate μ of each habitat are functions of the number of species in the habitat and they are calculated as follows:

$$\begin{aligned}\lambda_i &= \frac{I}{2} \left(\cos \left(\frac{\pi S_i}{S_{\max}} \right) + 1 \right), \quad i = 1, 2, \dots, N_{\text{pop}} \\ \mu_i &= \frac{E}{2} \left(-\cos \left(\frac{\pi S_i}{S_{\max}} \right) + 1 \right), \quad i = 1, 2, \dots, N_{\text{pop}},\end{aligned}\quad (1)$$

where I , E are maximum immigration and emigration rate, respectively, N_{pop} is the number of habitats, S_i is the number of species in habitat i , and $S_{\max}$ is the largest possible number of species that a habitat can support. From (1), it can be seen that a good habitat tends to have a large value of μ and a small value of λ , which enables poor habitats to get information from good habitats to improve their quality.

There are two main operators in BBO: (1) migration operator and (2) mutation operator. In migration process, based on λ and μ , information is probabilistically emigrated from the good habitats to the poor ones. The quality of those poor habitats is enhanced by accepting new information from high-quality habitats. Hence, both good and bad habitats can collaborate with each other and move together toward more promising areas in the search space and finally converge to an optimal solution. In mutation process, some variables in each habitat are possible to be replaced by new variables which are randomly generated in the entire solution space, which aims to keep population diversity.

3. The Proposed Methodology

In allusion to the defects of the migration operator in basic BBO, a new BBO variant based on multitopology migration and QOL, called MTQLBBO, is proposed. The key points of our methods are described in detail in this section.

3.1. Population Initialization. Assuming that the number of decision variables in H_i (population i) is n , the initialization of H_i is expressed as follows:

$$x_{i,j} = x_j^{\min} + (x_j^{\max} - x_j^{\min}) + L(n)|_j, \quad (2)$$

where $x_{i,j}$ is the j th decision variable in H_i , $i = 1, 2, \dots, N_{\text{pop}}$ and $j = 1, 2, \dots, n$, $L(n)$ is an n -dimensional vector which contains a Latin Hypercube Sampling of n values from the interval $[0, 1]$, and $L(n)|_j$ is the j th element in vector $L(n)$.

FIGURE 1: A schematic diagram of multitopology.

3.2. Multitopology Migration Operator. Migration operator is the main operator of BBO algorithm, because it determines how the new population is generated from the previous population. So, it influences BBO's search trajectory from the initial population. To enhance the performance of BBO, it is very necessary to analyze the characteristics of the migration operator.

During the migration process, we first use λ to decide H_i (immigrated habitat). The habitats with greater values of λ are more likely to be selected as H_i . Then use μ to decide H_e (emigrated habitat) from all other habitats except H_i . Also, the habitats with greater values of μ are more likely to be selected as H_e . As a result, it is from several habitats with great values of μ that H_i tends to get information and H_i directly copies information from them. Moreover, each habitat is possible to get information from any other habitats and information in each habitat is possible to be emigrated to any other habitats, which means the migration process is based on global topology, as shown in Figure 1(a). In global topology, habitats have strong connection with each other and information spreads very quickly, just like virus propagation, according to social network theory [28]. Due to global topology and direct-copying migration mechanism, as mentioned above, information in several good-quality habitats would be copied to the whole habitats soon. Although it can rapidly improve the quality of bad habitats, all candidate solutions (i.e., habitats) would be homogeneous soon, resulting in poor exploration ability. In allusion to the defect, multitopology migration operator is developed by modifying topology and copy mode.

Multitopology includes two kinds of topology: global topology and ring topology, as shown in Figure 1. Figure 1(a) represents global topology and Figure 1(b) represents ring topology where every habitat is only connected with two adjacent habitats. Divide the population into two groups, that is, Group A and Group B. Habitats in Group A are migrated based on global topology and habitats in Group B

are based on ring topology. To characterize the proportion of ring topology in multitopology, define *Mixed Degree of Ring Topology* as σ_{MDR} , which is expressed as

$$\sigma_{\text{MDR}} = \frac{N_{\text{pop.B}}}{N_{\text{pop}}}, \quad (3)$$

where $N_{\text{pop.B}}$ is the number of habitats in Group B.

The motivation of introducing the parameter σ_{MDR} is to balance both exploitation and exploration; that is, if $\sigma_{\text{MDR}} \rightarrow 0$, all the habitats are migrated based on global topology, as the ones migrated in basic BBO. It can speed up the convergence but easily make the algorithm get stuck in local optima; on the other hand, if $\sigma_{\text{MDR}} \rightarrow 1$, it can increase the population diversity but reduce the convergence speed. Therefore, we can tune the parameter to achieve a proper balance between the exploitation and the exploration so as to adapt the algorithm for optimization problems with different characteristics.

As for copy-mode modification, an indirect-copying migration operator is introduced. This operator is expressed as

$$H_i = H_e + \varphi \cdot (H_e - H_r), \quad (4)$$

where H_r is a habitat randomly selected from the habitats except H_i and H_e and φ is a uniformly distributed real random number within the range $[-1, 1]$.

The core idea of constructing this operator is based on two considerations. One is that, instead of directly copying information from H_e , this operator uses the relatively good habitat H_e as the base and uses $\varphi(H_e - H_r)$ as a perturbation on H_e , thus bringing more information from exploited feasible space. The other is that, by adding a perturbation in the operator, it can avoid the homogenization of habitats and thus helps to remedy the mentioned defect. The pseudocode of migration operator is as in Algorithm 1.

There are three points that should be noted. First, the two groups are required to have roughly the same average HSI level. The grouping method in this paper is as follows:

```

START
Divide the whole habitats into Group A and Group B
Implement migration operator in Group A
for  $i = 1$  to  $N_{\text{pop},A}$ 
  for  $d = 1$  to  $n$ 
    if  $\text{rand}(0, 1) < \lambda_i$  then
      Select emigrated habitat  $H_e$  with respect to rate  $\mu_e$  from all habitat except  $H_i$ 
      if  $\text{rand}(0, 1) < \mu_e$  then
        Randomly select a habitat from all habitats both in A and B group ( $r \neq i \neq e$ )
         $V_{i,d} = H_{e,d} + \varphi(H_{e,d} - H_{r,d})$ 
      else
         $V_i = H_i$ 
      end if
    else
       $V_i = H_i$ 
    end if
  end for
end for
Implement migration operator in Group B
Randomly arrange the habitats in Group B in a ring topology
for  $i = 1$  to  $N_{\text{pop},B}$ 
  for  $d = 1$  to  $n$ 
    if  $\text{rand}(0, 1) < \lambda_i$  then
      Select emigrated habitat  $H_e$  with respect to rate  $\mu_e$  from the two habitats adjacent to  $H_e$ 
      if  $\text{rand}(0, 1) < \mu_e$  then
        Randomly select a habitat from the whole habitat ( $r \neq i \neq e$ )
         $V_{i,d} = H_{e,d} + \varphi(H_{e,d} - H_{r,d})$ 
      else
         $V_i = H_i$ 
      end if
    else
       $V_i = H_i$ 
    end if
  end for
end for
Mix the whole habitats together
END

```

ALGORITHM 1: Migration operator.

- (1) Sort all habitats by the HSI from high to low and then number all of them from 1 to N_{pop} .
- (2) Select $N_{\text{pop},B}$ integers from interval $[1, N_{\text{pop}}]$ by using Uniform Sampling Technique.
- (3) Put habitats whose numbers are those selected integers into Group B and the rest of the habitats are put into Group A.

Secondly, habitat H_r , which is used for adding perturbation in operator (4), is selected from all habitats instead of only from the habitats in Group A or Group B. Thirdly, the two groups are mixed up at the end of each migration process and the population is regrouped at the beginning of each migration process. The second and third points show that although the habitats in the two groups are based on totally different topology to implement the migration process, the two groups can still adequately share their information and collaborate to move toward a more promising direction.

To verify the effectiveness and efficiency of the multitopology migration process, an enhanced BBO with the proposed migration operator (EBBO) is constructed to compare with basic BBO. Note that EBBO differs from BBO only in that it uses multitopology migration operator to replace the basic migration operator. Both EBBO and basic BBO are conducted on a 30-dimensional Alpine function f_{14} described in Table 1.

For a fair comparison, both BBO and EBBO start their iterative process from the same initial population. The values of x_1 and x_2 in each habitat observed at different iterative stages are plotted in Figure 2. As analyzed at the beginning of this subsection, BBO's habitats move much closer to each other before the 100th generation than the habitats of EBBO and BBO falls into local optima minima. Oppositely, EBBO finally pulls its population to swarm together toward the optimal region and converge to the optimal global solution. The comparative results show that the proposed migration

TABLE 1: Benchmark functions in experimental tests.

F	Function name	Formula	Dimension	Search domain
f_{01}	Elliptic	$f(X) = \sum_{i=1}^D x_i^2 \cdot (10^6)^{[(i-1)/(D-1)]}$	D	$-100 \leq x_i \leq 100$
f_{02}	Exponential	$f(x) = \exp\left(0.5 \cdot \sum_{i=1}^D x_i\right) - 1$	D	$-1.28 \leq x_i \leq 1.28$
f_{03}	Quartic	$f(X) = \sum_{i=1}^D i \cdot x_i^4 + \text{random}[0, 1]$	D	$-100 \leq x_i \leq 100$
f_{04}	Rosenbrock	$f(X) = \sum_{i=1}^D \left[100 (x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$	D	$-100 \leq x_i \leq 100$
f_{05}	Schwefel 2	$f(X) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j \right)^2$	D	$-100 \leq x_i \leq 100$
f_{06}	Schwefel 3	$f(X) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	D	$-100 \leq x_i \leq 100$
f_{07}	Schwefel 4	$f(X) = \max_{i=1}^D x_i $	D	$-100 \leq x_i \leq 100$
f_{08}	Sphere	$f(X) = \sum_{i=1}^D x_i^2$	D	$-100 \leq x_i \leq 100$
f_{09}	Step	$f(X) = \sum_{i=1}^D (0.5 + x_i)^2$	D	$-100 \leq x_i \leq 100$
f_{10}	SumPower	$f(X) = \sum_{i=1}^D x_i ^{2^{i+1}}$	D	$-1 \leq x_i \leq 1$
f_{11}	SumSquare	$f(X) = \sum_{i=1}^D i \cdot x_i^2$	D	$-10 \leq x_i \leq 10$
f_{12}	Zakharov	$f(X) = \sum_{i=1}^D x_i^2 + \left(\sum_{i=1}^D 0.5 i x_i \right)^2 + \left(\sum_{i=1}^D 0.5 i x_i \right)^4$	D	$-5 \leq x_i \leq 10$
f_{13}	Ackley	$f(x, y) = -20 \exp\left(-0.2 \sqrt{0.5 (x^2 + y^2)}\right) - \exp(0.5 (\cos(2\pi x) + \cos(2\pi y))) + e + 20$	D	$-100 \leq x_i \leq 100$
f_{14}	Alpine	$f(X) = \sum_{i=1}^D x_i \sin x_i + 0.1 x_i $	D	$-10 \leq x_i \leq 10$
f_{15}	Griewank	$f(X) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	D	$-100 \leq x_i \leq 100$
f_{16}	Levy	$f(X) = \sum_{i=1}^{D-1} (x_i - 1)^2 \cdot [1 + 10 \sin^2(3\pi x_{i+1})] + \sin^2(3\pi x_1) + x_D - 1 \cdot [1 + \sin^2(3\pi x_D)]$	D	$-100 \leq x_i \leq 100$

TABLE I: Continued.

