

Localized biogeography-based optimization

Yu-Jun Zheng · Hai-Feng Ling ·
Xiao-Bei Wu · Jin-Yun Xue

Published online: 21 December 2013
© The Author(s) 2013. This article is published with open access at Springerlink.com

Abstract Biogeography-based optimization (BBO) is a relatively new heuristic method, where a population of habitats (solutions) are continuously evolved and improved mainly by migrating features from high-quality solutions to low-quality ones. In this paper we equip BBO with local topologies, which limit that the migration can only occur within the neighborhood zone of each habitat. We develop three versions of localized BBO algorithms, which use three different local topologies namely the ring topology, the square topology, and the random topology respectively. Our approach is quite easy to implement, but it can effectively improve the search capability and prevent the algorithm from being trapped in local optima. We demonstrate the effectiveness of our approach on a set of well-known benchmark problems. We also introduce the local topologies to a hybrid DE/BBO method, resulting in three localized DE/BBO algorithms, and show that our approach can improve the performance of the state-of-the-art algorithm as well.

Keywords Global optimization · Evolutionary algorithms (EA) · Biogeography-based optimization (BBO) · Local topologies · Differential evolution (DE)

Communicated by Y. Zeng.

Y.-J. Zheng (✉) · X.-B. Wu
College of Computer Science and Technology, Zhejiang University
of Technology, Hangzhou 310023, China
e-mail: yujun.zheng@computer.org

H.-F. Ling
College of Field Engineering, PLA University of Science
and Technology, Nanjing 210007, China

J.-Y. Xue
Jiangxi Provincial Lab of High-Performance Computing,
Jiangxi Normal University, Nanchang 330022, China

1 Introduction

The complexity of real-world optimization problems gives rise to various kinds of evolutionary algorithms (EA) such as genetic algorithms (GA) (Holland 1992), particle swarm optimization (PSO) (Kennedy and Eberhart 1995), differential evolution (DE) (Storn and Price 1997), etc. Drawing inspiration from biological systems, EA typically use a population of candidate individual solutions to stochastically explore the search space. As a result of the great success of such heuristic methods, viewing various natural processes as computations has become more and more essential, desirable, and inevitable (Kari and Rozenberg 2008).

Proposed by Simon (2008), biogeography-based optimization (BBO) is a relatively new EA borrowing ideas from biogeographic evolution for global optimization. Theoretically, it uses a linear time-invariant system with zero input to model migration, emigration, and mutation of creatures in an island (Sinha et al. 2011). In the metaheuristic algorithm, each individual solution is considered as a “habitat” or “island” with a habitat suitability index (HSI), based on which an immigration rate and an emigration rate can be calculated. High HSI solutions tend to share their features with low HSI solutions, and low HSI solutions are likely to accept many new features from high HSI solutions. BBO has proven itself a competitive heuristic to other EA on a wide set of problems (Simon 2008; Du et al. 2009; Song et al. 2010). Moreover, the Markov analysis proved that BBO outperforms GA on simple unimodal, multimodal and deceptive benchmark functions when used with low mutation rates (Simon et al. 2011).

In the original BBO, any two habitats in the population have a chance to communicate with each other, i.e., the algorithm uses a *global topology* which is computationally intensive and easy to cause premature convergence. In this paper

we modify the original BBO by using *local topologies*, where each habitat can only communicate with a subset of the population, i.e., its neighboring habitats. We introduce three local topologies, namely the ring topology, the square topology, and the random topology, to BBO, and demonstrate that our approach can achieve significant performance improvement. We also test the local topologies on a hybrid algorithm, namely DE/BBO (Gong et al. 2010), and show that our approach is also useful to improve the state-of-the-art method. To our best knowledge, the paper is the first attempt to improve BBO by modifying its internal structure with local topologies.

In the remainder of the paper, we first introduce the original BBO in Sect. 2, and then discuss related work in Sect. 3. In Sect. 4 we propose our approach for improving BBO, the effectiveness of which is demonstrated by the experiments in Sect. 5. Finally we conclude in Sect. 6.

2 Biogeography-based optimization

Biogeography is the science of the geographical distribution of biological organisms over space and time. MacArthur and Wilson (1967) established the mathematical models of island biogeography, showing that the species richness of an island can be predicted in terms of such factors as habitat area, immigration rate, and extinction rate. Inspired by this, Simon (2008) developed BBO, where a solution is analogous to a habitat, the solution components are analogous to a set of suitability index variables (SIV), and the solution fitness is analogous to the species richness or HSI of the habitat. Central to the algorithm is the equilibrium theory of biogeography, which indicates that high HSI habitats have a high species emigration rate and low HSI habitats have a high species immigration rate. For example, in a linear model of species richness (Fig. 1), a habitat H_i 's immigration rate λ_i and emigration rate μ_i are calculated based on its fitness f_i as follows:

$$\lambda_i = I \left(1 - \frac{f_i}{f_{\max}} \right) \quad (1)$$

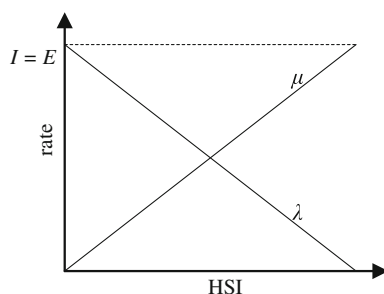


Fig. 1 A linear model of emigration and immigration rates

$$\mu_i = E \left(\frac{f_i}{f_{\max}} \right) \quad (2)$$

where f_{\max} is the maximum fitness value among the population, and I and E are the maximum possible immigration rate and emigration rate respectively. However, there are other non-linear mathematical models of biogeography that can be used for calculating the migration rates (Simon 2008).

Migration modifies habitats by mixing features within the population. BBO also use a mutation operator to change SIV within a habitat itself, and thus probably increasing diversity of the population. For each habitat H_i , a species count probability P_i computed from λ_i and μ_i indicates the likelihood that the habitat was expected a priori to exist as a solution for the problem. In this context, very high HSI habitats and very low HSI habitats are both improbable, and medium HSI habitats are relatively probable. The habitat's mutation rate is inversely proportional to its probability:

$$\pi_i = \pi_{\max} \left(1 - \frac{P_i}{P_{\max}} \right) \quad (3)$$

where π_{\max} is a control parameter and P_{\max} is the maximum habitat probability in the population.

