



imBBO: An Improved Biogeography-Based Optimization Algorithm

Kai Shi^{1,2}, Huiqun Yu^{1(✉)}, Guisheng Fan^{1(✉)}, Xingguang Yang¹,
and Zheng Song³

¹ Department of Computer Science and Engineering,
East China University of Science and Technology, Shanghai 200237, China
BH4AWS@163.com, xingguang2955@163.com,
{yhq, gsfan}@ecust.edu.cn

² Shanghai Key Laboratory of Computer Software Evaluating and Testing,
Shanghai 201112, China

³ The Third Research Institute of the Ministry of Public Security,
Shanghai, China
songzheng@stars.org.cn

Abstract. Biogeography based Optimization (BBO) is a new evolutionary optimization algorithm based on the science of biogeography for global optimization. However, its direct-copying-based migration and random mutation operators make it easily possess local exploitation ability. To enhance the performance of BBO, we propose an improved BBO algorithm called imBBO. A hybrid migration operation is designed to further improve the population diversity and enhance the algorithm exploration ability. Empirical results demonstrate that our imBBO effectively gains the high optimization performance by comparing with the original BBO and three BBO variants for 23 out of 30 CEC'2017 benchmarks. Moreover, our imBBO presents a faster convergence speed.

Keywords: Hybrid migration · Biogeography-based Optimization · Global optimization

1 Introduction

Optimization problems are of increasing importance in modern science and engineering fields, especially for the global continuous optimization problem. During the past decade, they have turned to be more complicated and diversified commensurate with the unceasing progress of science and technology [41]. The major challenge of the global continuous optimization is that the problems to be optimized may have many local optima. This issue is particularly challenging when the dimension is high [15]. Thus, numerous optimization techniques have been advanced to handle these problems [2, 20]. Currently, the most popular method is meta-heuristics, such as Genetic Algorithms (GAs) [6], Evolution Strategy (ES) [42], Particle Swarm Optimization (PSO) [21], Differential Evolution (DE) [40], Ant Colony Optimization (ACO) [9] and Biogeography-based Optimization (BBO) [36].

Biogeography-based Optimization (BBO) proposed by Dan Simon in 2008 is a novel Evolutionary Algorithm (EA) for global optimization based on the equilibrium theory of island biogeography. Different from other population-based algorithms, in BBO, poor solutions can improve their qualities by accepting new features from good ones [19]. Concretely, a habitat in BBO algorithm is called an island and a group of habitats construct the ecosystem. The habitat suitability index (HSI) presents the habitability of an island, which can be analogized as the fitness values to evaluate the problems. The HSI is always determined by a series of independent decision variables called Suitability Index Variable (SIV), such as the features of temperature, disease and earthquake in real world. Specially, the most important characteristics in BBO are migration and mutation operation. The migration, including immigration and emigration process, is designed to conditionally share the SIV information among habitats in an ecosystem. The mutation is a probabilistic operation that can randomly modify the habitat SIVs based on the a priori probability of the habitat.

Similar to other EAs, the BBO algorithm has also some certain weaknesses. The probabilistic migration can make the population share different information among solutions to guide good exploitation ability [14]. However, *its directcopying-based migration and random mutation operators make BBO lack enough exploration ability and cannot improve the diversity of population* [15, 41]. Although its convergence speed is relatively fast at the beginning of the evolutionary process, it easily falls into local optima. To mitigate these weakness, an improved BBO variant (called imBBO) is proposed in this paper.

Our contributions of this paper are summarized as follows:

- An imBBO algorithm is proposed to mitigate part weakness of BBO for global optimization problems, which is composed of a hybrid migration operation and a scalable direction mutation operation.
- A hybrid migration operation is designed based on the combination of the DE theory and a scalable method, which helps our imBBO to improve the population diversity and enhance the algorithm exploration ability.
- We conduct an optimization performance comparison among our imBBO, the original BBO and three BBO variants. Empirical results demonstrate that our algorithm effectively outperforms the competitors for 23 out of 30 CEC'2017 benchmarks. Moreover, our imBBO presents a faster convergence speed.

The rest of this paper is organized as follows. Related works are discussed in Sect. 2. Section 3 defines the problem formulation. Section 4 proposes our imBBO algorithm. We describe our experimental setup in Sect. 5 and present the results in Sect. 6. Section 7 discusses our experimental results by answering two research questions. Section 8 concludes the paper.

2 Related Work

For global optimization problem, many references have showed that meta-heuristic algorithms [30], including the categories of nature-inspired meta-heuristic algorithms, physics-based algorithms and swarm-based methods, become much more popular to solve these problems or various engineering applications [28]. Part of typical algorithms include the first category of Genetic Algorithms (GAs) [6], Genetic Programming (GP) [29] and Biogeography-based Optimization (BBO) [36], the second category of Simulated Annealing (SA) [5], Gravitational Search Algorithm (GSA) [33] and Artificial Chemical Reaction Optimization Algorithm (ACROA) [1] and the third category of Particle Swarm Optimization Algorithm (PSO) [21], Ant Colony Optimization (ACO) [9] and Artificial Bee Colony Algorithm (ABC) [18].