F	Function name	Formula	Dimension	Search domain
f_{17}	Penalty 1	$f(X) = \pi D^{-1} \left\{ 10 \sin^2(\pi y_i) + \sum_{i=1}^{D-1} (y_i - 1)^2 \cdot [1 + 10 \sin^2(\pi y_{i+1})] \right\} + \pi D^{-1} (y_D - 1)^2 + \sum_{i=1}^n u(x_i, 10, 100, 4)$	D	$-50 \leq x_i \leq 50$
f_{18}	Penalty 2	$f(X) = \sin^2(3\pi x_1) + 0.1 \sum_{i=1}^{D-1} (x_i - 1)^2 \cdot [1 + 10 \sin^2(3\pi x_{i+1})] + (x_D - 1)^2 \cdot [1 + 10 \sin^2(3\pi x_D)] + \sum_{i=1}^D u(x_i, 10, 100, 4)$	D	$-50 \leq x_i \leq 50$
f_{19}	Rastrigin	$f(X) = 10D + \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i)]$	D	$-100 \leq x_i \leq 100$
f_{20}	Schwefel	$f(X) = 418.98288 \times D - \sum_{i=1}^D x_i \sin(\sqrt{ x_i })$	D	$-512.03 \leq x_i \leq 511.97$
f_{21}	Weierstrass	$f(X) = \sum_{i=1}^D \left(\sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k (x_i + 0.5))] \right) - D \sum_{k=0}^{k_{\max}} a^k \pi b^k$ $a = 0.5, b = 3, k_{\max} = 30$	D	$-0.5 \leq x_i \leq 0.5$
f_{22}	Bohachevsky 2	$f(X) = \sum_{i=1}^D [x_i^2 + 2x_{i+1}^2 - 0.3 \cos(3\pi x_i) \cos(3\pi x_{i+1}) + 0.3]$	D	$-100 \leq x_i \leq 100$
f_{23}	Expansion of F_{10}	$f(X) = F_{10}(x_1, x_2) + \dots + F_{10}(x_{i-1}, x_i) + \dots + F_{10}(x_{n_2}, x_1)$ $F_{10}(x, y) = (x^2 + y^2)^{0.25} \cdot \left[\sin^2(50(x^2 + y^2)^{0.1} + 1) \right]$	D	$-100 \leq x_i \leq 100$
f_{24}	NCRastrigin	$f(X) = \sum_{i=1}^D [y_i^2 - 10 \cos(2\pi y_i) + 10]$ $y_i = \begin{cases} x_i, & x_i < 0.5; \\ \text{round}(\frac{2x_i}{2}), & x_i > 0.5 \end{cases}$	D	$-5.12 \leq x_i \leq 5.12$

FIGURE 2: Variables x_1 and x_2 observed at different iterative stages of BBO and EBBO.

operator can indeed enhance the performance of basic BBO algorithm.

3.3. Quantitative Orthogonal Learning Operator. Assume solutions are three-dimensional, and denote H_i as (x_1, y_1, z_1) and H_e as (x_2, y_2, z_2) . Their possibly generated solutions via migration process are (x_1, y_2, z_1) , (x_2, y_1, z_1) , (x_1, y_1, z_2) , (x_2, y_2, z_1) , (x_2, y_1, z_2) , and (x_1, y_2, z_2) , respectively. As shown in Figure 3, all of them are located on the vertexes of the cuboid determined by H_i and H_e .

It can be seen from Figure 3 that the inner area of the cuboid cannot be investigated. However, more competitive solutions are very likely to locate in the inner area. Ignoring the inner area is bad for the solution quality of the algorithm. The purpose of introducing QOL process is to make the

algorithm able to search the solution space thoroughly so that more competitive solutions can be provided.

The process is described as follows. Suppose H_1 and H_2 are the two parental habitats of the QOL process, which are noted as follows:

$$\begin{aligned} H_1 &= [x_{1,1}, x_{1,2}, \dots, x_{1,n-1}, x_{1,n}] \\ H_2 &= [x_{2,1}, x_{2,2}, \dots, x_{2,n-1}, x_{2,n}]. \end{aligned} \quad (5)$$

Each of them has n decision variables. Denote L and U as the lower/upper bound of the space determined by H_1 and H_2 . The low/upper bound are expressed as follows:

$$\begin{aligned} L &= [\min(x_{1,1}, x_{2,1}), \min(x_{1,2}, x_{2,2}), \dots, \\ &\quad \min(x_{1,n}, x_{2,n})] \end{aligned}$$

FIGURE 3: Spatial location of H_i , H_e and their possibly generated solution in migration process.

$$U = [\max(x_{1,1}, x_{2,1}), \max(x_{1,2}, x_{2,2}), \dots, \max(x_{1,n}, x_{2,n})]. \quad (6)$$

The space determined by H_1 and H_2 is quantized into Q levels such that the difference between any two successive levels is the same. The quantized space is represented by matrix A , where the element in i th row and j th column is expressed as follows:

$$a_{ij} = \min(x_{1,j}, x_{2,j}) + \frac{i-1}{Q-1} \cdot (u_j - l_j), \quad (7)$$

where, for $1 \leq i \leq Q$, $1 \leq j \leq n$, u_j and l_j are the j th elements in vectors U and L , respectively. A has n columns and each column corresponds to a decision variable and Q rows correspond to Q quantized values of the corresponding decision variable of the column. As a result, the solution space is quantized into Q^n points, which is a large number. If we just identify these Q^n points one by one to determine the optimal solution, it will waste lots of computational time. To improve computational efficiency, firstly, merge n decision variables into M factors. It means level (i.e., value) combinations of variables in a factor are not considered. In this paper, the number of variables in each factor is roughly the same. By merging variables, the number of quantized points (i.e., combinations) is decreased to Q^M . Secondly, apply orthogonal design to select some representative samples of the quantized points and then use them to determine the optimal solutions (i.e., level combinations). As for how to use representative samples selected by orthogonal design to determine the optimal level combinations for a multifactor and multilevel optimization problem, please refer to literature [29]. Thirdly, instead of being applied to every habitat H_i ,

QOL process is implemented only once at each iteration. When optimal solutions are determined, use them to replace the two parental habitats.

3.4. Diversity Mechanism. To skip out on local optimal solutions, diversity mechanism is introduced. When the average HSI of the current population does not change over some certain generations in the evolutionary process, which is expressed as

$$|(f_{g-k} - f_g)| < \varepsilon, \quad (8)$$

where f_g and f_{g-k} are the best fitness values at the g th and $(g-k)$ th generations, respectively, and k, ε are tolerance factors, some habitats are replaced as follows:

$$x_{i,j} = \begin{cases} x_j^{\min} + (x_j^{\max} - x_j^{\min}) \cdot \text{rand}, & \text{rand} \leq \phi \\ x_{i,j}, & \text{rand} > \phi, \end{cases} \quad (9)$$

where $i = N_{\text{pop}}/2, N_{\text{pop}}/2 + 1, \dots, N_{\text{pop}}$, $j = 1, 2, \dots, n$, $x_{i,j}$ denotes the j th variable in the i th habitat, and rand is a randomly generated value from interval $[0, 1]$. ϕ is a user-defined parameter, which represents the randomization rate. Here, we set it to 0.2.

3.5. Algorithm Structure and Main Procedures. By incorporating the multitopology migration operator, QOL strategy, and diversity mechanism into BBO, the proposed MTQLBBO is developed and its main procedures are described as follows:

- (1) Set the parameters of MTQLBBO, mainly including I , E , Q , M , the max number of fitness evaluations (FES_Max), N_{pop} , and σ_{MDR} .
- (2) Initialize the population based on Latin Hypercube Sampling.
- (3) Sort the population from best to worst and obtain each habitat's species number according to its ranking in the population.
- (4) Calculate λ and μ by formula (1) and implement the migration and mutation process based on λ and μ .
- (5) Implement QOL operator.
- (6) Similar to BBO, perform elite reservation strategy.
- (7) If required, apply the diversity mechanism.
- (8) Check the stopping criteria. If met, stop and output the result; otherwise, go back to step (3) and continue.

4. Case Studies

4.1. Experimental Settings and Experiment Outline. To verify the performance of MTQLBBO, a test suit, consisting of 24 scalable benchmark functions [30, 31] with different characteristics, which are summarized in Table 1, is used, adopted in this section. $f_{01}-f_{12}$ are unimodal functions and $f_{13}-f_{24}$ are multimodal functions. The parameters are stated in Table 2. As for stopping criteria, the maximum number of fitness evaluations for each function is described in Table 3.

FIGURE 4: The experiment structure diagram.

TABLE 2: Parameter settings of the MTQLBBO algorithm used in this paper.

Parameters	Notation	Value
Maximum immigration rate	I	1
Maximum emigration rate	E	1
Population size	N_{pop}	100
Orthogonal array	$L_C(Q^M)$	$L_{16}(3^4)$
Mixed degree of ring topology	σ_{MDR}	0.3

TABLE 3: Maximum evaluations for each function.

FES_Max/	Functions
150,000	$f_{03}, f_{04}, f_{08}, f_{12}, f_{14}, f_{15}, f_{17}, f_{19}, f_{22}$
200,000	$f_{06}, f_{11}, f_{13}, f_{16}, f_{18}$
250,000	f_{01}, f_{07}, f_{10}
300,000	$f_{14}, f_{20}, f_{21}, f_{23}, f_{24}$
400,000	f_{02}, f_{05}, f_{09}

For all experiments, these parameter settings are employed unless a change is mentioned.

For fair comparison between different algorithms, to eliminate the contingency, all experiment results are obtained based on 40 independent runs. The mean and the standard deviation of the best-of-run errors over 40 independent runs are provided to measure the performance. In addition, to compare the statistically significant differences between the

computational results of two algorithms, Mann-Whitney U test at a 0.05 level is adopted between all the compared algorithms throughout the paper. For clarity, the results of the best algorithm are marked in boldface.

To fully evaluate the performance of the proposed algorithm, several group experiments are carried out. The experiment structure is described as Figure 4.

The experiment can be divided into two parts: (1) to evaluate the performance of MTQLBBO; (2) to investigate the benefit of each modified component.

4.1.1. Evaluate the Performance of MTQLBBO

(a) *Results Comparison between BBO and MTQLBBO.* It aims to investigate whether the proposed improvement strategies can effectively enhance the performance of basic BBO algorithm. BBO and MTQLBBO are conducted on $f_{01}-f_{24}$ on $D = 30$.

(b) *Scalability for MTQLBBO.* A new proposed algorithm is expected to have excellent performance when solving optimization problems with different dimensions. So it is necessary to apply MTQLBBO on benchmark functions with different dimensions and investigate whether the algorithm can maintain good performance on different number of dimensions. MTQLBBO is conducted on $f_{01}-f_{24}$ on $D = 10, 40, 80, \text{ and } 160$ and $\text{FE_Max} = D \times 10,000$.

TABLE 4: Comparison of BBO and MTQLBBO for all benchmark functions ($N_{\text{pop}} = 100$, $D = 30$).