Algorithm 1 shows the general framework of BBO for a D -dimensional numerical optimization problem (where l_d and u_d are the lower and upper bounds of the d th dimension respectively, and *rand* is a function generating a random value uniformly distributed in $[0,1]$).

Algorithm 1 The original BBO algorithm.

```

1 Randomly initialize a population of  $n$  solutions (habitats);
2 while stop criterion is not satisfied do
3   for  $i = 1$  to  $n$  do
4     Calculate  $\lambda_i$ ,  $\mu_i$ , and  $\pi_i$  according to  $f_i$ ;
5   for  $i = 1$  to  $n$  do
6     for  $d = 1$  to  $D$  do
7       if  $\text{rand}() < \lambda_i$  then //migration
8         Select a habitat  $H_j$  with probability  $\propto \mu_j$ ;
9          $H_{i,d} \leftarrow H_{j,d}$ ;
10    for  $i=1$  to  $n$  do
11      for  $d = 1$  to  $D$  do
12        if  $\text{rand}() < \pi_i$  then //mutation
13           $H_{i,d} \leftarrow l_d + \text{rand}() \times (u_d - l_d)$ ;
14    Evaluate the fitness values of the habitats;
15    Update  $f_{\max}$ ,  $P_{\max}$ , and the best known solution;
16  return the best known solution.
```

Typically, in Line 8 we can use a roulette wheel method for selection, the time complexity of which is $O(n)$. It is not difficult to see that the complexity of each iteration of the algorithm is $O(n^2D + nf)$, where f denotes the time complexity of the fitness function.

3 Related work

Since the proposal of the original BBO, several work has been devoted to improve its performance of by incorporating features of other heuristics. [Du et al. \(2009\)](#) combined BBO with the elitism mechanism of evolutionary strategy (ES) ([Beyer and Schwefel 2002](#)), that is, at each generation the best n solutions from the n parents and n children as the population are chosen for the next generation. Based on opposite numbers theory ([Tizhoosh 2005](#); [Rahnamayan et al. 2008](#); [Wang et al. 2011](#); [Ergezer et al. 2009](#)) employed opposition-based learning to create the oppositional BBO, which uses a basic population together with an opposite population in problem solving. Each basic solution has an opposite reflection in the opposite population, which is probably closer to the expected solution than the basic one. Consequently, the required search space can be reduced and the convergence speed can increase.

The migration operator makes BBO be good at exploiting the information of the population. [Gong et al. \(2010\)](#) proposed a hybrid DE/BBO algorithm, which combines the exploration of DE with the exploitation of BBO effectively. The core idea is to hybridize DE's mutation operator with BBO's migration operator, such that good solutions would be less destroyed, while poor solutions can accept a lot of new features from good solutions. In some detail, at each iteration, each d th component of each i th solution is updated using the following procedure (where CR is the crossover rate, r_1 , r_2 and r_3 are mutually distinct random indices of the population, and F is a differential weight):

```

1  if rand() <  $\lambda_i$  then
2    if rand() <  $CR \vee d = d_{rand}$  then
3       $H_{i,d} \leftarrow H_{r_1,d} + F(H_{r_2,d} - H_{r_3,d})$ ;
4    else
5      Select a habitat  $H_j$  with probability  $\propto \mu_j$ ;
6       $H_{i,d} \leftarrow H_{j,d}$ ;

```

[Li and Yin \(2012\)](#) proposed two extensions of BBO, the first using a new migration operation based on multi-parent crossover for balancing the exploration and exploitation capability of the algorithm, and the second integrating a Gaussian mutation operator to enhance its exploration ability.

Observing that the performance of BBO is sensitive to the migration model, [Ma et al. \(2012a\)](#) proposed an approach that equips BBO with an ensemble of migration models. They realized such an algorithm using three parallel populations, each of which implements a different migration model. Experimental results show that the new method generally outperforms various versions of BBO with a single migration model.

The original BBO is for unconstrained optimization. [Ma and Simon \(2011\)](#) developed a blended BBO for constrained optimization, which uses a blended migration operator motivated by blended crossover in GA ([Mühlenbein and Schlierkamp-Voosen 1993](#)), and determines whether a modified solution can enter the population of the next generation by comparing it with its parent. In their approach, an infeasible solution is always considered worse than a feasible one, and among two infeasible solutions the one with a smaller constraint violation is considered better.

Some recent studies ([Mo and Xu 2011](#); [Ma et al. 2012b](#); [Costa e Silva et al. 2012](#)) have been carried out on the application of BBO to multiobjective optimization problems. Those methods typically evaluate solutions based on the concept of Pareto optimality, use an external archive for maintaining non-dominated solutions, and employ specific strategy to improve diversity of the archive. Experiments show that they can provide comparable performance to some well-known multiobjective EA such as NSGA-II ([Deb et al. 2002](#)).

[Boussaïd et al. \(2011\)](#) developed another approach combining BBO and DE, which evolves the population by alternately applying BBO and DE iterations. [Boussaïd et al. \(2012\)](#) extended the approach for constrained optimization, which replaces the original mutation operator of BBO with the DE mutation operator, and includes a penalty function to the objective function to handle problem constraints.

Recently [Lohokare et al. \(2013\)](#) proposed a new hybrid BBO method, namely aBBOmDE, which uses a modified mutation operator to enhance exploration, introduces a clear duplicate operator to suppress the generation of similar solutions, and embeds a DE-based neighborhood search operator to further enhance exploitation. The improved memetic algorithm exhibits significantly better performance than BBO and DE and comparable performance with DE/BBO.

All of the above work focuses on hybridization and inherits the global topology of the original BBO, i.e., any two habitats in the population have a chance to communicate with each other. Up to now, few researches have been carried out on the improvement of internal structure of BBO. Here we refer to PSO, the most popular swarm-based EA which also uses a global best model in its original version, i.e., at each iteration each particle is updated by learning from its own memory and the very best particle in the entire population. However, many later researches have revealed that the global best model shows inferior performance on many problems and prefer to use a local best model ([Kennedy and Mendes 2006](#)), i.e., make every particle learn from a local best particle among its immediate neighbors rather than the entire population. Drawing inspiration from this, we conduct this research that introduces local topologies to BBO.