This paper focuses on the Biogeography-based Optimization [36], which shows the excellent performance on various unconstrained or constrained benchmarks [7]. During the last decade, more BBO variants are proposed and are available in literature [8, 24, 37]. On the one hand, Mehmet et al. proposed an oppositional Biogeography-based Optimization [11] which is composed of the opposition based learning and migration rates of original BBO in 2009. To enhance the mutation operation, a real coded BBO algorithm [15] was proposed in 2010. Similarly, there are some additional BBO variants that the migration operation is also modified, such as the literatures [13, 17, 19, 20, 22, 41]. Moreover, Haiping Ma analyzed the equilibrium of migration models [23]. Haiping Ma and Dan Simon discussed migration models using Markov theory [25] and the blended BBO [26] for constrained optimization. Considering the random initial generation of population, Simon et al. focused on the re-initialization and local search in Linearized BBO [38].

On the other hand, Wenying et al. proposed a combination DE/BBO [14] that combines the DE algorithm with BBO to improve the searching capability. Moreover, a hybrid BBO [27] was proposed in 2014, which combine the various EAs with BBO in different ways. That is, two types of hybridization named as iteration-level hybridization and algorithm-level hybridization are used.

Naturally, BBO has been widely applied to solve the real-world and engineering problems, such as the sensor selection [36], power system optimization [32], economic load dispatch [4] and antenna design [12, 39].

3 Problem Definition

For the different areas of engineering or scientific application, the optimization problems should be solved to achieve approximate optimal solutions. However, different problems may have the special constraints and conflicting objectives. So an effective method is to design a global search algorithm to find these optimal or near-optimal solutions.

Without loss of generality, the mathematical expression presents that the unconstrained continuous global minimization problem can formulize as a pair (S, f) , where the $S \subseteq R^D$ is a bounded set based on R^D and $f : S \rightarrow R$ is a Ddimensional real-valued function. Finally, the purpose of these problems is to find a point $X^* \in S$ [15] and the

X^* belongs to a D -dimensional ($D \in \{1, 2, 3, \dots\}$) vector. Thus, the $f(X^*)$ value is the global minimum on $S[1]$. More specifically, it is required to find an $X^* \in S$, as the formula 1:

$$\forall X \in S : f(X^*) \leq f(X) \quad (1)$$

Note that the function f does not need to be continuous but it must be bounded. Moreover, the different variables contain the different bound in realworld constraint optimization problems. We only consider the unconstrained and continuous functions optimization in this paper.

As mentioned in above, the major challenge in global continuous optimization problem is that the optimization problems to be solved may easily lead EAs to trap into local optima. These issues are particularly challenging when the problem has the high dimension. So, one of the effective methods is that different EAs adopt the special search and modification method for different optimization problems.

4 Our Approach

In this section, we firstly introduce our algorithm implementation in Algorithm 1. Then, a hybrid migration operation is discussed in detail.

Algorithm 1. The main algorithm structure of imBBO	
Input:	objectives, constraint condition, maximum iterations and migration rate, population size NP , different algorithm parameters, etc.
Output:	the optimal objective fitness, solutions and iteration optimum curves
1:	Begin
2:	Generating the initial population
3:	Setting relevant algorithm Parameters
4:	Evaluating fitness values for each individual in NP
5:	while the halting criterion is not satisfied do
6:	Elites inheritance operation
7:	for each individual do
8:	modifying the number of species
9:	end
10:	Calculating migration rate λ_i and μ_i for each habitat X_i
11:	Modifying the population size with a hybrid migration operation shown in Section 4.2
12:	Enhancing exploration with the mutation operation referring from [34]
13:	Other mechanisms, such as data validation and boundary checking
14:	Evaluating the fitness for each individual in NP
15:	Elites selection operation
16:	end
17:	end

4.1 The Structure of imBBO

Our algorithm structure is showed in Algorithm 1, which is mainly composed of a hybrid migration operation introduced in Sect. 4.2 and a scalable direction mutation operation derived from our previous work [34].

Concretely, the hybrid migration operation combines the DE theory with a scalable method to further improve the population diversity, exploit the population information and enhance the algorithm exploration ability. Moreover, considering the excellent algorithm performance of iCPBBOCO [34], we still try to adopt the same mutation operation into our imBBO algorithm.

To make a fair comparison, all parameters values of our algorithm are set by referring to [14, 34, 36], which is shown in Table 1. The modification of migration and mutation operation are marked from Line 11 to Line 12 in Algorithm 1.

4.2 Hybrid Migration Operation

A hybrid migration operation is deployed in our imBBO algorithm, which combines the DE theory [14, 31] with the component of migration operation from our previous iCPBBOCO algorithm [34]. The core idea of this proposed method is that good solutions would be less destroyed, while poor solutions can accept a lot of new features from good solutions. Furthermore, the implementation is listed in formula 2, which is composed of the relevant DE theory and a scalable method in formula 3 and formula 4, respectively. To enhance the diversity, we define a parameter ξ to further guide the specific migration operation.

$$Migration \begin{cases} \text{Formual } 3 & rand \leq \xi \\ \text{Formual } 4 & rand > \xi \end{cases} \tag{2}$$

For operations of the DE theory, the main mathematical principle is illustrated in formula 3. Recently, the DE algorithm is used for global permutation based combinatorial optimization problems [14], successfully. It is good at exploring the search space and locating the region of global minimum. It uses the distance and direction information from the current population and the characteristics of problem to guide the further search. So, we try to adopt it to our migration operation.