F	BBO		MTQLBBO	
	Mean	Standard deviation	Mean	Standard deviation
f_1	$1.98E + 04$	$1.43E + 04 +$	$3.45E - 70$	$6.24E - 72$
f_2	$5.12E - 05$	$2.21E - 05 +$	$0.00E + 00$	$0.00E + 00$
f_3	$2.95E - 02$	$1.12E - 03 +$	$0.00E + 00$	$0.00E + 00$
f_4	$1.01E + 00$	$7.89E + 01 -$	$1.43E + 00$	$6.87E - 01$
f_5	$6.50E - 01$	$8.52E + 01 -$	$6.87E - 01$	$4.52E - 01$
f_6	$8.40E - 02$	$6.72E - 02 +$	$0.00E + 00$	$0.00E + 00$
f_7	$3.11E + 00$	$5.23E - 02 +$	$4.23E - 03$	$7.23E - 06$
f_8	$1.34E - 01$	$8.32E - 02 +$	$0.00E + 00$	$0.00E + 00$
f_9	$1.23E - 01$	$2.21E - 02 +$	$0.00E + 00$	$0.00E + 00$
f_{10}	$7.42E - 02$	$2.64E - 02 +$	$6.78E - 72$	$8.23E - 75$
f_{11}	$8.98E - 07$	$1.24E - 06 +$	$8.23E - 85$	$1.43E - 83$
f_{12}	$4.14E - 03$	$1.03E - 03 +$	$2.18E - 71$	$5.36E - 70$
f_{13}	$8.82E - 02$	$2.98E - 01 +$	$7.29E - 16$	$5.05E - 15$
f_{14}	$1.01E - 02$	$2.76E - 03 +$	$5.04E - 14$	$3.93E - 15$
f_{15}	$3.68E - 02$	$1.98E - 02 +$	$2.36E - 07$	$5.52E - 08$
f_{16}	$2.89E - 02$	$2.36E - 02 +$	$1.12E - 42$	$2.51E - 43$
f_{17}	$2.52E - 03$	$3.70E - 03 +$	$2.67E - 34$	$8.25E - 38$
f_{18}	$2.57E - 02$	$1.21E - 02 +$	$1.35E - 34$	$2.47E - 35$
f_{19}	$1.21E + 01$	$2.58E + 00 +$	$0.00E + 00$	$0.00E + 00$
f_{20}	$6.82E - 01$	$5.23E - 02 +$	$8.94E - 22$	$1.12E - 13$
f_{21}	$9.51E - 01$	$1.23E - 01 +$	$0.00E + 00$	$0.00E + 00$
f_{22}	$8.63E + 00$	$1.02E + 00 +$	$0.00E + 00$	$0.00E + 00$
f_{23}	$1.36E + 01$	$1.89E + 00 +$	$1.89E - 40$	$5.69E - 39$
f_{24}	$5.74E - 02$	$4.23E - 02 +$	$5.98E - 09$	$9.36E - 10$
$w/t/l$	22/0/2			

(c) *Results Comparison between MTQLBBO and Other BBO Variant Algorithms.* Since basic BBO algorithm was proposed, some BBO variants have been proposed and they have been proved to have better excellence performance than basic BBO. So it is necessary to compare the computational results of MTQLBBO and other BBO variants and verify whether MTQLBBO has better performance than other BBO variants. The experiment is conducted on $D = 30$.

(d) *Results Comparison between MTQLBBO and Other Orthogonal Learning (OL) Based Algorithms.* Due to the advantage of OL strategy, it has been employed to many EAs. To prove superiority of the OL-blended strategy in our work, the results of MTQLBBO are compared with those of other OL-based algorithms. The experiment is conducted on $D = 30$.

(e) *Results Comparison between MTQLBBO and Other Non-OL-Based EAs.* Besides OL-based EAs, there are other types of EAs that have shown excellent performance. To validate the deeper significance of our work, MTQLBBO is compared with other non-OL-based EAs. The experiment is conducted on $D = 30$.

4.1.2. *Investigating the Benefit of Improvement Components.* It is known from Section 3 that there are mainly two improved components in MTQLBBO. An experiment is conducted to identify the benefit of each component. In addition, the effects of three additional parameters brought by these two improvement components are also analyzed.

4.2. *Comparison with Basic BBO.* To validate the effectiveness of MTQLBBO, the optimization problems are solved 40 times independently for each benchmark function using BBO and MTQLBBO. The number of dimensions D equals 30 in this subsection. The experimental statistical results are shown in Table 4. The comparison results according to Wilcoxon's Test are marked as "+", "-", and "=", which means that the performance of MTQLBBO is better than, equal to, and worse than that of its competitors, respectively. The statistical results denoted as " $w/t/l$ " are shown in the last row of the table. " $w/t/l$ " denotes that MTQLBBO wins on w , ties on t , and loses on l benchmark functions, respectively, compared with its competitors. It can be seen from Table 2 that MTQLBBO significantly performs better than basic BBO on 22 out of 24 functions except for functions f_4 and f_5 . For either unimodal functions or multimodal functions, MTQLBBO outperforms

FIGURE 5: Convergence graphs of MTQLBBO and other five BBO variants for six representative test functions.

BBO. It indicates that the modified components not only enhance the local search ability and search accuracy, but also enable the algorithm to have a better ability to escape from local minima and jump to a better near-global optimum. Namely, both excellent exploitation and exploration ability are achieved.

Besides solution quality, convergence property is also an important evaluation indicator for judging the performance of an algorithm. To evaluate the convergence performance, convergence curves of some representative benchmark functions are plotted and shown in Figure 5. It is evident that MTQLBBO converges faster than BBO.

4.3. Scalability Study. To explore the effect of the problem dimension on the performance of MTQLBBO, a scalability study based on $D = 10, 40, 80$, and 160 is carried out, and the experimental results after $D \times 10,000$ FEs are summarized in Table 5. Note that, in this subsection, all the parameters remain unchanged except D .

From the results, it can be seen that MTQLBBO performs better than BBO in all different dimensions. For $D = 10$, MTQLBBO obtains better results on 21 functions. For $D = 40$, MTQLBBO achieves better results than BBO on 22 functions while BBO gives better results only on f_7 and f_9 . In the case of $D = 80$, MTQLBBO performs better than BBO

TABLE 5: Mean and standard deviation of optimal results of BBO and MTQLBBO on different dimensions ($N_{\text{pop}} = 100$).

F	BBO			MTQLBBO
	Mean (std. dev.)			Mean (std. dev.)
D = 10				
f ₁	4.81E + 03 (5.02E + 03)	+		0.00E + 00 (0.00E + 00)
f ₂	3.87E − 06 (2.47E − 06)	+		0.00E + 00 (0.00E + 00)
f ₃	1.70E − 05 (1.10E − 05)	−		4.97E − 05 (4.94E − 06)
f ₄	3.92E + 01 (3.71E + 01)	+		0.00E + 00 (0.00E + 00)
f ₅	6.70E + 00 (7.54E − 01)	+		3.97E − 13 (3.23E − 12)
f ₆	9.11E − 04 (2.04E − 04)	+		7.59E − 77 (4.40E − 76)
f ₇	6.49E − 02 (1.34E − 02)	+		4.98E − 45 (2.22E − 46)
f ₈	0.00E + 00 (0.00E + 00)	=		0.00E + 00 (0.00E + 00)
f ₉	7.17E − 04 (1.23E − 04)	+		0.00E + 00 (0.00E + 00)
f ₁₀	5.48E − 03 (8.15E − 04)	+		1.45E − 102 (3.23E − 104)
f ₁₁	1.91E − 09 (4.39E − 10)	+		0.00E + 00 (0.00E + 00)
f ₁₂	2.42E − 04 (1.89E − 04)	+		8.70E − 75 (8.47E − 74)
f ₁₃	1.15E − 03 (5.73E − 04)	+		4.58E − 18 (3.24E − 19)
f ₁₄	2.02E − 03 (5.91E − 04)	+		3.87E − 75 (6.46E − 77)
f ₁₅	1.18E − 02 (5.54E − 03)	+		0.00E + 00 (0.00E + 00)
f ₁₆	8.27E − 03 (6.07E − 03)	+		0.00E + 00 (0.00E + 00)
f ₁₇	7.05E − 06 (3.42E − 05)	+		0.00E + 00 (0.00E + 00)
f ₁₈	4.21E − 03 (2.03E − 02)	+		8.42E − 55 (5.32E − 58)
f ₁₉	0.00E + 00 (0.00E + 00)	=		0.00E + 00 (0.00E + 00)
f ₂₀	8.87E − 02 (2.46E − 02)	+		1.45E − 10 (4.14E − 09)
f ₂₁	3.58E − 03 (9.12E − 04)	+		0.00E + 00 (0.00E + 00)
f ₂₂	1.14E + 00 (5.14E − 01)	+		0.00E + 00 (0.00E + 00)
f ₂₃	6.10E + 00 (5.18E − 01)	+		5.07E − 26 (8.59E − 27)
f ₂₄	2.29E − 02 (1.59E − 02)	+		7.36E − 11 (3.49E − 12)
w/t/l			21/2/1	
D = 40				
f ₁	4.04E + 04 (7.42E + 02)	+		4.32E − 77 (3.42E − 81)
f ₂	8.32E − 05 (3.32E − 05)	+		0.00E + 00 (0.00E + 00)
f ₃	5.15E − 01 (2.81E − 02)	+		4.01E − 04 (4.45E − 05)
f ₄	2.18E + 02 (8.53E + 01)	+		3.47E + 00 (5.43E − 01)
f ₅	2.09E + 01 (4.21E − 01)	+		3.37E − 01 (6.01E − 02)
f ₆	6.15E − 02 (3.10E − 02)	+		8.15E − 40 (4.09E − 42)
f ₇	2.02E − 01 (4.57E − 03)	−		8.34E − 01 (1.65E − 02)
f ₈	3.24E − 01 (6.73E − 02)	+		0.00E + 00 (0.00E + 00)
f ₉	1.47E − 01 (7.31E − 02)	−		2.86E − 01 (9.93E − 02)
f ₁₀	1.21E − 01 (9.34E − 03)	+		3.58E − 40 (5.43E − 41)
f ₁₁	1.23E − 06 (3.95E − 07)	+		2.53E − 65 (9.12E − 67)
f ₁₂	1.84E − 03 (8.06E − 04)	+		4.59E − 75 (1.23E − 73)
f ₁₃	9.99E − 02 (2.31E − 03)	+		1.39E − 15 (3.43E − 16)
f ₁₄	3.21E − 02 (1.04E − 02)	+		9.67E − 64 (3.61E − 66)
f ₁₅	6.12E − 02 (9.43E − 03)	+		0.00E + 00 (0.00E + 00)
f ₁₆	5.32E − 02 (4.01E − 02)	+		5.73E − 29 (6.92E − 31)
f ₁₇	2.83E − 03 (1.81E − 03)	+		6.16E − 27 (2.42E − 29)
f ₁₈	3.54E − 02 (9.64E − 03)	+		4.23E − 32 (8.89E − 33)
f ₁₉	1.43E − 14 (2.43E − 16)	+		0.00E + 00 (0.00E + 00)
f ₂₀	8.12E − 01 (4.91E − 01)	+		1.62E − 03 (4.93E − 04)
f ₂₁	1.53E + 00 (6.93E − 01)	+		9.41E − 06 (7.15E − 06)
f ₂₂	1.23E + 01 (1.12E + 00)	+		0.00E + 00 (0.00E + 00)
f ₂₃	2.69E + 01 (5.58E + 00)	+		1.21E − 10 (4.25E − 12)
f ₂₄	2.50E − 01 (5.95E − 02)	+		2.14E − 10 (6.54E − 11)
w/t/l			22/0/2	

TABLE 5: Continued.