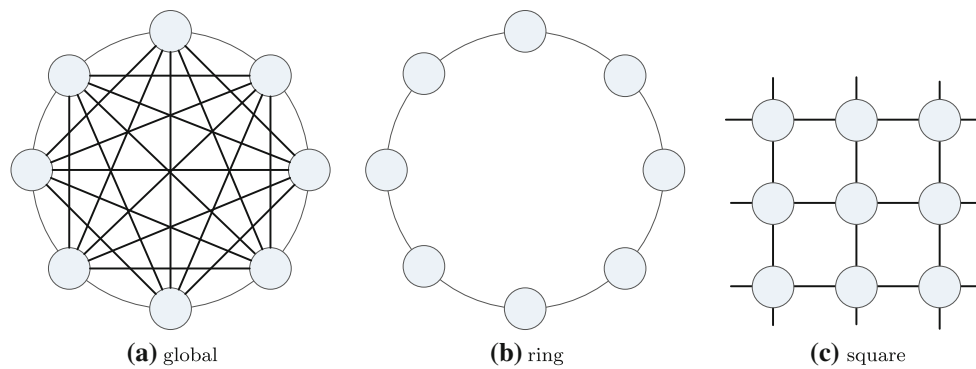


Fig. 2 Illustration of the global, the local ring and the local square topologies

4 Localized biogeography-based optimization

The original BBO uses a global topology where any two habitats can communicate with each other: if a given habitat is chosen for immigration, any other habitat has a chance to be an emigrating habitat, as shown in Fig. 2a. However, such a migration mechanism is computationally expensive. More importantly, if one or several habitats near local optima are much better than others in the population, then all other solutions are very likely to accept many features from them, and thus the algorithm is easy to be trapped into the local optima.

A way to overcome this problem is to use a local topology, where each individual can only communicate with its (immediate) neighbors rather than any others in the population. Under this scheme, as some individuals in one part of the population interact with each other to search one region of the solution space, another part of the population could explore another. However, the different parts of the population are not isolated. Instead they are interconnected by some intermediate individuals, and the flow of information can be moderately pass through the neighborhood. In consequence, the algorithm can achieve a better balance between global search (exploration) and local search (exploitation).

4.1 Using the ring and square topologies

The first local topology we introduce to BBO is the *ring* topology, where each habitat is directly connected to two other habitats, as shown in Fig. 2b. The second is the *square* topology, where the habitats are arranged in a grid and each habitat has four immediate neighbors, as shown in Fig. 2c.

Suppose the population of n habitats are stored in an array or a list, the i th of which is denoted by H_i . When using the ring topology, each H_i has two neighbors $H_{(i-1)\%n}$ and

$H_{(i+1)\%n}$, where $\%$ denotes the modulo operation. In this case, Line 8 of Algorithm 1 can be refined to the following procedure:

```

1 let  $i1 = (i - 1)\%n, i2 = (i + 1)\%n$ ;
2 if  $rand() < \mu_{i1}/(\mu_{i1} + \mu_{i2})$  then Select  $H_{i1}$  as  $H_j$ ;
3 else Select  $H_{i2}$  as  $H_j$ ;

```

For the square topology where the width of the grid is w , the indices of the neighbors of habitat H_i are respectively $(i - 1)\%n$, $(i + 1)\%n$, $(i - w)\%n$ and $(i + w)\%n$. In this case, Line 8 of Algorithm 1 can be refined to the following procedure:

```

1 let  $i1 = (i - 1)\%n, i2 = (i + 1)\%n$ ,
    $i3 = (i - w)\%n, i4 = (i + w)\%n$ ;
2 let  $r = rand()$ ;
3 if  $r < \mu_{i1}$  then select  $H_{i1}$  as  $H_j$ ;
4 else if  $r < (\mu_{i1} + \mu_{i2})$  then select  $H_{i2}$  as  $H_j$ ;
5 else if  $r < (\mu_{i1} + \mu_{i2} + \mu_{i3})$  then select  $H_{i3}$  as  $H_j$ ;
6 else select  $H_{i4}$  as  $H_j$ ;

```

For both the two topologies, the complexity of each iteration of the algorithm decreases to $O(nD + nf)$. In case that the fitness function of the problem is not very difficult, the BBO with the ring topology can save much computational cost.

4.2 Using the random topology

The neighborhood size is fixed to 2 in the ring topology and 4 in the square topology. A more general approach is to set the neighborhood size to K ($0 < K < n$), which is a parameter that can be adjusted according to the properties of the problem at hand. But how to set the topology for an arbitrary K ? A simple way is to randomly select K neighbors for each habitat, the result of which is called a *random* topology. In the algorithm implementation, the neighborhood structure

Table 1 The experimental results of BBO and localized BBO algorithms on test problems