$$U_i(j) = \begin{cases} U_i(j) + f_1 * d_1 + f_2 * d_2 & rand \leq \kappa \\ U_i(j) - f_1 * d_3 - f_2 * d_2 & rand > \kappa \end{cases} \tag{3}$$

$$d_1 = U_1(j) - U_{r_1}(j), d_2 = U_{r_2}(j) - U_{r_3}(j)$$

$$d_3 = U_{NP}(j) - U_{r_1}(j)$$

To inherit the excellent performance, we define the standard parameters setting of DE theory in our algorithm. In formula 3, the variable $U_i(j)$ presents the $j - th$ decision variable in $i - th$ solution. According to the principle of DE theory [14, 16, 31], we randomly select three additional population r_1, r_2 and r_3 , where $r_1 \neq r_2 \neq r_3$. A series of variables, such as d_1, d_2 and d_3 , are the difference values of corresponding populations.

Then, two scale factors of f_1 and f_2 are related with the lower/upper bound for immigration probability per individual, which indicates that how much amount of differential variation [10] will influence on target. Similarly, we also define the guide parameter κ .

$$U_i(j) = \begin{cases} \partial * U_i(j) + (1 - \partial) * U_j(j) & rand \leq \varphi \\ (1 - \partial) * U_i(j) + \partial * U_j(j) & rand > \varphi \end{cases} \quad (4)$$

In description of formula 4, it is an additional method to perform the migration operation. The new offspring solution comes from a different combination of the source solution of U_i and a target solution of U_j . We define a scalable factor ∂ to decide the migration size of individuals. To maintain the stability, we exchanges the scalable coefficient ∂ among each other. That is, the purpose is to apply for asymmetrical migration, further enhance potential population diversity and exploit the population information.

5 Experimental Setup

In this section, we describe settings of our conducted experiments to evaluate our algorithm performance. Concretely, we detail the benchmarks, the performance criteria and algorithms setting.

5.1 Benchmark Functions

To evaluate the performance of the proposed algorithm, 30 benchmarks from CEC'2017 [3] which is the latest set of benchmarks are employed in our experiments. These benchmarks are divided into four categories, including unimodal functions (F01-F03), simple multimodal functions (F04-F10), hybrid functions (F11-F20) and composition functions (F21-F30). *The more complex benchmarks in evaluation process, the better performance superiority of competition will be shown.*

Currently, these benchmarks from CEC'2017 [3] are related with the real-parameter single objective optimization without making use of the exact equations of the test functions. Some benchmarks are developing with novel features such as new basic problems, composing test problems by extracting features dimension-wise from several problems, graded level of linkages, rotated trap problems, and so on.

5.2 Performance Metrics

We evaluate the performance of our algorithm in terms of two aspects [14, 34], which are described as follows in detail.

- **Error:** The error value of a solution X is defined as $f(X) - f(X^*)$, where X^* is the standard, global optimization of the objective. The minimum average error is recorded when the maximum number of fitness function evaluations ($maxFEs$) reached in 30 independent runnings. Moreover, we also calculate the average median values.

- **Convergence graphs:** We present the convergence speed of our algorithm compared to the competitors. In order to observably demonstrate differences, we recalculate all of error values by the logarithm (log) function.

5.3 Algorithm Settings

Table 1 lists the values of key parameters of imBBO. To enable a fair performance comparison to the competitors, we use the same settings as the ones reported by [14, 15, 19, 20, 34, 36]. All of algorithms are developed by Matlab. We generate the initial population by uniform random initialization within the search space. According to [3], we set the problem dimension $D = 100$. That is, the search rang is $[-100, 100]$. Moreover, we define the $maxFEs$ is $10000 * 10$. All algorithms need to be terminated when reaching $maxFEs$ or the error value is smaller than 10^{-08} .

Table 1. Overview of parameters setting of imBBO and BBO variants.

Parameter (imBBO and Variant BBOs)	Default
Population size: NP	100
Habitat modification probability	1.0
Mutation probability	0.005
Maximum migration rate: $I\&E$	1.0 & 1.0
Number of elites	2
Migration scaling factor: ∂	0.3
Migration guide parameter of ξ, κ, φ	0.5

The competitors in our experiments include original BBO [36] and three BBO variants. They are the MOBBO [20], the PBBO [19] and the RCBBO [15] algorithms.

5.4 Measurement Settings

All measurements of each algorithm are performed on Windows 7 (64bit) Machine with Intel Core i5-4690 K CPU 3.5 GHz and 16 GB RAM. Each algorithm variant is compute-bound and not memory intensive.

To reduce measurement fluctuations caused by randomness (e.g., the randomness of performing migration and mutation), we independently execute each algorithm 30 times. We take both the median and mean values of the measurements for analysis.

6 Results

In this section, we report the experiment results in detail. we aim at answering the following two research questions.

RQ1: What is the performance of our imBBO, compared to the BBO and three BBO variants? (Section 6.1)

RQ2: How fast is the convergence speed of our imBBO? (Section 6.2).

6.1 Performance Results

Table 2 records the experimental results of the comparison between our imBBO algorithm and competitors when applied to 30 CEC'2017 benchmarks in reaching to *maxFEs*. The columns BBO, imBBO, MOBBO, PBBO and RCBBO list the measured results of each algorithm. We report the median and mean values for 30 executions of each algorithm. The rows of the table record the measured details for each benchmark. Moreover, we highlight the median and mean values in bold, which are the best for each benchmark.