F	BBO			MTQLBBO
	Mean (std. dev.)			Mean (std. dev.)
D = 80				
f ₁	6.12E + 04 (5.95E + 03)	+		1.48E − 82 (5.34E − 87)
f ₂	2.97E − 04 (1.37E − 05)	+		8.13E − 06 (9.12E − 08)
f ₃	7.41E − 01 (2.87E − 02)	−		8.81E − 01 (2.49E − 01)
f ₄	3.17E + 02 (5.91E + 01)	+		0.00E + 00 (0.00E + 00)
f ₅	5.49E + 01 (7.01E + 00)	+		8.17E − 01 (6.52E − 02)
f ₆	9.91E − 02 (5.37E − 03)	+		6.21E − 34 (5.04E − 35)
f ₇	2.96E + 00 (9.69E − 01)	+		9.14E − 01 (4.91E − 02)
f ₈	8.71E − 01 (1.73E − 01)	+		0.00E + 00 (0.00E + 00)
f ₉	9.97E − 01 (9.77E − 02)	+		4.86E − 01 (5.40E − 02)
f ₁₀	7.79E − 01 (6.71E − 02)	+		4.52E − 29 (7.61E − 31)
f ₁₁	5.19E − 06 (8.75E − 08)	+		7.31E − 41 (2.01E − 43)
f ₁₂	2.56E − 01 (5.97E − 02)	+		1.23E − 69 (3.58E − 70)
f ₁₃	5.43E − 01 (6.31E − 03)	+		4.91E − 14 (7.75E − 16)
f ₁₄	8.92E − 02 (7.74E − 02)	+		9.67E − 34 (4.90E − 36)
f ₁₅	8.32E − 02 (1.21E − 02)	+		0.00E + 00 (0.00E + 00)
f ₁₆	9.14E − 02 (3.08E − 02)	+		0.00E + 00 (0.00E + 00)
f ₁₇	7.02E − 03 (3.43E − 03)	+		3.61E − 31 (3.21E − 32)
f ₁₈	6.98E − 02 (1.29E − 02)	+		5.21E − 18 (1.20E − 20)
f ₁₉	7.51E − 08 (6.19E − 09)	+		0.00E + 00 (0.00E + 00)
f ₂₀	3.31E + 00 (6.29E − 01)	+		7.37E − 03 (9.13E − 05)
f ₂₁	3.19E + 00 (6.09E − 01)	+		9.41E − 06 (7.15E − 06)
f ₂₂	2.87E + 01 (4.56E + 00)	+		0.00E + 00 (0.00E + 00)
f ₂₃	4.87E + 01 (1.23E + 00)	+		6.57E − 08 (3.56E − 09)
f ₂₄	1.23E − 01 (5.47E − 01)	+		9.14E − 07 (3.58E − 08)
w/t/l			23/0/1	
D = 160				
f ₁	9.25E + 04 (9.95E + 03)	+		0.00E + 00 (0.00E + 00)
f ₂	6.95E − 03 (7.79 − 04)	+		4.29E − 01 (5.78E − 02)
f ₃	2.10E + 00 (9.13E − 01)	+		8.01E − 01 (5.30E − 02)
f ₄	4.91E + 02 (6.35E + 01)	+		4.04E − 15 (1.27E − 16)
f ₅	7.09E + 01 (4.01E + 00)	+		1.43E + 00 (8.42E − 01)
f ₆	4.24E − 02 (1.45E − 03)	+		7.51E − 38 (3.96E − 40)
f ₇	1.49E + 00 (4.92E − 01)	+		8.57E − 01 (5.73E − 02)
f ₈	1.32E + 00 (7.53E − 01)	+		0.00E + 00 (0.00E + 00)
f ₉	4.40E − 01 (4.97E − 02)	−		7.86E − 01 (4.09E − 02)
f ₁₀	5.64E − 01 (3.23E − 02)	+		6.31E − 30 (9.09E − 31)
f ₁₁	8.13E − 04 (8.75E − 08)	+		5.14E − 41 (4.91E − 43)
f ₁₂	2.14E + 02 (5.21E + 01)	+		2.36E − 65 (2.97E − 70)
f ₁₃	1.73E + 00 (5.91E − 01)	+		6.23E − 15 (7.75E − 16)
f ₁₄	2.21E − 01 (9.14E − 02)	+		3.31E − 23 (5.34E − 26)
f ₁₅	1.21E − 01 (5.39E − 01)	+		0.00E + 00 (0.00E + 00)
f ₁₆	4.30E − 01 (3.08E − 02)	+		2.49E − 20 (6.92E − 23)
f ₁₇	4.42E − 02 (5.31E − 03)	+		4.31E − 26 (7.79 − 28)
f ₁₈	3.45E − 01 (4.57E − 02)	+		6.10E − 15 (9.23E − 16)
f ₁₉	4.93E − 10 (6.19E − 12)	+		0.00E + 00 (0.00E + 00)
f ₂₀	5.63E + 00 (8.39E − 01)	+		3.29E − 02 (7.13E − 04)
f ₂₁	3.57E + 00 (4.37E − 01)	+		1.43E − 05 (2.30E − 07)
f ₂₂	2.45E + 02 (2.12E + 01)	+		0.00E + 00 (0.00E + 00)
f ₂₃	5.47E + 01 (3.78E + 00)	+		1.45E − 02 (5.69E − 03)
f ₂₄	3.21E + 00 (2.25E − 01)	+		5.45E − 07 (2.36E − 08)
w/t/l			23/0/1

FIGURE 6: Convergence graphs of MTQLBBO and other five BBO variants for six representative test functions.

on 23 functions while BBO achieves better results only on f_3 . In the case of $D = 160$, BBO gets better results only on f_9 .

From the above comparison, we can conclude that MTQLBBO has better performance than BBO for solving both low dimensional and high dimensional optimization problems on the selected instances. In addition, it is noticed that, for MTQLBBO, although it is able to provide better results on some functions when the dimension increases from 10 to 160, its performance deteriorates on more functions with the growth of problem dimension. The main reason is that the search space increases dramatically, thus raising the difficulty of solving the problem.

4.4. Comparison with Other BBO Variant Algorithms. The performance of MTQLBBO is compared with other five BBO variant algorithms at $D = 30$; $N_{\text{pop}} = 100$ in this subsection. These BBO variant algorithms are MOBBO [24], PBBO-G [22], RCBBO-G [32], DBBO [26], and POLBBO [23]. All the other parameters of these variants algorithms are set the same as those in their original papers. Each benchmark function is optimized 40 times independently, and the optimal results are summarized in Table 6.

It can be seen from Table 6 that MTQLBBO surpasses all BBO variant algorithms on most benchmark functions. MTQLBBO obtains better results than MOBBO, PBBO-G, DBBO, RCBBO-G, and POLBBO on 18, 21, 20, 23, and 13 functions, respectively. MTQLBBO is surpassed by MOBBO, PBBO-G, DBBO, RCBBO-G, and POLBBO only on 4, 2, 2, 0, and 6 functions, respectively. As for convergence property, convergence curves from some benchmark functions are plotted in Figure 6. It shows that MTQLBBO converges faster than its competitors.

4.5. Comparison with Other OL-Based Algorithms. Due to the advantages of orthogonal learning (OL) strategy, it has been employed in many EAs to improve algorithms' performance. In this subsection, the results achieved by MTQLBBO are compared with those achieved by other OL-based evolutionary algorithms, which include OXDE [33], ODE/2 [34], OPSP [35], OLPSO-G [36], OLPSO-L [36], OXBBO [37], and POLBBO [23]. OXDE and ODE/2 are proposed by embedding OL strategy into DE algorithm while OPSP, OLPSO-G, and OLPSO-L are developed by embedding OL strategy into PSO algorithm. Similar to MTQLBBO, OXBBO

TABLE 6: Comparison with other BBO variant algorithms ($N_{\text{pop}} = 100$, $D = 30$).