ID	NFE	Fitness value				Success rate (%)			
		BBO	RingBBO	SquareBBO	RandBBO	BBO	RingBBO	SquareBBO	RandBBO
f_1	150,000	2.43E+00 (8.64E-01)	2.90E-01* (1.01E-01)	3.09E-01* (1.19E-01)	3.26E-01* (1.38E-01)	0	0	0	0
f_2	150,000	5.87E-01 (8.60E-02)	1.96E-01* (3.72E-02)	1.95E-01* (4.18E-02)	2.09E-01* (3.73E-02)	0	0	0	0
f_3	500,000	4.23E+00 (1.73E+00)	4.88E-01* (2.09E-01)	4.09E-01* (1.58E-01)	3.98E-01* (1.58E-01)	0	0	0	0
f_4	500,000	1.62E+00 (2.80E-01)	7.61E-01* (1.00E-01)	6.55E-01* (1.44E-01)	7.00E-01* (1.25E-01)	0	0	0	0
f_5	500,000	1.15E+02 (3.66E+01)	8.60E+01* (2.84E+01)	6.63E+01* (3.73E+01)	7.83E+01* (3.25E+01)	0	0	0	0
f_6	150,000	2.20E+00 (1.61E+00)	0.00E+00* (0.00E+00)	3.33E-02* (1.83E-01)	3.33E-02* (1.83E-01)	13.3	100	96.7	96.7
f_7	300,000	3.31E-03 (3.01E-03)	6.16E-03° (3.65E-03)	5.93E-03° (3.13E-03)	5.90E-03° (2.44E-03)	96.7	83.3	90	93.3
f_8	300,000	1.90E+00 (8.02E-01)	2.44E-01* (1.17E-01)	2.35E-01* (9.27E-02)	2.04E-01* (5.92E-02)	0	0	0	0
f_9	300,000	3.51E-01 (1.40E-01)	3.89E-02* (1.73E-02)	3.94E-02* (1.18E-02)	3.69E-02* (1.06E-02)	0	0	0	0
f_{10}	150,000	7.79E-01 (2.03E-01)	1.76E-01* (4.02E-02)	1.77E-01* (4.85E-02)	1.63E-01* (4.21E-02)	0	0	0	0
f_{11}	200,000	8.92E-01 (1.01E-01)	2.36E-01* (7.00E-02)	2.35E-01* (8.34E-02)	2.82E-01* (8.34E-02)	0	0	0	0
f_{12}	150,000	7.35E-02 (2.87E-02)	7.32E-03* (3.68E-03)	6.29E-03* (4.23E-03)	7.71E-03* (7.19E-03)	0	0	0	0
f_{13}	150,000	3.48E-01 (1.23E-01)	3.81E-02* (2.81E-02)	4.19E-02* (2.87E-02)	3.90E-02* (2.71E-02)	0	0	0	0
f_{14}	50,000	3.80E-03 (1.01E-02)	3.38E-06* (1.41E-05)	3.44E-06* (1.87E-05)	1.01E-06* (4.04E-06)	13.3	50	56.7	40
f_{15}	50,000	6.59E-03 (7.70E-03)	2.42E-03* (2.12E-03)	3.52E-03* (5.23E-03)	2.99E-03* (4.25E-03)	0	0	0	0
f_{16}	50,000	2.61E-03 (3.85E-03)	2.40E-04* (3.71E-04)	1.55E-04* (1.51E-04)	2.40E-04* (4.87E-04)	0	0	0	0
f_{17}	50,000	4.71E-03 (1.18E-02)	5.78E-04* (9.65E-04)	3.90E-04* (7.37E-04)	6.61E-04* (1.05E-03)	0	0	0	0
f_{18}	50,000	9.30E-01 (4.93E+00)	3.44E-03 (7.08E-03)	3.50E-03 (6.32E-03)	2.54E-03 (4.54E-03)	0	0	0	0
f_{19}	50,000	2.59E-04 (2.37E-04)	2.19E-05* (3.62E-05)	1.41E-05* (1.66E-05)	2.49E-05* (1.97E-05)	0	0	0	0
f_{20}	50,000	2.83E-02 (5.12E-02)	3.99E-03* (2.17E-02)	7.95E-03* (3.02E-02)	3.36E-05* (3.63E-05)	0	0	0	0
f_{21}	20,000	5.08E+00 (3.40E+00)	2.70E+00* (3.39E+00)	2.95E+00* (3.63E+00)	3.60E+00* (3.67E+00)	0	0	0	0
f_{22}	20,000	4.25E+00 (3.44E+00)	3.14E+00 (3.41E+00)	3.91E+00 (3.48E+00)	2.32E+00* (3.34E+00)	0	0	0	0

Table 1 continued

ID	NFE	Fitness value				Success rate (%)			
		BBO	RingBBO	SquareBBO	RandBBO	BBO	RingBBO	SquareBBO	RandBBO
f_{23}	20,000	5.01E+00 (3.37E+00)	3.64E+00* (3.48E+00)	2.55E+00* (3.36E+00)	1.98E+00* (3.31E+00)	0	0	0	0
Win			20 vs 1	20 vs 1	21 vs 1				

In column 3–5, * means that the localized BBO has statistically significant improvement over BBO and ° vice versa (at 95 % confidence level). The last row compares the number of problems on which the algorithm outperform BBO and vice versa. In table column 2–5, the upper part of each cell corresponds to the mean best and the lower part corresponds to the standard deviation

can be saved in an $n \times n$ matrix *Link*: if two habitats H_i and H_j are directly connected then $Link(i, j) = 1$, otherwise $Link(i, j) = 0$.

However, such an approach has the following limitations:

- K has to be an integer, which limits the adjustability of the parameter.
- All habitats in the population have the same number of neighbors. But for an EA modeling a social network, it can be better that some individuals have more informants while others have less (Kennedy and Mendes 2006).

A more effective way is to make that each habitat has *probably* K neighbors. This gives a Bernoulli distribution where any value between $0 \sim (n - 1)$ is possible, such that all possible topologies may appear. Precisely, the probability of any two habitats being connected is $K/(n - 1)$, and thus we can use the following procedure for setting the topology with an expected neighborhood size of K :

```

1 Initialize an  $n \times n$  matrix Link;
2 let  $p = K/(n - 1)$ ;
3 for  $i = 1$  to  $n$  do
4   if  $rand() < p$  then  $Link(i, j) \leftarrow 1$ ;
5   else  $Link(i, j) \leftarrow 0$ ;
```

The random topology provides enough diversification for effectively dealing with multimodal problems, but its neighborhood structure should be dynamically changed according to the state of convergence. There are different strategies, such as resetting the neighborhood structure at each iteration, after a certain number of iterations, or after one or a certain number of non-improvement iterations. Empirically, we suggest setting K to a value between 2 and 4 and resetting the neighborhood structure when the best solution has not been improved after a whole iteration.

Algorithm 2 presents the framework of such a localized BBO with the random topology.

Algorithm 2 The localized BBO with the random topology.

```

1 Randomly initialize a population of  $n$  solutions;
2 Initialize the neighborhood structure;
3 while stop criterion is not satisfied do
4   for  $i = 1$  to  $n$  do
5     Calculate  $\lambda_i$ ,  $\mu_i$ , and  $\pi_i$  according to  $f_i$ ;
6   for  $i = 1$  to  $n$  do
7     for  $d = 1$  to  $D$  do
8       if  $rand() < \lambda_i$  then
9         Select a habitat  $H_j$  satisfying  $Link(i, j) = 1$ 
           with probability  $\propto \mu_j$ ;
10         $H_{i,d} \leftarrow H_{j,d}$ ;
11   for  $i=1$  to  $n$  do
12     for  $d = 1$  to  $D$  do
13       if  $rand() < \pi_i$  then
14          $H_{i,d} \leftarrow l_d + rand() \times (u_d - l_d)$ ;
15   Evaluate the fitness values of the habitats;
16   Update  $f_{max}$ ,  $P_{max}$ , and the best known solution;
17   if no new best known solution is found then
18     Reset the neighborhood structure;
19 return the best solution found so far.
```

5 Computational experiments

5.1 Benchmark functions and experimental settings

We test the localized BBO algorithms on a set of well-known benchmark functions from Yao et al. (1999), denoted as $f_1 \sim f_{23}$, which span a diverse set of features of numerical optimization problems. $f_1 \sim f_{13}$ are high dimensional and scalable problems, and $f_{14} \sim f_{23}$ are low dimensional problems with a few local minima. For the high dimensional f_8 and all the low dimensional functions that have non-zero optimal values, in the experiments we simply add some constants to the objectives such that their best fitness values become 0.