Our experimental results reveal that our imBBO outperforms the competitors for 23 out of 30 CEC'2017 benchmarks, except for F1, F3, F4, F7, F12, F17 and F24. Analyzing the experimental results, our algorithm achieve the better algorithm performance of the hybrid and composition functions by comparing with the unimodal and simple multimodal functions. Hence, we conjecture that the components of our imBBO make an effect to be the complex searching.

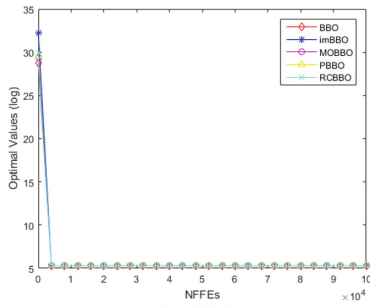
6.2 Convergence Speed

It is interesting to understand the convergence speed of our approach compared to the others. The convergence is an important metric to illustrate whether or not a algorithm has reached to the steady state.

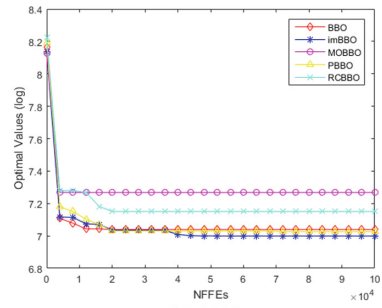
Our experimental results demonstrate that the convergence speed of imBBO is much faster than other competitors, that part of representative curves are shown in Fig. 1 (blue line).

Table 2. Experimental results of the best imBBO algorithm and four competitors (BBO, MOBBO, PBBO and RCBBBO) for CEC'2017 continuous optimization benchmarks. The number in gray indicates a better case than others.