F	MOBBO Mean (std. dev.)	PBBO-G Mean (std. dev.)	DBBO Mean (std. dev.)	RCBBO-G Mean (std. dev.)	POLBBO Mean (std. dev.)	MTQLBBO Mean (std. dev.)
f_1	$2.30E-34$ (2.56E-35)	$6.99E-06$ (5.42E-5)	$7.37E-03$ (5.95E-04)	$3.72E+01$ (3.47E+00)	$8.69E-68$ (7.40E-70)	$3.45E-70$ (6.24E-72)
f_2	$5.35E-16$ (5.33E-15)	$4.18E-11$ (6.64E-12)	$4.31E-10$ (2.17E-12)	$7.95E-05$ (4.52E-04)	$0.00E+00$ (0.00E+00)	$0.00E+00$ (0.00E+00)
f_3	$7.83E-02$ (3.95E-2)	$5.53E-03$ (4.83E-04)	$4.02E-02$ (5.49E-02)	$5.62E+00$ (1.63E+00)	$8.74E-04$ (6.84E-04)	$0.00E+00$ (0.00E+00)
f_4	$5.57E+01$ (8.31E+00)	$6.99E+01$ (8.97E+00)	$4.90E+01$ (8.01E+00)	$2.64E+01$ (8.37E+00)	$1.46E+00$ (7.38E-01)	$1.43E+00$ (6.87E-01)
f_5	$1.45E-03$ (4.81E-3)	$6.16E-02$ (2.06E-02)	$6.22E-07$ (2.53E-07)	$6.82E+00$ (4.88E-1)	$8.80E-05$ (9.46E-06)	$6.87E-01$ (4.52E-01)
f_6	$5.80E-32$ (4.73E-33)	$5.95E-07$ (6.63E-08)	$5.23E-03$ (9.04E-03)	$2.03E-03$ (4.33E-4)	$6.13E-50$ (5.86E-61)	$0.00E+00$ (0.00E+00)
f_7	$1.39E-03$ (6.01E-04)	$5.19E-02$ (2.49E-02)	$9.38E-02$ (4.60E-03)	$1.54E-02$ (1.51E-02)	$6.19E-18$ (5.00E-20)	$4.23E-03$ (7.23E-06)
f_8	$7.44E-37$ (2.98E-36)	$4.78E-13$ (7.96E-15)	$4.61E-35$ (7.60E-39)	$1.12E-04$ (4.96E-05)	$5.59E-71$ (7.65E-79)	$0.00E+00$ (0.00E+00)
f_9	$0.00E+00$ (0.00E+00)	$0.00E+00$ (0.00E+00)	$0.00E+00$ (0.00E+00)	$0.00E+00$ (0.00E+00)	$0.00E+00$ (0.00E+00)	$0.00E+00$ (0.00E+00)
f_{10}	$8.30E-40$ (2.98E-43)	$3.93E-10$ (8.49E-11)	$4.51E-12$ (4.94E-14)	$1.55E-05$ (6.10E-06)	$5.95E-186$ (1.83E-185)	$6.78E-72$ (8.23E-75)
f_{11}	$1.60E-38$ (4.85E-39)	$2.32E-10$ (2.68E-11)	$4.20E-07$ (2.87E-08)	$2.52E+00$ (3.79E-01)	$5.24E-70$ (6.89E-72)	$8.23E-85$ (1.43E-83)
f_{12}	$5.14E-40$ (5.10E-41)	$2.45E-24$ (4.25E-25)	$2.98E-09$ (4.21E-09)	$6.54E-04$ (5.52E-04)	$9.25E-71$ (2.69E-70)	$2.18E-71$ (5.36E-70)
f_{13}	$3.33E-15$ (4.69E-15)	$3.95E-08$ (5.51E-07)	$6.64E-02$ (8.17E-03)	$2.12E-04$ (7.32E-04)	$4.00E-15$ (5.10E-14)	$7.29E-16$ (5.05E-15)
f_{14}	$8.23E-12$ (9.24E-13)	$8.32E-06$ (6.51E-06)	$9.08E-03$ (4.89E-04)	$1.83E-03$ (2.31E-03)	$2.87E-42$ (8.39E-40)	$5.04E-14$ (3.93E-15)
f_{15}	$4.05E-02$ (3.49E-04)	$5.62E-03$ (7.98E-02)	$4.73E-17$ (6.36E-19)	$4.22E-02$ (5.81E-01)	$0.00E+00$ (0.00E+00)	$2.36E-07$ (5.52E-08)
f_{16}	$1.43E-31$ (3.70E-33)	$4.30E-18$ (3.72E-19)	$3.10E-07$ (7.30E-08)	$3.01E-03$ (6.85E-04)	$7.43E-31$ (7.17E-30)	$1.12E-42$ (2.51E-43)
f_{17}	$5.96E-08$ (2.91E-10)	$1.92E-32$ (5.16E-34)	$2.25E-05$ (6.58E-06)	$5.36E-07$ (8.81E-06)	$1.57E-32$ (8.92E-30)	$2.67E-34$ (8.25E-38)
f_{18}	$5.96E-08$ (3.98E-07)	$2.29E-09$ (7.19E-10)	$8.26E-07$ (8.69E-06)	$3.31E-05$ (3.05E-04)	$2.46E-32$ (8.56E-31)	$1.35E-34$ (2.47E-35)
f_{19}	$0.00E+00$ (0.00E+00)	$5.29E-10$ (2.09E-08)	$0.00E+00$ (0.00E+00)	$3.61E-02$ (5.07E-03)	$0.00E+00$ (0.00E+00)	$0.00E+00$ (0.00E+00)
f_{20}	$8.58E-01$ (2.14E-02)	$7.75E-03$ (2.05E-04)	$6.18E-03$ (4.44E-03)	$6.12E-01$ (7.54E-01)	$7.79E-07$ (5.73E-08)	$8.94E-22$ (1.12E-13)
f_{21}	$1.38E-06$ (5.45E-07)	$2.46E-03$ (8.35E-04)	$2.76E-02$ (5.64E-03)	$1.86E+00$ (5.16E-01)	$0.00E+00$ (0.00E+00)	$0.00E+00$ (0.00E+00)
f_{22}	$5.86E-06$ (6.36E-05)	$8.12E-08$ (6.45E-07)	$2.45E-03$ (7.56E-03)	$1.45E-01$ (8.89E-01)	$0.00E+00$ (0.00E+00)	$0.00E+00$ (0.00E+00)
f_{23}	$0.00E+00$ (0.00E+00)	$4.98E-02$ (5.69E-02)	$1.56E+01$ (5.16E+00)	$2.41E+00$ (2.12E-01)	$2.58E-38$ (5.78E-39)	$1.89E-40$ (5.69E-39)
f_{24}	$2.13E-16$ (5.14E-16)	$4.12E-09$ (1.45E-08)	$2.45E-02$ (3.25E-03)	$3.14E-02$ (1.32E-02)	$0.00E+00$ (0.00E+00)	$5.98E-09$ (9.36E-10)
$w/t/l$	18/2/4	21/1/2	20/2/2	23/1/0	13/5/6	

and POLBBO are developed by embedding OL strategy into BBO algorithm. The comparative results are summarized in Table 7. The results of these seven compared algorithms are all taken from their corresponding references except that the results of OPSO are obtained from [36] and the results of POLBBO are from the conducted experiment in the previous subsection.

MTQLBBO outperforms OXDE, ODE/2, OPSO, OLPSO-G, OLPSO-L, OXBBO, and POLBBO on 6, 7, 10, 10, 7, 7, and 8 functions, respectively. The algorithm is surpassed by OXDE, ODE/2, OLPSO-L, OXBBO, and POLBBO on 2, 4, 2, 3, and 3 functions, respectively. For OPSO and OLPSO-G, the algorithm wins both of them on all functions. Overall, MTQLBBO provides relatively competitive solutions compared to other OL-based algorithms.

4.6. Comparison with Other Non-OL-Based EAs. The proposed algorithm MTQLBBO is further compared with some other non-OL-based EAs, which include CLPSO [38], CMA-ES [39], GL-25 [40], SLPSO [41], MGBGE [42], iMEABC [43], MSODPSO-G [44], and MSODPSO-L [44]. The comparative results are summarized in Table 8. CLPSO seems not to be better than MTQLBBO on any function. Both GL-25 and SLPSO only perform better than MTQLBBO on two functions. CMA-ES, MGBGE, iMEABC, MSODPSO-G, and MSODPSO-L each only perform better than MTQLBBO on three functions. In addition, compared with all listed non-OL-based EAs, MTQLBBO performs the best on 9 out of 13 functions, showing that MTQLBBO has superior performance in terms of solution quality.

5. Analysis of Modified Components

In this section, we have analyzed effects of modified components, that is, multitopology based migration strategy and QOL strategy, on the performance of the algorithm. The influences of three additional parameters, brought by the modified components, are also investigated in this section.

5.1. Effect on Convergence Property. Convergence property is one of the most important characteristics for an optimization algorithm. To analyze the effects different components have on convergence property, two variants of MTQLBBO, that is, MTBBO and QOLBBO, are developed. They differ from MTQLBBO only in that MTBBO does not contain the QOL process, while QLBBO implements BBO's original migration operator instead of the multitopology migration operator. Both of them are used to solve all benchmark functions, and their performance is compared with that of BBO and MTQLBBO. A representative convergence graph is plotted in Figure 7.

Comparing the convergence curves of BBO and MTBBO, we can see that MTBBO converges to a better solution than BBO, but its convergence speed in early stages is slower than BBO. It shows that multitopology migration strategy exactly improves exploration ability and can help converge

FIGURE 7: Convergence graphs of BBO, MBBO, QOLBBO, and MTQLBBO on function f_{13} .

to a better solution. However, the paid expense is reducing the convergence speed in the early iterative stages. QOLBBO converges faster and achieves a better solution than BBO by investigating the solution space more systematically. The paid expense is extra computational overhead brought by the QOL process.

Moreover, it is worth noting that, different from slower convergence speed in the early stages resulting from adding multitopology migration process to BBO, adding BBO to QOLBBO accelerates the convergence speed instead. The reason might be as follows. Good habitats guide the searching directions of the population. Information in good habitats spreads to the whole habitats very rapidly in BBO, and the whole habitats can move toward a more promising direction quickly. Multitopology reduces the speed of information spread and thus decelerates the convergence speed. However, MTBBO has good population diversity. When QOL process is implemented among habitats with better population diversity, the space that can be potentially investigated becomes much wider. Hence, the solution space can be investigated more thoroughly and a better-quality solution is more likely to be obtained at each iteration, which accelerates the convergence speed. It also proves that the mutually beneficial cooperation between the multitopology migration operator and QOL process can significantly help MTQLBBO improve the searching process toward the promising directions.

5.2. Effects on Solution Quality. To analyze the effects the modified components have on solution quality, both MTBBO and QOLBBO are used to optimize all test functions and the results are summarized in Table 9. It can be seen from Table 9 that both MTBBO and QOLBBO significantly perform better than BBO. Specifically, MTBBO obtains better solutions than BBO on 21 functions except f_{03} , f_{04} , and f_{12} and QOLBBO obtains better solutions on 21 functions except f_4 , f_5 , and f_{20} . Such results show that both multitopology migration

TABLE 7: Comparison with other OL-based algorithms.

F	OXDE ^a Mean (std. dev.)	ODE/2 Mean (std. dev.)	OPSO ^b Mean (std. dev.)	OLPSO-G ^b Mean (std. dev.)	OLPSO-L ^b Mean (std. dev.)	OXBBO Mean (std. dev.)	POLBBO ^c Mean (std. dev.)	MTQLBBO Mean (std. dev.)
f_3	NA ^d	1.45E-03 (4.20E-04)	5.50E-02 (1.70E-03)	1.16E-02 (4.10E-03)	1.64E-02 (3.25E-03)	2.01E-03 (6.45E-04)	8.74E-04 (2.10E-04)	0.00E+00 (0.00E+00)
f_4	4.78E-01 (1.30E+00)	0.00E+00 (0.00E+00)	4.96E+01 (3.65E+01)	2.15E+00 (2.99E+00)	1.26E+00 (1.40E+00)	1.89E+01 (1.84E-01)	4.40E+00 (3.17E-01)	1.43E+00 (6.87E-01)
f_5	NA	5.27E-27 (9.66E-27)	NA	NA	NA	3.21E-03 (1.96E-03)	8.80E-05 (8.32E-07)	6.87E-01 (4.52E-01)
f_6	NA	1.43E-18 (8.11E-19)	1.26E-10 (5.58E-11)	9.85E-30 (1.01E-29)	7.67E-22 (5.63E-22)	1.12E-19 (4.01E-20)	6.13E-50 (4.19E-55)	0.00E+00 (0.00E+00)
f_7	NA	2.72E-15 (9.30E-15)	NA	NA	NA	2.61E-15 (8.83E-15)	6.19E-18 (6.05E-17)	4.23E-03 (7.23E-06)
f_8	5.21E-59 (1.82E-58)	2.06E-23 (1.83E-23)	6.45E-18 (4.46E-18)	4.12E-54 (6.34E-54)	1.11E-38 (1.28E-38)	3.96E-28 (2.47E-28)	5.59E-71 (3.72E-72)	0.00E+00 (0.00E+00)
f_9	NA	0.00E+00 (0.00E+00)	NA	NA	NA	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)
f_{13}	2.66E-15 (3.97E-14)	4.67E-13 (1.86E-13)	6.23E-09 (1.87E-09)	7.98E-15 (2.03E-15)	4.14E-15 (0.00E+00)	6.79E-15 (1.50E-15)	4.00E-15 (7.86E-12)	7.29E-16 (5.05E-15)
f_{15}	1.82E-03 (4.44E-03)	0.00E+00 (0.00E+00)	2.29E-03 (5.48E-09)	4.83E-03 (8.63E-03)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	2.36E-07 (5.52E-08)
f_{17}	1.03E-02 (7.32E-02)	6.73E-26 (9.27E-26)	1.56E-19 (1.67E-19)	1.59E-32 (1.03E-33)	1.57E-32 (2.79E-48)	7.06E-30 (3.95E-30)	1.57E-32 (2.75E-31)	2.67E-34 (8.25E-38)
f_{18}	2.25E-32 (6.37E-32)	4.37E-24 (3.67E-24)	1.46E-18 (1.33E-18)	4.39E-04 (2.20E-03)	1.35E-32 (5.59E-48)	7.06E-30 (3.00E-29)	2.46E-32 (5.49E-34)	1.35E-34 (2.47E-35)
f_{19}	8.99E+00 (2.29E+00)	0.00E+00 (0.00E+00)	6.97E+00 (3.07E+00)	1.07E+00 (9.90E-01)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)
f_{20}	0.00E+00 (0.00E+00)	NA	2.93E+03 (5.57E+02)	3.84E+02 (2.17E+02)	3.82E-04 (0.00E+00)	NA	7.79E-07 (3.55E-08)	8.94E-22 (1.12E-21)
$w/t/l$ ^e	6/0/2	7/1/4	10/0/0	10/0/0	7/1/2	7/12/3	8/2/3	

^aThe maximum number of FEs of OXDE is 300,000 for each function; ^bthe maximum number of FEs of OPSO, OLPSO-G, and OLPSO-L is 200,000 for each function; ^cthe result of POLBBO is from the conducted experiment in the previous subsection instead of the corresponding reference; ^dNA means the results are not available in the original references; ^e"w/t/l" summarizes the quality of the final results between MTQLBBO and its competitors in this table.