The experiments are conducted on a computer of Intel Core i5-2520M processor and 4 GB memory. For a fair comparison, all the algorithms use the same population size $n = 50$ and the same maximum number of function evaluations (NFE) preset for each test problem. For the local BBO with the random topology we set $K = 3$. Every algorithm has been run on each problem for 60 times with different

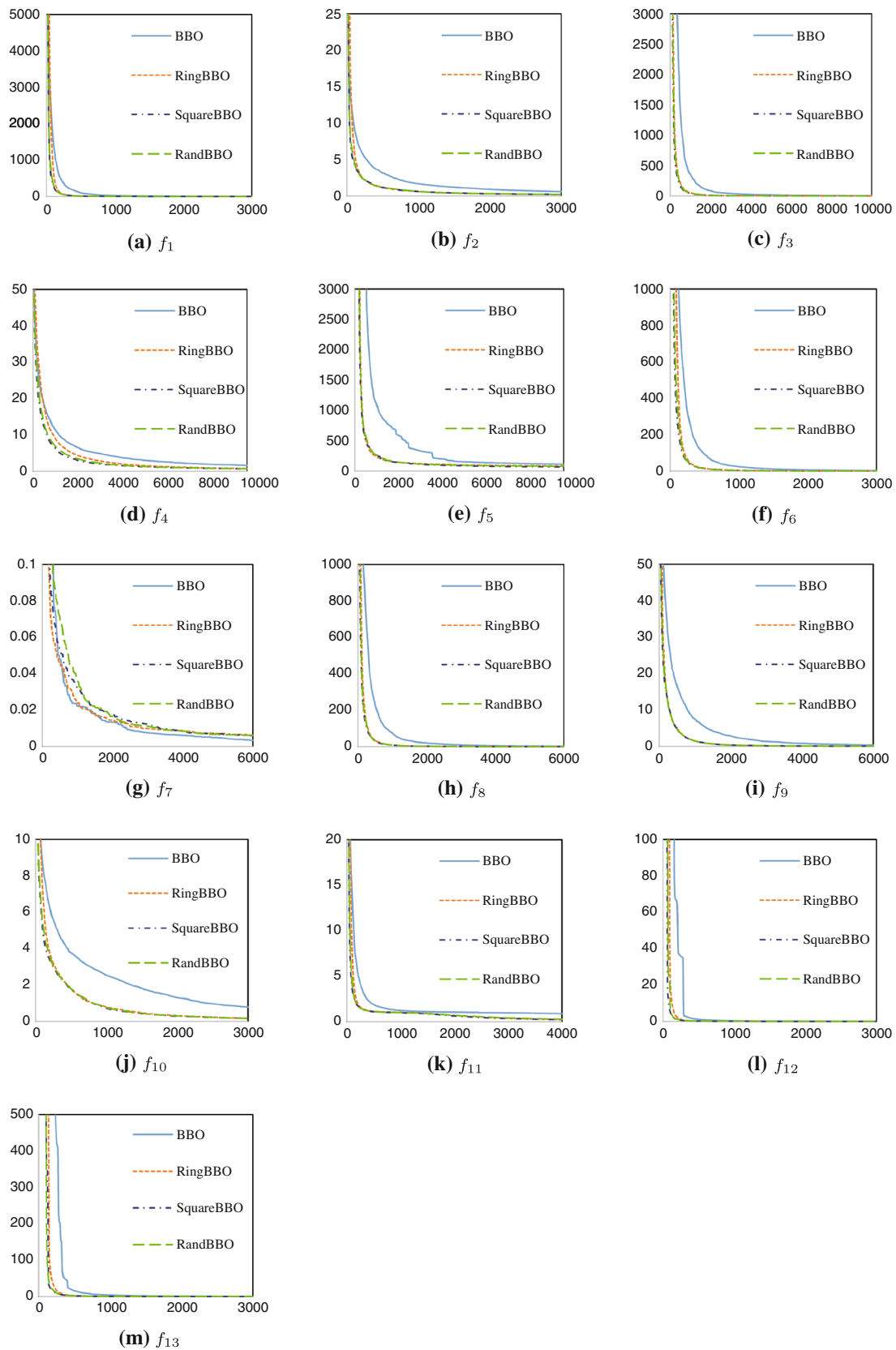


Fig. 3 Converge curves of the BBO and localized BBO algorithms on test functions $f_1 \sim f_{13}$. The horizontal axis denotes the number of algorithm iterations and the vertical axis denotes the mean best fitness

Table 2 The experimental results of DE, DE/BBO and localized DE/BBO algorithms on test problems