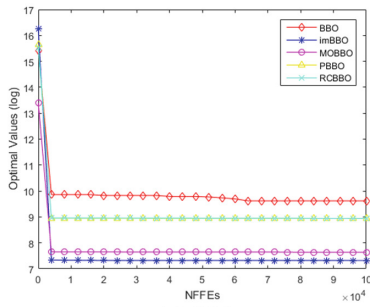
Fun	BBO			imBBO			MOBBO			PBBO			RCBBBO		
	Mean	Median	Mean	Mean	Median	Mean	Mean	Median	Mean	Median	Mean	Median	Mean	Median	
F1	9.78546559e+05	9.78546559e+05	4.41817176e-03	7.15077575e+03	2.53159472e+03	2.53170605e+03	4.16124086e+03	4.16137624e+03	1.71890871e+03	2.00000000e+02	2.00000000e+02	1.71890613e+03	2.00000000e+02	2.00000000e+02	
F2	2.00000000e+02	2.00000000e+02	2.00000000e+02	2.00000000e+02	2.00000000e+02	2.00000000e+02	2.00000000e+02	2.00000000e+02	2.00000000e+02	2.00000000e+02	2.00000000e+02	2.00000000e+02	2.00000000e+02	2.00000000e+02	
F3	3.02100739e+02	3.03158585e+02	3.01816730e+02	3.03558234e+02	3.00001702e+02	3.00006563e+02	3.00000178e+02	3.0000796e+02	3.00000027e+02	3.00000000e+02	3.00000000e+02	3.0000226e+02	3.00000000e+02	3.00000000e+02	
F4	4.00346418e+02	9.33701140e+02	4.03637302e+02	5.43359455e+02	4.00071176e+02	2.76890951e+03	4.00040753e+02	7.48551891e+02	4.00040753e+02	5.07299636e+02	5.24968718e+02	4.00041288e+02	7.68668493e+02	7.68668493e+02	
F5	5.06479857e+02	5.32410313e+02	5.03993244e+02	5.913300152e+02	5.27195472e+02	6.1746487e+02	5.00000179e+02	5.00000179e+02	5.00000179e+02	5.00000179e+02	5.00000179e+02	5.00000179e+02	5.00000179e+02	5.00000179e+02	
F6	6.00770668e+02	6.00770668e+02	6.00031748e+02	6.00031748e+02	6.00000002e+02	6.00000002e+02	6.00000002e+02	6.00000002e+02	6.00000002e+02	6.00000002e+02	6.00000002e+02	6.00000002e+02	6.00000002e+02	6.00000002e+02	
F7	7.17307487e+02	7.65299560e+02	7.14040077e+02	8.92635957e+02	8.14924371e+02	8.75000612e+02	8.75000612e+02	8.75000612e+02	8.75000612e+02	8.75000612e+02	8.75000612e+02	8.75000612e+02	8.75000612e+02	8.75000612e+02	
F8	8.07056264e+02	8.30904774e+02	8.04645032e+02	8.92635957e+02	9.00000002e+02	9.00000002e+02	9.00000002e+02	9.00000002e+02	9.00000002e+02	9.00000002e+02	9.00000002e+02	9.00000002e+02	9.00000002e+02	9.00000002e+02	
F9	9.00230815e+02	9.00230815e+02	9.00006111e+02	9.00000002e+02	9.00000002e+02	9.00000002e+02	9.00000002e+02	9.00000002e+02	9.00000002e+02	9.00000002e+02	9.00000002e+02	9.00000002e+02	9.00000002e+02	9.00000002e+02	
F10	1.14053397e+03	1.69256700e+03	1.09599842e+03	3.87124069e+03	1.43290929e+03	3.44339092e+03	1.12536302e+03	1.53097514e+03	1.27499313e+03	1.53097514e+03	1.53097514e+03	1.60924493e+03	1.60924493e+03	1.60924493e+03	
F11	1.10434240e+03	1.10488514e+03	1.10328536e+03	1.10418539e+03	1.10670450e+03	1.10673822e+03	1.10329167e+03	1.10374043e+03	1.10329167e+03	1.10329167e+03	1.10329167e+03	1.10329167e+03	1.10329167e+03	1.10329167e+03	
F12	1.48175677e+03	1.32795682e+05	1.00555983e+04	1.01425263e+04	1.32226031e+04	1.14057230e+04	1.18658035e+04	1.19145648e+04	8.42452020e+03	1.18658035e+04	1.18658035e+04	8.42452020e+03	1.18658035e+04	8.42452020e+03	
F13	6.55796090e+04	8.42129974e+04	1.41701869e+03	1.52490864e+03	5.45733377e+03	5.45767596e+03	7.48492112e+03	7.48649670e+03	7.48492112e+03	7.48492112e+03	7.48492112e+03	7.48492112e+03	7.48492112e+03	7.48492112e+03	
F14	1.00071584e+04	1.00071584e+04	1.4121283e+03	4.72304439e+03	4.72361057e+03	5.60220131e+03	5.60220131e+03	5.60246475e+03	5.60220131e+03	5.60220131e+03	5.60246475e+03	5.60220131e+03	5.60246475e+03	5.60220131e+03	
F15	1.50526300e+04	1.74918763e+04	1.50856460e+03	5.09200856e+03	2.07423301e+03	2.09189423e+03	7.63392195e+03	7.63392195e+03	7.63392195e+03	7.63392195e+03	7.63392195e+03	7.63392195e+03	7.63392195e+03	7.63392195e+03	
F16	1.69425497e+03	2.20331472e+03	1.60051753e+03	2.37628643e+03	1.75293246e+03	3.07561961e+03	1.70538053e+03	2.14134763e+03	1.70538053e+03	1.70538053e+03	2.14134763e+03	1.70538053e+03	2.14134763e+03	2.14134763e+03	
F17	1.70881113e+03	1.70884909e+03	1.71495833e+03	1.71282842e+03	1.71774219e+03	1.71830341e+03	1.702333694e+03	1.70233694e+03	1.70233694e+03	1.70233694e+03	1.70233694e+03	1.70233694e+03	1.70233694e+03	1.70233694e+03	
F18	3.97270951e+04	5.79282919e+04	1.94085018e+03	2.28397015e+03	1.96776333e+04	2.07415248e+04	7.93578912e+03	8.246099658e+03	7.93578912e+03	8.246099658e+03	7.93578912e+03	8.246099658e+03	7.93578912e+03	8.246099658e+03	
F19	1.44886269e+04	1.48099770e+04	1.90225667e+03	1.10233848e+04	1.10233848e+04	1.11814860e+04	7.11491750e+03	7.18647289e+03	7.11491750e+03	7.18647289e+03	7.18647289e+03	7.11491750e+03	7.18647289e+03	7.11491750e+03	
F20	2.00924831e+03	2.04420085e+03	2.00003572e+03	2.39494215e+03	2.00826263e+03	2.50251636e+03	2.00304019e+03	2.02659547e+03	2.00304019e+03	2.00304019e+03	2.02659547e+03	2.00304019e+03	2.02659547e+03	2.00304019e+03	
F21	2.30685508e+03	2.34346330e+03	2.26815842e+03	2.38344694e+03	2.28357371e+03	2.34850391e+03	2.30426053e+03	2.33671683e+03	2.30426053e+03	2.30426053e+03	2.33671683e+03	2.30426053e+03	2.33671683e+03	2.30426053e+03	
F22	2.30198510e+03	2.46215658e+03	2.27992970e+03	2.41139681e+03	2.29967460e+03	3.45666747e+03	2.30355774e+03	2.48318293e+03	2.29967460e+03	2.30355774e+03	2.48318293e+03	2.29967460e+03	2.30355774e+03	2.29967460e+03	
F23	2.61084751e+03	2.64257156e+03	2.60599957e+03	2.69254276e+03	2.61590866e+03	3.08815452e+03	2.61133148e+03	2.63063551e+03	2.61133148e+03	2.63063551e+03	2.61064042e+03	2.63063551e+03	2.61064042e+03	2.63063551e+03	
F24	2.73783700e+03	2.79252244e+03	2.73783700e+03	2.82845379e+03	2.74936662e+03	3.04478493e+03	2.73722586e+03	2.7454706e+03	2.73722586e+03	2.73722586e+03	2.73671628e+03	2.73722586e+03	2.73671628e+03	2.78472691e+03	
F25	2.93755468e+03	3.22498366e+03	2.92950040e+03	3.06733172e+03	2.93207429e+03	3.85377698e+03	2.93407338e+03	3.20763851e+03	2.93407338e+03	3.20763851e+03	2.93769720e+03	2.93769720e+03	2.93769720e+03	3.18967729e+03	
F26	3.09152760e+03	3.411706209e+03	3.08918725e+03	3.12398320e+03	2.93162574e+03	4.83753566e+03	2.93812322e+03	3.36356031e+03	2.93812322e+03	3.36356031e+03	2.93472024e+03	3.36356031e+03	2.93472024e+03	3.38539297e+03	
F27	3.12915276e+03	3.141308059e+03	3.11009000e+03	3.38377055e+03	3.32476041e+03	4.22682020e+03	3.32606876e+03	3.50122091e+03	3.32606876e+03	3.50122091e+03	3.29696272e+03	3.32606876e+03	3.29696272e+03	3.49274291e+03	
F28	3.23216686e+03	3.51546787e+03	3.19920000e+03	3.85377055e+03	3.62140358e+03	4.83003860e+03	3.15871979e+03	3.66233126e+04	3.15871979e+03	3.66233126e+04	3.15871979e+03	3.66233126e+04	3.15871979e+03	3.98577447e+04	
F29	3.15241031e+03	5.62143036e+03	3.13952104e+03	4.83003860e+03	3.15871979e+03	3.66233126e+04	3.15871979e+03	3.66233126e+04	3.15871979e+03	3.66233126e+04	3.15871979e+03	3.66233126e+04	3.15871979e+03	3.98577447e+04	
F30	7.24427257e+03	7.40424362e+03	5.16732890e+03	5.50406827e+03	1.46830694e+04	1.92864281e+04	6.24876854e+03	8.50512952e+03	5.16732890e+03	6.24876854e+03	8.50512952e+03	5.16732890e+03	6.24876854e+03	9.27910406e+03	