TABLE 8: Comparison with other non-OL-based EAs.

F	CLPSO	CMA-ES	GL-25	SLPSO ^a	MGBGE ^b	iMEABC	MSODPSO-G	MSODPSO-L	MTQLBBO
	Mean (std. dev.)	Mean (std. dev.)	Mean (std. dev.)	Mean (std. dev.)	Mean (std. dev.)	Mean (std. dev.)	Mean (std. dev.)	Mean (std. dev.)	Mean (std. dev.)
f_3	9.10E-03 (2.20E-03)	2.35E-01 (6.07E-02)	1.80E-03 (5.64E-04)	NA	2.14E-03 (1.08E-03)	1.50E-03 (6.77E-04)	NA ^c	NA	0.00E+00 (0.00E+00)
f_4	5.47E+00 (2.45E+00)	5.32E-01 (1.4E+00)	2.12E+01 (7.93E-01)	2.06E+00 (1.23E+01)	1.69E+00 (2.24E+00)	2.81E+01 (5.65E-01)	3.78E-02 (1.65E-01)	1.22E-01 (1.59E-01)	1.43E+00 (6.87E-01)
f_5	8.19E+02 (1.30E+02)	1.44E-26 (3.01E-27)	2.47E-01 (4.82E-01)	7.93E-03 (2.33E-01)	6.10E-11 (2.96E-11)	1.14E-09 (1.60E-09)	1.54E-05 (1.53E-05)	1.02E-02 (7.40E-03)	6.87E-01 (4.52E-01)
f_6	3.14E-04 (4.78E-05)	1.80E-03 (6.90E-03)	2.56E-38 (9.90E-38)	1.35E-26 (3.88E-25)	8.50E-41 (3.38E-41)	7.15E-17 (3.41E-17)	1.46E-48 (2.77E-48)	1.35E-29 (1.55E-29)	0.00E+00 (0.00E+00)
f_7	3.35E+00 (3.56E-01)	3.87E-15 (5.61E-16)	4.04E-02 (2.46E-02)	2.54E-03 (1.86E-02)	1.31E-05 (30.46E-06)	5.13E-06 (2.55E-06)	NA	NA	4.23E-03 (7.23E-06)
f_8	1.90E-03 (7.05E-04)	5.84E-29 (1.58E-29)	2.87E-120 (7.31E-120)	2.78E-50 (8.09E-49)	8.79E-68 (5.21E-69)	1.08E-31 (9.98E-32)	NA	NA	0.00E+00 (0.00E+00)
f_9	2.10E-03 (7.00E-03)	6.49E-29 (1.79E-29)	1.86E-31 (3.53E-31)	NA	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	NA	NA	0.00E+00 (0.00E+00)
f_{13}	2.68E-02 (4.00E-03)	1.95E+00 (1.58E-01)	1.03E-12 (3.05E-12)	3.47E-14 (4.45E-14)	7.69E-15 (0.00E+00)	4.94E-15 (1.57E-15)	3.73E-15 (1.66E-15)	2.90E-15 (9.01E-16)	7.29E-16 (5.05E-15)
f_{15}	1.42E-04 (4.89E-05)	6.57E-04 (2.50E-03)	1.24E-08 (3.91E-08)	2.27E-02 (1.44E-02)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	2.36E-07 (5.52E-08)
f_{17}	4.56E-05 (2.05E-05)	2.40E-29 (4.32E-30)	1.03E-02 (3.27E-02)	1.57E-32 (4.50E-47)	1.57E-32 (0.00E+00)	2.16E-28 (2.53E-28)	1.57E-32 (0.00E+00)	1.57E-32 (0.00E+00)	2.67E-34 (8.25E-38)
f_{18}	1.10E-03 (3.60E-04)	1.40E-03 (3.90E-03)	4.20E-03 (7.40E-03)	NA	1.35E-22 (0.00E+00)	3.36E-27 (3.08E-27)	NA	NA	1.35E-34 (2.47E-35)
f_{19}	9.19E-04 (5.44E-04)	2.44E+02 (3.69E+01)	2.47E+01 (6.49E+00)	0.00E+00 (0.00E+00)	3.98E+00 (2.98E+00)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)
f_{20}	4.87E-01 (1.37E-07) ^d	5.55E+03 (6.81E+02) ^d	3.57E+03 (7.40E+02) ^d	3.82E-04 (7.31E-12)	4.87E-01 (1.09E-12) ^d	3.82E-04 (3.27E-13)	3.82E-04 (6.89E-13)	1.18E+01 (3.61E+01)	8.94E-22 (1.12E-21)
$w/t/f^e$	13/0/0	10/0/3	11/0/2	71/1/2	6/1/3	8/2/3	4/1/3	4/1/3	

^aThe maximum number of FEs of CLPSO is 300,000 for each function; ^bthe maximum number of FEs of MGBGE is 200,000 for each function; ^cNA means the result is not available; ^dthe mean result of this function has been added, 418.9829 $\times n$, to make the global optimum equal to 0; ^e $w/t/f^e$ summarizes the quality of the final results between MTQLBBO and its competitors in this table based on Mann-Whitney U test.

TABLE 9: Optimal results of MTBBO, QOLBBO, and MTQLBBO on all test functions ($D = 30$, $N_{\text{pop}} = 100$).

F	BBO			MTBBO			QOLBBO			MTQLBBO		
	Mean	Standard deviation	Mean	Standard deviation	Mean	Standard deviation	Mean	Standard deviation	Mean	Standard deviation	Mean	Standard deviation
f_1	1.98E+04	1.43E+04	1.23E-24	2.32E-23	+	2.43E-28	7.43E-29	+	3.45E-70	6.24E-72	+	+
f_2	5.12E-05	2.21E-05	5.09E-07	1.23E-08	+	0.00E+00	0.00E+00	=	0.00E+00	0.00E+00	=	+
f_3	2.95E-02	1.12E-03	7.35E-02	9.21E-03	+	6.43E-03	5.23E-04	+	0.00E+00	0.00E+00	+	+
f_4	1.01E+00	7.89E+01	7.23E+01	2.91E+01	+	9.23E+01	3.12E+01	+	1.43E+00	6.87E-01	+	+
f_5	6.50E-01	8.52E+01	3.40E-01	9.19E-02	-	7.35E+00	9.15E-01	+	6.87E-01	4.52E-01	+	+
f_6	8.40E-02	6.72E-02	6.01E-19	6.12E-21	+	9.34E-22	4.92E-24	+	0.00E+00	0.00E+00	+	+
f_7	3.11E+00	5.23E-02	6.39E-02	7.29E-02	+	8.12E-04	4.56E-04	-	4.23E-03	7.23E-06	-	+
f_8	1.34E-01	8.32E-02	9.14E-10	4.23E-08	+	4.87E-09	3.42E-02	+	0.00E+00	0.00E+00	+	+
f_9	1.23E-01	2.21E-02	1.17E-17	8.93E-19	+	1.43E-04	5.43E-06	+	0.00E+00	0.00E+00	+	+
f_{10}	7.42E-02	2.64E-02	9.07E-23	2.12E-20	+	6.87E-34	7.32E-35	+	6.78E-72	8.23E-75	+	+
f_{11}	8.98E-07	1.24E-06	5.34E-33	4.23E-34	+	5.54E-20	4.19E-21	+	8.23E-85	1.43E-83	+	+
f_{12}	4.14E-03	1.03E-03	9.09E-03	1.94E-04	+	8.82E-11	9.71E-12	+	2.18E-71	5.36E-70	+	+
f_{13}	8.82E-02	2.98E-01	4.34E-12	5.87E-13	+	6.94E-09	9.23E-10	+	7.29E-16	5.05E-15	+	+
f_{14}	1.01E-02	2.76E-03	1.23E-10	5.87E-10	+	6.12E-11	9.23E-12	+	5.04E-14	3.93E-15	+	+
f_{15}	3.68E-02	1.98E-02	2.03E-06	1.01E-05	+	3.84E-07	6.90E-08	=	2.36E-07	5.52E-08	=	+
f_{16}	2.89E-02	2.36E-02	4.95E-12	3.29E-03	+	3.13E-14	4.38E-16	+	1.12E-42	2.51E-43	+	+
f_{17}	2.52E-03	3.70E-03	3.56E-12	3.47E-14	+	1.29E-03	3.21E-04	+	2.67E-34	8.25E-38	+	+
f_{18}	2.57E-02	1.21E-02	2.37E-07	2.18E-08	+	1.87E-05	7.34E-06	+	1.35E-34	2.47E-35	+	+
f_{19}	1.21E+01	2.58E+00	8.23E+00	3.23E+00	+	9.74E-02	8.54E-02	+	0.00E+00	0.00E+00	+	+
f_{20}	6.82E-01	5.23E-02	3.82E-01	2.89E-01	+	6.64E+00	1.32E-01	+	8.94E-22	1.12E-13	+	+
f_{21}	9.51E-01	1.23E-01	1.61E-04	2.71E-05	+	1.01E-03	9.48E-04	+	0.00E+00	0.00E+00	+	+
f_{22}	8.63E+00	1.02E+00	6.15E-03	4.23E-04	+	4.14E-04	8.04E-05	+	0.00E+00	0.00E+00	+	+
f_{23}	1.36E+01	1.89E+00	1.03E-18	3.01E-19	+	7.12E-27	5.97E-29	+	1.89E-40	5.69E-39	+	+
f_{24}	5.74E-02	4.23E-02	2.12E-03	5.12E-04	+	6.47E-06	4.79E-07	+	5.98E-09	9.36E-10	+	+
$w/t/l^a$				23/0/1			22/2/1					
$w/t/l^b$				21/0/3			21/0/3					

$w/t/l^a$ summarizes the number of functions where MTQLBBO wins, ties, and loses, compared with MTBBO and QOLBBO; $w/t/l^b$ summarizes the number of functions where MTBBO and QOLBBO win, tie, and lose, compared with BBO.

TABLE 10: Influence of the parameter σ_{MDR} ($D = 30$, $N_{\text{pop}} = 100$).