ID	NFE	Fitness value					Success rate (%)				
		DE	DE/BBO	RingDB	SquareDB	RandDB	DE	DE/BBO	RingDB	SquareDB	RandDB
f_1	150,000	2.69E-51 (7.28E-51)	5.48E-57 ⁺ (6.70E-57)	2.96E-83 ^{*+} (4.40E-83)	1.92E-71 ^{*+} (1.49E-70)	5.92E-70 ^{*+} (4.01E-69)	100	100	100	100	100
f_2	150,000	8.38E-28 (1.20E-27)	5.27E-33 ⁺ (2.63E-33)	3.04E-50 ^{*+} (2.56E-50)	4.79E-49 ^{*+} (2.63E-48)	6.59E-47 ^{*+} (3.59E-46)	100	100	100	100	100
f_3	500,000	1.09E-178 (0.00E+00)	2.93E-198 (0.00E+00)	1.79E-285 (0.00E+00)	4.41E-225 (0.00E+00)	5.81E-216 (0.00E+00)	100	100	100	100	100
f_4	500,000	9.89E+00 (4.33E+00)	1.84E-09 ⁺ (1.01E-08)	2.32E-05 [°] (7.84E-05)	3.47E-05 (1.36E-04)	6.63E-05 [°] (2.40E-04)	0	96.7	80	56.7	63.3
f_5	500,000	1.46E+01 (4.13E+00)	2.57E+02 ⁻ (6.17E-01)	2.60E+02 ⁻ (1.37E+01)	2.63E+02 ^{°-} (6.84E-01)	2.62E+02 ^{°-} (1.41E+01)	0	0	0	0	0
f_6	50,000	3.33E-02 (1.83E-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)	96.7	100	100	100	100
f_7	300,000	7.22E-03 (5.49E-03)	5.66E-03 (4.03E-03)	1.77E-03 ^{*+} (1.42E-03)	1.81E-03 ^{*+} (1.11E-03)	1.60E-03 ^{*+} (9.62E-04)	70	83.3	100	100	100
f_8	300,000	4.99E+02 (3.0E+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)	10	100	100	100	100
f_9	300,000	1.32E+01 (3.62E+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	100	100	100	100
f_{10}	150,000	7.19E-15 (1.95E-15)	5.30E-15 ⁺ (1.74E-15)	4.47E-15 ^{*+} (1.23E-15)	4.23E-15 ^{*+} (9.01E-16)	4.59E-15 ^{*+} (1.35E-15)	100	100	100	100	100
f_{11}	150,000	2.96E-03 (5.31E-03)	0.00E+00 ⁺ (0.00E+00)	0.00E+00 ⁺ (0.00E+00)	0.00E+00 ⁺ (0.00E+00)	0.00E+00 ⁺ (0.00E+00)	100	100	100	100	100
f_{12}	150,000	2.00E-32 (1.02E-32)	1.57E-32 (0.00E+00)	1.57E-32 (0.00E+00)	1.57E-32 (0.00E+00)	1.57E-32 (0.00E+00)	100	100	100	100	100
f_{13}	150,000	3.30E-18 (1.81E-17)	1.35E-32 (0.00E+00)	1.35E-32 (0.00E+00)	1.35E-32 (0.00E+00)	1.35E-32 (0.00E+00)	100	100	100	100	100
f_{14}	10,000	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)	100	100	100	100	100
f_{15}	40,000	5.53E-19 (9.16E-20)	7.39E-05 ⁻ (8.76E-05)	1.35E-04 ⁻ (1.31E-04)	1.27E-04 ⁻ (1.26E-04)	1.33E-04 ⁻ (1.87E-04)	100	20	3.33	13.3	13.3
f_{16}	10,000	1.22E-13 (0.00E+00)	1.22E-13 (0.00E+00)	1.22E-13 (0.00E+00)	1.22E-13 (0.00E+00)	1.22E-13 (0.00E+00)	100	100	100	100	100
f_{17}	10,000	1.74E-16 (0.00E+00)	1.74E-16 (0.00E+00)	1.74E-16 (0.00E+00)	1.74E-16 (0.00E+00)	1.74E-16 (0.00E+00)	100	100	100	100	100
f_{18}	10,000	9.59E-14 (0.00E+00)	9.59E-14 (0.00E+00)	9.59E-14 (0.00E+00)	9.59E-14 (0.00E+00)	9.59E-14 (0.00E+00)	100	100	100	100	100
f_{19}	10,000	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)	100	100	100	100	100
f_{20}	10,000	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)	100	100	100	100	100
f_{21}	10,000	1.68E-01 (9.22E-01)	2.49E-01 (1.36E+00)	4.59E-01 (1.75E+00)	2.56E-01 (1.36E+00)	1.20E-01 (5.17E-01)	96.7	96.7	90	93.3	93.3
f_{22}	10,000	1.81E-10 (8.95E-16)	1.81E-10 (6.73E-16)	1.81E-10 (7.99E-16)	1.81E-10 (6.73E-16)	1.81E-10 (8.95E-16)	100	100	100	100	100

Table 2 continued

ID	NFE	Fitness value					Success rate (%)				
		DE	DE/BBO	RingDB	SquareDB	RandDB	DE	DE/BBO	RingDB	SquareDB	RandDB
f_{23}	10,000	2.00E-15 (5.42E-16)	2.00E-15 (5.42E-16)	2.13E-15 (7.23E-16)	1.78E-15 (0.00E+00)	2.07E-15 (6.73E-16)	100	100	100	100	100
Win			7 vs 2	4 vs 1 7 vs 2	4 vs 1 7 vs 2	4 vs 2 7 vs 2					

In table column 2–6, the upper part of each cell corresponds to the mean best and the lower part corresponds to the standard deviation. In column 4–6, • means that the localized BBO has statistically significant improvement over DE/BBO and ° vice versa; In column 3–6, + means that the algorithm has statistically significant improvement over DE and - vice versa (at 95 % confidence level). The last row compares the number of problems on which the algorithm outperform DE/BBO and vice versa, and the last row compares the number of problems on which the algorithm outperform DE and vice versa

random seeds, and the resulting mean best fitness values and the success rate (with respect to the required accuracies) are averaged over 60 runs. The required accuracies are set to 10^{-8} for 22 test functions except 10^{-2} for f_7 (Suganthan et al. 2005).

5.2 Comparative experiments for BBO with global and local topologies

We first compare our three versions of localized BBO, denoted by RingBBO, SquareBBO and RandBBO respectively, with the basic BBO (for which we use program code from <http://academic.csuohio.edu/simond/bbo>). For all the four algorithms, we set $I = E = 1$. We also set π_{max} to 0.01 for BBO, but 0.02 for localized BBO to increase the chance for mutation since the local topologies decrease the impact of migration.

Table 1 presents the mean bests and success rates of the four BBO on the test functions. As we can see, the mean best fitness values of the three localized BBO algorithms are always better than BBO on all the functions except f_7 . In particular, on high dimensional functions except f_7 , the mean bests of the localized BBO are typically about 1–20 % of that of the BBO. The localized BBO also have obvious improvement of the success rates on functions f_6 and f_{14} . The BBO achieves better mean best fitness only on f_7 , but in terms of success rate there is no significant difference between the BBO and localized BBO.

We have performed paired t -tests on the mean values of the BBO and localized BBO. As the results show, RingBBO and SquareBBO have significant improvement over the BBO on 20 functions, and RandBBO has significant improvement on 21 functions. There are no significant differences between the four algorithms on function f_{17} , which is low dimensional and is one of the simplest functions among the test problems. BBO is significantly better only on f_7 .

Moreover, Fig. 3a–m respectively present the convergence curves of the four algorithms on high dimensional functions

$f_1 - f_{13}$. From the curves we can see that, on all the functions except f_7 , the three localized BBO have obvious advantages in convergence speed. There are no obvious differences between the convergence speeds of the three localized BBO.