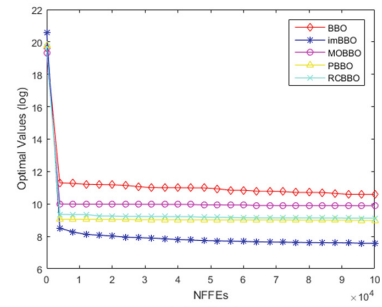
(a) F02



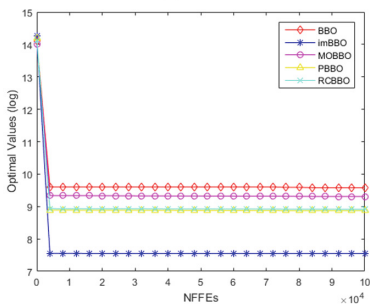
(b) F10



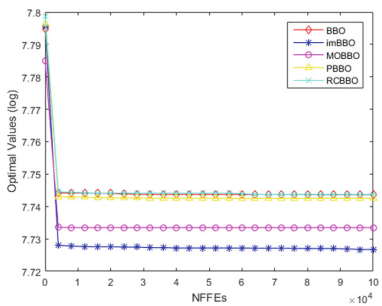
(c) F15



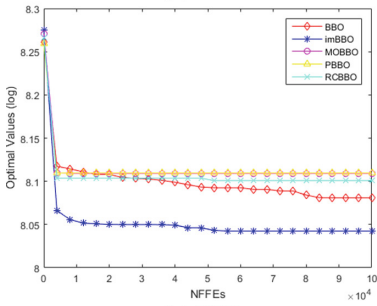
(d) F18



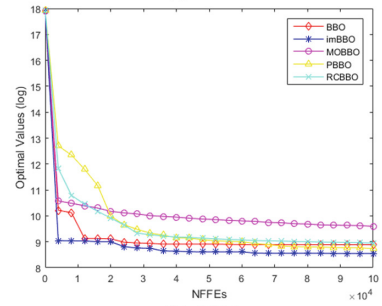
(e) F19



(f) F21



(g) F28



(h) F30

Fig. 1. The convergence curves of imBBO algorithm and four competitors (BBO, MOBBO, PBBO and RCBB0) for F02, F10, F15, F18, F19, F21, F28 and F30 benchmarks. The X-axis shows the number of iteration (NFFEs). The Y-axis presents the algorithm values of each iteration (log-error). (Color figure online)

7 Discussion

In this section, we answer the research question according to the above experimental results. Moreover, we further analyze the experimental results.

7.1 Research Questions

Regarding RQ1, we compare the algorithm performance of our imBBO with the original BBO and three BBO variants. The statistical results show that our algorithm works better than the competitors for 23 out of 30 CEC'2017 benchmarks.

Regarding RQ2, we present part of functions convergence speed of our imBBO and the competitors in Fig. 1 when the evolution running reaches to the *maxFEs*. The experimental results indicate that our algorithm quick converges to a relatively stable state (the blue line in Fig. 1).

7.2 Results Analyzing

Although our imBBO algorithm outperforms the other competitors, the final algorithm results present some differences compared with standard optimization values from CEC'2017. Thus, we analyze the reasons as follows: (1) The functions from CEC'2017 are the latest test benchmarks. Much more complex functions are introduced, especially for hybrid functions and composition functions. Thus, these conditions take the potential probability to influence the algorithm performance. (2) Our global is to verify our imBBO for mitigating part of issues of exploration ability and diversity. We focus on the performance superiority by comparing with other popular BBO variants, especially for the latest CEC'2017 benchmarks set. (3) We insist upon our own view that different exploration methods should be involved into algorithm for different objectives. That is, different characteristics of objectives should be analyzed at the begin of evolution process. Since it is the first time to do the test in CEC'2017, there is no special consideration of objectives in our imBBO algorithm. Furthermore, more components should be developed in future.