σ_{MDR}	0.05	0.1	0.2	0.3	0.4	0.5	0.8	1.0
	Mean (std. dev.)	Mean (std. dev.)	Mean (std. dev.)	Mean (std. dev.)	Mean (std. dev.)	Mean (std. dev.)	Mean (std. dev.)	Mean (std. dev.)
f_1	2.16E - 32 (5.36E - 34)	5.97E - 45 (8.77E - 44)	7.42E - 79 (9.49E - 80)	3.45E - 70 (6.24E - 72)	4.39E - 65 (5.83E - 66)	6.85E - 20 (7.59E - 21)	6.71E + 05 (4.44E + 04)	4.85E + 06 (1.60E + 07)
f_2	0.00E + 00 (0.00E + 00)	0.00E + 00 (0.00E + 00)	0.00E + 00 (0.00E + 00)	0.00E + 00 (0.00E + 00)	9.56E - 30 (1.77E - 32)	3.27E - 09 (3.58E - 13)	1.97E + 00 (7.06E + 01)	1.68E + 04 (9.36E + 04)
f_3	7.04E - 05 (6.45E - 05)	2.43E - 10 (9.08E - 12)	0.00E + 00 (0.00E + 00)	0.00E + 00 (0.00E + 00)	1.66E - 03 (1.24E - 04)	5.87E - 01 (5.97E - 03)	1.05E + 02 (6.78E + 02)	4.36E + 06 (7.69E + 06)
f_4	2.94E + 01 (8.16E + 00)	7.78E + 00 (4.96E - 01)	1.23E + 00 (1.25E - 02)	1.43E + 00 (6.87E - 01)	8.54E + 00 (1.10E - 01)	1.40E + 01 (2.53E - 02)	3.52E + 03 (5.33E + 03)	8.09E + 07 (1.29E + 07)
f_5	5.21E + 00 (4.31E - 01)	2.66E - 01 (1.90E - 02)	1.53E - 01 (9.63E - 03)	6.87E - 01 (4.52E - 01)	6.11E - 01 (9.34E - 02)	8.61E + 00 (5.28E - 02)	5.32E + 02 (9.57E + 02)	5.35E + 03 (9.75E + 03)
f_6	6.75E - 22 (3.16E - 24)	8.11E - 38 (9.03E - 40)	5.25E - 47 (8.22E - 50)	0.00E + 00 (0.00E + 00)	4.16E - 41 (4.93E - 43)	8.52E - 20 (8.30E - 22)	6.56E + 03 (7.46E + 03)	8.62E + 05 (6.66E + 06)
f_7	2.33E - 05 (4.76E - 05)	2.27E - 07 (1.39E - 07)	7.40E - 06 (3.81E - 08)	4.23E - 03 (7.23E - 06)	1.91E - 05 (8.28E - 06)	8.06E - 02 (3.05E - 02)	4.20E + 02 (2.65E + 03)	9.67E + 01 (5.87E + 01)
f_8	1.55E - 12 (8.42E - 13)	3.54E - 24 (5.68E - 25)	0.00E + 00 (0.00E + 00)	0.00E + 00 (0.00E + 00)	4.17E - 20 (8.72E - 22)	2.67E - 10 (4.07E - 12)	8.07E + 01 (3.09E + 02)	9.14E + 03 (6.86E + 03)
f_9	4.89E - 08 (9.86E - 06)	3.83E - 17 (8.10E - 13)	0.00E + 00 (0.00E + 00)	0.00E + 00 (0.00E + 00)	1.67E - 18 (4.83E - 15)	6.00E - 13 (1.55E - 14)	7.00E + 02 (1.29E + 03)	1.02E + 04 (9.39E + 04)
f_{10}	9.00E - 36 (6.53E - 34)	2.34E - 53 (5.24E - 42)	5.66E - 74 (5.70E - 42)	6.78E - 72 (8.23E - 75)	4.45E - 36 (9.17E - 38)	6.20E - 21 (4.72E - 21)	9.30E - 02 (8.16E - 01)	7.45E + 00 (9.61E - 01)
f_{11}	3.90E - 18 (9.78E - 18)	2.94E - 34 (4.44E - 32)	4.74E - 69 (3.04E - 67)	8.23E - 85 (1.43E - 83)	5.98E - 56 (9.58E - 57)	4.62E - 30 (4.50E - 32)	5.02E + 00 (4.59E + 01)	1.09E + 04 (1.72E + 04)
f_{12}	6.14E - 12 (3.27E - 19)	6.78E - 35 (6.78E - 36)	6.70E - 55 (6.01E - 56)	2.18E - 71 (5.36E - 70)	1.18E - 35 (4.14E - 34)	2.80E - 08 (5.57E - 09)	1.11E + 01 (7.13E + 00)	8.98E + 01 (9.40E + 00)
f_{13}	5.63E - 08 (9.03E - 09)	7.22E - 06 (7.63E - 06)	9.08E - 10 (5.60E - 12)	7.29E - 16 (5.05E - 16)	2.54E - 16 (1.44E - 17)	3.68E - 08 (6.27E - 08)	3.10E + 01 (1.79E + 02)	8.23E + 01 (2.60E + 01)
f_{14}	1.48E - 09 (6.01E - 09)	9.07E - 12 (1.29E - 12)	1.48E - 21 (2.58E - 26)	5.04E - 14 (3.93E - 15)	3.92E - 54 (5.93E - 53)	1.69E - 06 (9.94E - 08)	1.70E + 01 (1.16E + 01)	6.99E + 00 (3.83E + 00)
f_{15}	6.02E - 09 (3.41E - 07)	4.72E - 20 (1.19E - 18)	0.00E + 00 (0.00E + 00)	2.36E - 07 (5.52E - 08)	5.91E - 34 (1.63E - 32)	2.16E - 07 (9.93E - 08)	5.67E + 02 (5.41E + 03)	1.34E + 01 (4.54E + 01)
f_{16}	1.27E - 13 (5.56E - 12)	1.27E - 16 (6.98E - 15)	4.01E - 18 (2.48E - 20)	1.12E - 42 (2.51E - 43)	2.89E - 23 (5.57E - 24)	4.40E - 12 (7.40E - 14)	7.43E + 02 (6.66E + 03)	2.65E + 04 (1.88E + 04)
f_{17}	4.85E - 02 (6.82E - 03)	7.68E - 09 (6.79E - 09)	7.27E - 19 (7.23E - 17)	2.67E - 34 (8.25E - 38)	3.30E - 25 (7.67E - 22)	1.38E - 10 (8.29E - 11)	4.26E + 03 (2.15E + 03)	1.50E + 09 (3.95E + 08)
f_{18}	6.72E - 04 (8.01E - 04)	9.11E - 10 (9.85E - 11)	7.62E - 23 (3.53E - 24)	1.35E - 34 (2.47E - 35)	6.17E - 31 (1.76E - 30)	6.25E - 05 (4.82E - 05)	1.69E + 04 (2.89E + 05)	6.21E + 08 (2.07E + 08)
f_{19}	8.11E - 01 (2.92E - 02)	6.92E - 19 (4.80E - 20)	2.12E - 30 (5.79E - 31)	0.00E + 00 (0.00E + 00)	3.26E - 50 (4.38E - 49)	4.53E - 07 (3.98E - 08)	3.14E + 03 (9.76E + 04)	7.89E + 03 (9.09E + 03)
f_{20}	8.87E + 00 (7.94E + 00)	7.68E - 04 (1.10E - 05)	4.54E - 13 (1.34E - 12)	8.94E - 22 (1.12E - 23)	7.62E - 04 (5.08E - 04)	6.60E - 01 (1.66E - 02)	5.95E + 04 (2.77E + 04)	1.17E + 05 (1.71E + 04)
f_{21}	5.80E - 02 (2.40E - 03)	6.80E - 05 (5.15E - 06)	4.27E - 14 (5.24E - 14)	0.00E + 00 (0.00E + 00)	9.51E - 12 (7.99E - 13)	6.47E - 02 (8.03E - 03)	9.38E + 04 (7.49E + 04)	7.70E + 02 (5.85E + 02)
f_{22}	2.09E - 04 (1.79E - 05)	6.55E - 06 (5.79E - 07)	6.67E - 24 (6.61E - 24)	0.00E + 00 (0.00E + 00)	3.14E - 14 (4.47E - 15)	5.44E - 07 (2.42E - 08)	2.26E + 02 (1.84E + 02)	2.99E + 04 (9.23E + 03)
f_{23}	7.12E - 27 (8.01E - 28)	3.56E - 20 (5.57E - 21)	9.61E - 21 (2.36E - 24)	1.89E - 40 (5.69E - 39)	6.70E - 10 (9.11E - 10)	1.23E - 02 (9.60E - 03)	4.06E + 00 (1.88E - 01)	3.89E + 01 (2.66E + 00)
f_{24}	1.22E - 06 (5.79E - 07)	7.27E - 08 (5.15E - 09)	1.16E - 11 (6.79E - 12)	5.98E - 09 (9.36E - 10)	4.64E - 05 (5.84E - 04)	7.99E - 01 (1.01E - 01)	3.51E + 00 (1.16E + 00)	5.48E + 01 (3.34E + 01)
$N/24^a$	1/24	2/24	11/24	15/24	1/24	0/24	0/24	0/24

^a"N/24" represents the number of the achieved best solutions out of 24 benchmark functions when σ_{MDR} is taken as the corresponding number of the column.

strategy and QOL strategy exactly contribute to enhancing the solution quality. On the other hand, MTQLBBO has better solution quality than MTBBO and QOLBBO. Compared with MTBBO, MTQLBBO wins on 23 functions and loses only on f_{05} . Compared with QOLBBO, MTQLBBO wins on 22 functions except for f_2 and f_7 . It shows that both multitopology migration operator and QOL strategy cooperate well to improve the solution quality mutually.

5.3. Influence of Three Additional Parameters. Although multitopology migration operator could improve the performance, the effects depend on the determination of the parameter σ_{MDR} . In this subsection, an experiment is conducted to investigate the effect of σ_{MDR} on the performance of MTQLBBO and find an appropriate value for the parameter. σ_{MDR} is set as 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.8, and 1.0, respectively and the experimental results are summarized in Table 10. The last row shows the number of the achieved best solutions when the parameter is taken as the corresponding number of the column. The best solutions are marked in bold black.

The value of σ_{MDR} has much influence on the experimental results and there is a big gap between the best solutions and the worst solutions. All the worst solutions are obtained when the parameter equals 1.0 and the best solutions are acquired when the parameter equals 0.1, 0.2, 0.3, or 0.4. The reason behind the phenomenon is as follows. The migration process is implemented based on ring topology when σ_{MDR} equals 1. Although the population diversity is good, immigrated habitats can hardly get information from good habitats to improve their quality and good habitats cannot guide the whole habitats to a more promising direction. It would lead to pure random search and discourage the solution quality. When σ_{MDR} is decreased to 0.8, more information in good habitats is flowed to the immigrated habitats and it is beneficial for solution quality. For the same reason, better solutions are obtained when σ_{MDR} is decreased to 0.5, 0.4, or even smaller values. However, with the parameter continuing to be decreased, the mentioned defects of basic BBO are exposed gradually. As a result, the solution quality becomes worse again.