Quartic function f_7 is the only function on which the BBO has performance advantage over the localized BBO, but the advantage is not very significant. Note that f_7 is a noisy function with one minima, where the noise is a random value uniformly distributed in $[0,1]$. For such a relatively simple function, the BBO with global communication topology is more capable of tolerating noises. However, since the RingBBO and SquareBBO have low time complexity and it is not expensive to evaluate f_7 , when given the same computational time, we observe that the RingBBO and SquareBBO still achieve better result than the BBO. For most functions without noises and multimodal functions, the local topologies have much more advantages.

On the other hand, there are no statistically significant differences between the three versions of localized BBO. Roughly speaking, the performance of the RandBBO is better than the RingBBO and SquareBBO, but RandBBO consumes a bit more computational cost. On most high dimensional problems, SquareBBO is slightly better than RingBBO in terms of mean best fitness, but RingBBO is better in terms of standard deviation. In other words, all of the three localized BBO provide performance improvement over the BBO; the improvement of RandBBO and SquareBBO is more obvious, and RingBBO is expected to be more stable.

5.3 Comparative experiments for DE/BBO with global and local topologies

As described in the related work, the DE/BBO (Gong et al. 2010) balances the exploration of DE with the exploitation of BBO by hybridizing DE mutation with BBO migration. In order to show that the local topologies can also improve the state-of-the-art hybrid BBO, we develop three new versions of DE/BBO by introducing the ring, square, and random

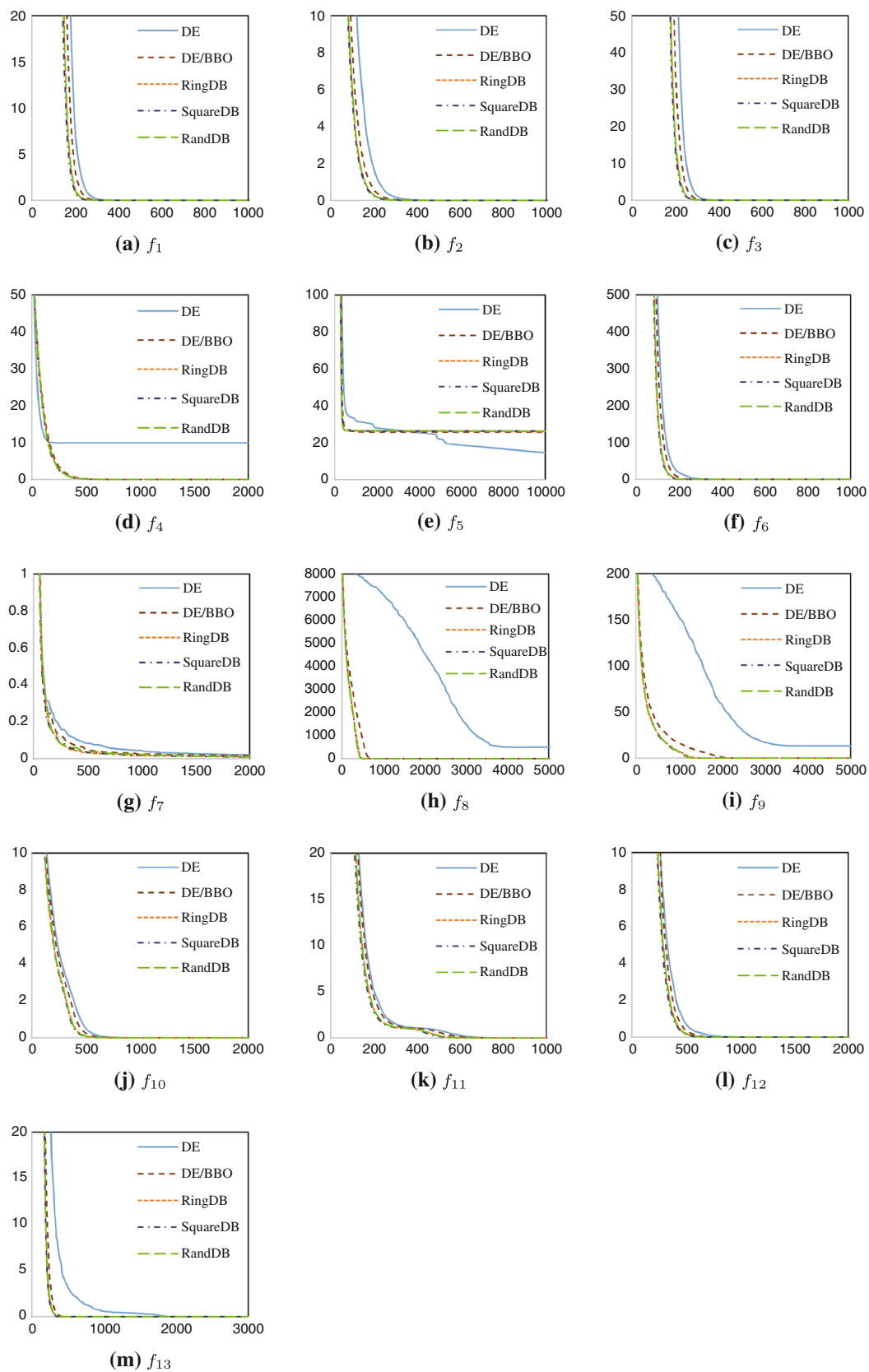


Fig. 4 Converge curves of the BBO and localized BBO algorithms on test functions $f_1 \sim f_{13}$. The *horizontal axis* denotes the number of algorithm iterations and the *vertical axis* denotes the mean best fitness

topologies, denoted by RingDB, SquareDB, and RandDB respectively. For parameters concerning with DE scheme, we set $F = 0.5$ and $CR = 0.9$.

We respectively run the DE, DE/BBO, RingDB, SquareDB, and RandDB for 60 times on the set of 23 functions, the results of which are shown in Table 2. As we can see, on f_1, f_2, f_3, f_7 and f_{10} , the mean bests obtained by the three local DE/BBO algorithms are better than the original DE/BBO with the default global topology; on the contrary, performance of DE/BBO is slightly better only on two unimodal functions f_4 and f_5 , but the advantages are not very obvious. Since DE/BBO scheme is very efficient for multimodal functions, all of the four DE/BBO algorithms reach the true (or very near true) optima on functions such as f_8, f_9, f_{11}, f_{12} and f_{13} .