8 Conclusion

In this paper, an improved BBO variant called imBBO, is proposed to solve the global optimization problems. Concretely, a hybrid migration operation is designed to further improve the algorithm exploration ability and exploit the population information, which conditionally combines the DE theory with a scalable method to increase the diversity of population in formula 4. Moreover, the mutation operation in our imBBO derives from our previous work because it has been proved its performance successfully.

To evaluate the algorithm performance of imBBO, we conduct the comparison by evaluating our algorithm to the original BBO and three BBO variants based on 30 CEC'2017 benchmarks [3] with different characteristics. Empirical experimental

results demonstrate that our algorithm effectively outperforms the competitors for 23 out of 30 benchmarks.

In future work, the influence of population size, other parameters tuning and the problem dimension will be further studied. Additionally, this research just focus on the unconstrained global numerical optimization problems. Another work will extend our imBBO to address some constrained, real-world optimization problems, such as virtual machine consolidation problems [35, 43].

Acknowledgments. This work is partially supported by the NSF of China under grants No. 61772200, 61702334 and No. 61472139, Shanghai Pujiang Talent Program under grants No. 17PJ1401900, Shanghai Municipal Natural Science Foundation under Grants No. 17ZR1406900 and 17ZR1429700, Educational Research Fund of ECUST under Grant No. ZH1726108, the Collaborative Innovation Foundation of Shanghai Institute of Technology under Grants No. XTCX2016-20, the Opening Project of Key Lab of Information Network Security of Ministry of Public Security Under No. C17604, Key Lab of Information Network Security of Ministry of Public Security Under No. C17604.

References

1. Alatas, B.: ACROA: artificial chemical reaction optimization algorithm for global optimization. *Expert Syst. Appl.* **38**(10), 13170–13180 (2011)
2. Arora, J.S.: Jan A. Snyman, practical mathematical optimization: an introduction to basic optimization theory and classical and new gradient-based algorithms. *Struct. Multi. Optim.* **31**(3), 249–249 (2006)
3. Awad, N., Ali, M., Liang, B., Qu, B., Suganthan, P.: Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective bound constrained real-parameter numerical optimization. Technical report (2016). http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC2017
4. Bhattacharya, A., Chattopadhyay, P.K.: Hybrid differential evolution with biogeography-based optimization algorithm for solution of economic emission load dispatch problems. *Expert Syst. Appl.* **38**(11), 14001–14010 (2011)
5. Černý, V.: Thermodynamical approach to the traveling salesman problem: AN efficient simulation algorithm. *J. Optim. Theory Appl.* **45**(1), 41–51 (1985)
6. Deep, K., Thakur, M.: A new mutation operator for real coded genetic algorithms. *Appl. Math. Comput.* **193**(1), 211–230 (2007)
7. Du, D., Simon, D., Ergezer, M.: Biogeography-based optimization combined with evolutionary strategy and immigration refusal. In: *Proceedings of International Conference on Systems, Man and Cybernetics, San Antonio, USA*, pp. 997–1002 (2009)
8. Ekta, M.K.: Biogeography based optimization: a review. In: *International Conference on Computing for Sustainable Global Development* (2015)
9. Ellabib, I., Calamai, P.H., Basir, O.A.: Exchange strategies for multiple Ant Colony System. *Inf. Sci.* **177**(5), 1248–1264 (2007)
10. Engelbrecht, A.P.: *Computational Intelligence - An Introduction*, 2nd edn. Wiley, Hoboken (2007)
11. Ergezer, M., Simon, D., Du, D.: Oppositional biogeography-based optimization. In: *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, San Antonio, USA*. pp. 1009–1014 (2009)