Most of the best solutions for unimodal functions, that is, 8 out of 12, are achieved when the parameter equals 0.2 while most of the best ones for multimodal functions, 9 out of 12, are obtained when the parameter equals 0.3. The main reason might be as follows. A larger value of parameter σ_{MDR} helps to improve the population diversity, which is beneficial for solving multimodal functions. Meanwhile, with a smaller value, all habitats have a more unified searching direction toward the promising space and this helps exploit the optimal region more effectively. By observing the overall performance on both unimodal functions and multimodal functions, $\sigma_{MDR} = 0.3$ is recommended. Noting that the experiment is only conducted in the case of $N_{pop} = 100$ and $D = 30$, $\sigma_{MDR} = 0.3$ is not necessarily the best parameter setting when population size or the number of dimensions is set to other values. More experiments should be carried out

to study the influences further. However, this is beyond the scope of the paper and we leave it for future research.

As for the other two parameters, that is, the number of levels (Q) and the number of factors (M) brought by QOL operator, it is obvious that the higher values of Q and M enable the algorithm to have better searching ability. However, the paid expense is extra overhead and computational time. Therefore, there are no best settings for Q and M . The determination of them depends on how you balance precision requirement and acceptable computational time.

6. Conclusion and Future Work

A new variant of BBO, referred to as MTQLBBO, is proposed to solve the global numerical optimization problems. On one hand, the proposed multitopology migration operator can avoid the homogenization of habitats and enhance the exploration ability. On the other hand, QOL operator is able to investigate the solution space thoroughly and systematically so that more competitive solutions can be obtained.

To verify the effectiveness of the proposed algorithm, benchmark functions with various characteristics are employed. Experiment tests, including basic comparison with BBO, comparison with five BBO variant algorithms, comparison with seven OL-based algorithms, comparison with six evolutionary algorithms, the effect of problem dimension, the effects of different modified components, and the influence of three additional parameters, are conducted. The experimental results demonstrate that the improvement strategies exactly enhance the performance of BBO in terms of solution quality, convergence rate, and so on. Moreover, comparisons also show that MTQLBBO performs better than some other evolutionary algorithms on the majority of the selected test functions.

For future work, this paper can be extended in several directions. This work only considers unconstrained optimization problems, and it can be extended to solve multiobjective optimization problems or constrained optimization problems. Another future work can be conducted to solve some practical engineering problems. Additionally, the influences of parameter σ_{MDR} have been preliminarily investigated and its influences can be further studied by varying the population size and even problem dimensions. Besides, its best parameter setting is obtained by manually tuning the value, and this tuning method is tedious and computationally expensive. It is not necessarily accurate or efficient, either. Therefore, whether a new strategy such as adaptive parameter tuning can be applied will be another future direction.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (2016YFB0900100).

References

- [1] Z. Drezner and A. Misevičius, "Enhancing the performance of hybrid genetic algorithms by differential improvement," *Computers and Operations Research*, vol. 40, no. 4, pp. 1038–1046, 2013.
- [2] M. Silva, A. Santos, E. Figueiredo, R. Santos, C. Sales, and J. C. W. A. Costa, "A novel unsupervised approach based on a genetic algorithm for structural damage detection in bridges," *Engineering Applications of Artificial Intelligence*, vol. 52, pp. 168–180, 2016.
- [3] S. H. Ling, K. Y. Chan, F. H. Leung, F. Jiang, and H. Nguyen, "Quality and robustness improvement for real world industrial systems using a fuzzy particle swarm optimization," *Engineering Applications of Artificial Intelligence*, vol. 47, pp. 68–80, 2016.
- [4] B. Haddar, M. Khemakhem, S. Hanafi, and C. Wilbaut, "A hybrid quantum particle swarm optimization for the Multi-dimensional Knapsack Problem," *Engineering Applications of Artificial Intelligence*, vol. 55, pp. 1–13, 2016.
- [5] K. C. Tan, C. K. Goh, A. A. Mamun, and E. Z. Ei, "An evolutionary artificial immune system for multi-objective optimization," *European Journal of Operational Research*, vol. 187, no. 2, pp. 371–392, 2008.
- [6] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [7] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.
- [8] S. Tabakhi, P. Moradi, and F. Akhlaghian, "An unsupervised feature selection algorithm based on ant colony optimization," *Engineering Applications of Artificial Intelligence*, vol. 32, pp. 112–123, 2014.
- [9] A. K. Dwivedi, S. Ghosh, and N. D. Londhe, "Low power FIR filter design using modified multi-objective artificial bee colony algorithm," *Engineering Applications of Artificial Intelligence*, vol. 55, pp. 58–69, 2016.
- [10] N. Imanian, M. E. Shiri, and P. Moradi, "Velocity based artificial bee colony algorithm for high dimensional continuous optimization problems," *Engineering Applications of Artificial Intelligence*, vol. 36, pp. 148–163, 2014.
- [11] S. Bandyopadhyay, S. Saha, U. Maulik, and K. Deb, "A simulated annealing-based multiobjective optimization algorithm: AMOSA," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 3, pp. 269–283, 2008.
- [12] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.
- [13] D. Simon, R. Rarick, M. Ergezer, and D. Du, "Analytical and numerical comparisons of biogeography-based optimization and genetic algorithms," *Information Sciences*, vol. 181, no. 7, pp. 1224–1248, 2011.
- [14] I. Boussaïd, A. Chatterjee, P. Siarry, and M. Ahmed-Nacer, "Biogeography-based optimization for constrained optimization problems," *Computers & Operations Research*, vol. 39, no. 12, pp. 3293–3304, 2012.
- [15] H. Ma, "An analysis of the equilibrium of migration models for biogeography-based optimization," *Information Sciences*, vol. 180, no. 18, pp. 3444–3464, 2010.
- [16] A. Bhattacharya and P. K. Chattopadhyay, "Biogeography-based optimization for different economic load dispatch problems," *IEEE Transactions on Power Systems*, vol. 25, no. 2, pp. 1064–1077, 2010.
- [17] K. Jamuna and K. S. Swarup, "Multi-objective biogeography based optimization for optimal PMU placement," *Applied Soft Computing*, vol. 12, no. 5, pp. 1503–1510, 2012.
- [18] G. Xiong, D. Shi, and X. Duan, "Multi-strategy ensemble biogeography-based optimization for economic dispatch problems," *Applied Energy*, vol. 111, pp. 801–811, 2013.
- [19] P. K. Roy, S. P. Ghoshal, and S. S. Thakur, "Biogeography based optimization for multi-constraint optimal power flow with emission and non-smooth cost function," *Expert Systems with Applications*, vol. 37, no. 12, pp. 8221–8228, 2010.
- [20] G. Xiong, Y. Li, J. Chen, D. Shi, and X. Duan, "Polyphyletic migration operator and orthogonal learning aided biogeography-based optimization for dynamic economic dispatch with valve-point effects," *Energy Conversion and Management*, vol. 80, pp. 457–468, 2014.
- [21] X. Li, J. Wang, J. Zhou, and M. Yin, "A perturb biogeography based optimization with mutation for global numerical optimization," *Applied Mathematics and Computation*, vol. 218, no. 2, pp. 598–609, 2011.
- [22] H. Ma and D. Simon, "Blended biogeography-based optimization for constrained optimization," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 3, pp. 517–525, 2011.
- [23] G. Xiong, D. Shi, and X. Duan, "Enhancing the performance of biogeography-based optimization using polyphyletic migration operator and orthogonal learning," *Computers and Operations Research*, vol. 41, pp. 125–139, 2014.
- [24] X. Li and M. Yin, "Multi-operator based biogeography based optimization with mutation for global numerical optimization," *Computers & Mathematics with Applications*, vol. 64, no. 9, pp. 2833–2844, 2012.
- [25] G. Wang, L. Guo, H. Duan, H. Wang, L. Liu, and M. Shao, "Hybridizing harmony search with biogeography based optimization for global numerical optimization," *Journal of Computational and Theoretical Nanoscience*, vol. 10, no. 10, pp. 2312–2322, 2013.
- [26] I. Boussaïd, A. Chatterjee, P. Siarry, and M. Ahmed-Nacer, "Two-stage update biogeography-based optimization using differential evolution algorithm (DBBO)," *Computers & Operations Research*, vol. 38, no. 8, pp. 1188–1198, 2011.
- [27] M. Ergezer, D. Simon, and D. Du, "Oppositional biogeography-based optimization," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '09)*, pp. 1009–1014, San Antonio, Tex, USA, October 2009.
- [28] D. Easley and J. Kleinberg, *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*, Cambridge University Press, Cambridge, UK, 2010.
- [29] A. C. Atkinson and A. N. Donev, *Optimum Experimental Designs*, Clarendon Press, Oxford, UK, 1992.
- [30] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, 1999.
- [31] F. Kang, J. Li, and Z. Ma, "Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical functions," *Information Sciences*, vol. 181, no. 16, pp. 3508–3531, 2011.
- [32] W. Gong, Z. Cai, C. X. Ling, and H. Li, "A real-coded biogeography-based optimization with mutation," *Applied Mathematics and Computation*, vol. 216, no. 9, pp. 2749–2758, 2010.

- [33] Y. Wang, Z. Cai, and Q. Zhang, "Enhancing the search ability of differential evolution through orthogonal crossover," *Information Sciences*, vol. 185, pp. 153–177, 2012.
- [34] W. Gong, Z. Cai, and L. Jiang, "Enhancing the performance of differential evolution using orthogonal design method," *Applied Mathematics and Computation*, vol. 206, no. 1, pp. 56–69, 2008.
- [35] S.-Y. Ho, H.-S. Lin, W.-H. Liauh, and S.-J. Ho, "OPSO: orthogonal particle swarm optimization and its application to task assignment problems," *IEEE Transactions on Systems, Man, and Cybernetics A: Systems and Humans*, vol. 38, no. 2, pp. 288–298, 2008.
- [36] Z.-H. Zhan, J. Zhang, Y. Li, and Y.-H. Shi, "Orthogonal learning particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 6, pp. 832–847, 2011.
- [37] Q. Feng, S. Liu, G. Tang, L. Yong, and J. Zhang, "Biogeography-based optimization with orthogonal crossover," *Mathematical Problems in Engineering*, vol. 2013, pp. 1–20, 2013.
- [38] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.
- [39] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.
- [40] C. García-Martínez, M. Lozano, F. Herrera, D. Molina, and A. M. Sánchez, "Global and local real-coded genetic algorithms based on parent-centric crossover operators," *European Journal of Operational Research*, vol. 185, no. 3, pp. 1088–1113, 2008.
- [41] C. Li, S. Yang, and T. T. Nguyen, "A self-learning particle swarm optimizer for global optimization problems," *IEEE Systems, Man, and Cybernetics Society. Part B: Cybernetics*, vol. 42, no. 3, pp. 627–646, 2012.
- [42] H. Wang, S. Rahnamayan, H. Sun, and M. G. H. Omran, "Gaussian bare-bones differential evolution," *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 634–647, 2013.
- [43] X. Y. Zhou, M. Y. Wang, and J. L. Zuo, "An improved multi-strategy ensemble artificial bee colony algorithm with neighborhood search," in *Proceedings of the 23rd International Conference on Neural Information Processing (ICONIP '16)*, Kyoto, Japan, 2016.
- [44] Q. Qin, S. Cheng, Q. Zhang, Y. Wei, and Y. Shi, "Multiple strategies based orthogonal design particle swarm optimizer for numerical optimization," *Computers & Operations Research*, vol. 60, pp. 91–110, 2015.