12. Feng, S.L., Zhu, Q.X., Gong, X.J., Zhong, S.: Hybridizing biogeography-based optimization with differential evolution for motif discovery problem. *Appl. Mech. Mater.* **457–458**(4), 309–312 (2014)
13. Garg, V., Deep, K.: Performance of Laplacian biogeography-based optimization algorithm on CEC 2014 continuous optimization benchmarks and camera calibration problem. *Swarm Evol. Comput.* **27**, 132–144 (2016)
14. Gong, W., Cai, Z., Ling, C.X.: DE/BBO: a hybrid differential evolution with biogeography-based optimization for global numerical optimization. *Soft. Comput.* **15**(4), 645–665 (2010)
15. Gong, W., Cai, Z., Ling, C.X., Li, H.: A real-coded biogeography-based optimization with mutation. *Appl. Math. Comput.* **216**(9), 2749–2758 (2010)
16. Jadon, S.S., Tiwari, R., Sharma, H., Bansal, J.C.: Hybrid artificial bee colony algorithm with differential evolution. *Appl. Soft Comput.* **58**, 11–24 (2017)
17. Kanoongo, S., Jain, P.: Blended biogeography based optimization for different economic load dispatch problem. In: Proceedings of the 25th International Conference on Electrical and Computer Engineering (CCECE), Montreal, QC, Canada, pp. 1–5 (2012)
18. Karaboga, D., Basturk, B.: Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems. In: Melin, P., Castillo, O., Aguilar, L.T., Kacprzyk, J., Pedrycz, W. (eds.) IFSA 2007. LNCS (LNAI), vol. 4529, pp. 789–798. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72950-1_77
19. Li, X., Wang, J., Zhou, J., Yin, M.: A perturb biogeography based optimization with mutation for global numerical optimization. *Appl. Math. Comput.* **218**(2), 598–609 (2011)
20. Li, X., Yin, M.: Multi-operator based biogeography based optimization with mutation for global numerical optimization. *Comput. Math Appl.* **64**(9), 2833–2844 (2012)
21. Liang, J.J., Qin, A.K., Suganthan, P.N., Baskar, S.: Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans. Evol. Comput.* **10**(3), 281–295 (2006)
22. Lohokare, M.R., Panigrahi, B.K., Pattnaik, S.S., Devi, S., Mohapatra, A.: Neighborhood search-driven accelerated biogeography-based optimization for optimal load dispatch. *IEEE Trans. Syst. Man Cybern. Part C* **42**(5), 641–652 (2012)
23. Ma, H.: An analysis of the equilibrium of migration models for biogeography-based optimization. *Inf. Sci.* **180**(18), 3444–3464 (2010)
24. Ma, H., Simon, D.: Biogeography-based optimization with blended migration for constrained optimization problems. In: Proceedings of the International Conference on Genetic and Evolutionary Computation Conference (GECCO), Portland, Oregon, USA, pp. 417–418 (2010)
25. Ma, H., Simon, D.: Analysis of migration models of biogeography-based optimization using markov theory. *Eng. Appl. AI* **24**(6), 1052–1060 (2011)
26. Ma, H., Simon, D.: Blended biogeography-based optimization for constrained optimization. *Eng. Appl. AI* **24**(3), 517–525 (2011)
27. Ma, H., Simon, D., Fei, M., Shu, X., Chen, Z.: Hybrid biogeography-based evolutionary algorithms. *Eng. Appl. AI* **30**, 213–224 (2014)
28. Mirjalili, S., Lewis, A.: The whale optimization algorithm. *Adv. Eng. Softw.* **95**, 51–67 (2016)
29. O'Reilly, U.: Genetic programming II: automatic discovery of reusable programs. *Artif. Life* **1**(4), 439–441 (1994)
30. Pholdee, N., Bureerat, S.: Comparative performance of meta-heuristic algorithms for mass minimisation of trusses with dynamic constraints. *Adv. Eng. Softw.* **75**, 1–13 (2014)
31. Price, K., Storn, R.M., Lampinen, J.A.: Differential Evolution: A Practical Approach to Global Optimization. Springer, Heidelberg (2006). <https://doi.org/10.1007/3-540-31306-0>

32. Rarick, R.A., Simon, D., Villaseca, F.E., Vyakaranam, B.: Biogeography-based optimization and the solution of the power flow problem. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, San Antonio, TX, USA, pp. 1003–1008 (2009)
33. Rashedi, E., Nezamabadi-pour, H., Saryazdi, S.: GSA: a gravitational search algorithm. *Inf. Sci.* **179**(13), 2232–2248 (2009)
34. Shi, K., Yu, H., Fan, G., Luo, F.: iCPBBOCO: a combination evaluation algorithm based on the extensional BBO. In: Proceedings of International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Chengdu, China, pp. 717–723 (2016)
35. Shi, K., Yu, H., Luo, F., Fan, G.: Multi-objective biogeography-based method to optimize virtual machine consolidation. In: Proceedings of 28th International Conference on Software Engineering and Knowledge Engineering (SEKE), Redwood City, San Francisco Bay, USA, pp. 225–230 (2016)
36. Simon, D.: Biogeography-based optimization. *IEEE Trans. Evol. Comput.* **12**(6), 702–713 (2008)
37. Simon, D.: A dynamic system model of biogeography-based optimization. *Appl. Soft Comput.* **11**(8), 5652–5661 (2011)
38. Simon, D., Omran, M.G.H., Clerc, M.: Linearized biogeography-based optimization with re-initialization and local search. *Inf. Sci.* **267**, 140–157 (2014)
39. Singh, U., Singh, D., Kaur, C.: Hybrid differential evolution with biogeography based optimization for Yagi-Uda antenna design. In: Proceedings of the International Conference on Circuit, Power and Computing Technologies, pp. 1163–1167 (2015)
40. Storn, R., Price, K.V.: Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **11**(4), 341–359 (1997)
41. Xiong, G., Shi, D., Duan, X.: Enhancing the performance of biogeography-based optimization using polyphyletic migration operator and orthogonal learning. *Comput. OR* **41**, 125–139 (2014)
42. Yao, X., Liu, Y.: Fast evolution strategies. In: Angeline, P.J., Reynolds, R.G., McDonnell, J.R., Eberhart, R. (eds.) EP 1997. LNCS, vol. 1213, pp. 149–161. Springer, Heidelberg (1997). <https://doi.org/10.1007/BFb0014808>
43. Zheng, Q., et al.: Virtual machine consolidated placement based on multi-objective biogeography-based optimization. *Futur. Gener. Comput. Syst.* **54**, 95–122 (2016)