Statistical tests show that, the three localized DE/BBO have significant performance improvement over DE/BBO on four high dimensional functions, and DE/BBO has significant improvement over RingDB and RandDB on f_4 and over SquareDB and RandDB on f_5 . In comparison with DE, the four versions of DE/BBO have significant performance improvement on five high dimensional functions, and DE has significant improvement over the DE/BBO algorithms only on high dimensional f_5 and low dimensional f_{15} . Moreover, on f_7 DE and the original DE/BBO have no significant difference, but the three localized DE/BBO have significant performance improvement over DE.

The convergence curves presented in Fig. 4a–m also show that, the four DE/BBO algorithms converge faster than DE on 12 high dimensional functions except f_5 ; the DE/BBO algorithms have similar convergence speed on f_4 and f_5 , but the three localized DE/BBO converge faster than DE/BBO on the remaining 11 functions.

Similarly, there are no statistically significant differences between the three versions of localized DE/BBO. However, the performance of RingDB is slightly better than SquareDB and RandDB on some high dimensional problems. This is mainly because that, the DE search method enhances the global information sharing in the population and provides good exploration capability, and a large neighborhood size (such as in SquareDB) or random topology is not very important for hybrid DE/BBO. A local topology can effectively enhance the BBO's exploitation capability, and the combination of DE and BBO with the ring topology can provide a promising search capability.

6 Conclusion

Biogeography-based optimization is a relatively new bio-inspired EA which has proven its quality and versatility on a wide range of optimization problems. Although many

researches have been devoted to improve the performance of BBO, most of them focus on combining with operators of some other heuristic methods and thus increase the difficulty of implementation. In this paper we propose a new approach to improve BBO by simply replacing the global topology of BBO with a local topology. Under this scheme, as some individuals in one part of the population interact with each other to search one region of the solution space, another part of the population could explore another, and the flow of information can be moderately pass through the neighborhood. Thus our method can achieve a much better balance between exploration and exploitation. The main contributions of the paper can be summarized as follows:

1. To our best knowledge, it is the first attempt to improve BBO with local topologies, which not only effectively improves the algorithm's search capability and avoid premature convergence, but also is much easier to implement than other hybrid approaches.
2. The localized approach has also been successfully used to greatly improve a hybrid DE/BBO method. Also, we believe that our approach can be utilized to further improve other enhanced BBO methods, such as Li and Yin (2012); Ma et al. (2012a); Lohokare et al. (2013), etc.
3. The localized approach can be very useful in designing new BBO-based heuristics, including those for constrained, multiobjective, and/or combinatorial optimization.

Currently, we are testing the hybridization of our localized BBO with ES and PSO. Another ongoing work is the parallelization of our BBO algorithms, which is much easier to implement on local topologies than on a global one (Zhu 2010).

In this paper we study three local topologies: ring, square, and random. There are many other interesting neighborhood topologies that are worth to study (Mendes et al. 2004). But we still have an intense interest in the random topology, especially in the dynamic changes of the topology within a single run of the algorithm. For example, how can we adaptively vary the neighborhood size K during the search to achieve better performance improvement? Or can we dynamically and randomly change the neighborhood structure, as dictated by the topic of variable neighborhood search (Mladenović and Hansen 1997)? All of these require deep theoretical analysis and extensive computational experiments.

Acknowledgments The work was supported by grants from National Natural Science Foundation (Grant No. 61020106009, 61105073, 61272075) of China.

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

References

- Beyer HG, Schwefel HP (2002) Evolution strategies—a comprehensive introduction. *Nat Comput* 1(1):3–52. doi:[10.1023/A:1015059928466](https://doi.org/10.1023/A:1015059928466)
- Boussaid I, Chatterjee A, Siarry P, Ahmed-Nacer M (2011) Two-stage update biogeography-based optimization using differential evolution algorithm (DBBO). *Comput Oper Res* 38(8):1188–1198. doi:[10.1016/j.cor.2010.11.004](https://doi.org/10.1016/j.cor.2010.11.004)
- Boussaid I, Chatterjee A, Siarry P, Ahmed-Nacer M (2012) Biogeography-based optimization for constrained optimization problems. *Comput Oper Res* 39(12):3293–3304. doi:[10.1016/j.cor.2012.04.012](https://doi.org/10.1016/j.cor.2012.04.012)
- Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Trans Evol Comput* 6(2):182–197. doi:[10.1109/4235.996017](https://doi.org/10.1109/4235.996017)
- Du D, Simon D, Ergezer M (2009) Biogeography-based optimization combined with evolutionary strategy and immigration refusal. In: *IEEE international conference on systems, man and cybernetics*, pp 997–1002. doi:[10.1109/ICSMC.2009.5346055](https://doi.org/10.1109/ICSMC.2009.5346055)
- Ergezer M, Simon D, Du D (2009) Oppositional biogeography-based optimization. In: *IEEE international conference on systems, man and cybernetics*, pp 1009–1014. doi:[10.1109/ICSMC.2009.5346043](https://doi.org/10.1109/ICSMC.2009.5346043)
- Gong W, Cai Z, Ling CX (2010) DE/BBO: a hybrid differential evolution with biogeography-based optimization for global numerical optimization. *Soft Comput* 15(4):645–665. doi:[10.1007/s00500-010-0591-1](https://doi.org/10.1007/s00500-010-0591-1)
- Holland JH (1992) *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control and artificial intelligence*. MIT Press, Cambridge
- Kari L, Rozenberg G (2008) The many facets of natural computing. *Commun ACM* 51(10):72–83. doi:[10.1145/1400181.1400200](https://doi.org/10.1145/1400181.1400200)
- Kennedy J, Eberhart R (1995) Particle swarm optimization. In: *IEEE international conference on neural networks*, vol 4, pp 1942–1948. doi:[10.1109/ICNN.1995.488968](https://doi.org/10.1109/ICNN.1995.488968)
- Kennedy J, Mendes R (2006) Neighborhood topologies in fully informed and best-of-neighborhood particle swarms. *IEEE Trans Syst Man Cybern Part C* 36(4):515–519. doi:[10.1109/TSMCC.2006.875410](https://doi.org/10.1109/TSMCC.2006.875410)
- Li X, Yin M (2012) Multi-operator based biogeography based optimization with mutation for global numerical optimization. *Comput Math Appl* 64(9):2833–2844. doi:[10.1016/j.camwa.2012.04.015](https://doi.org/10.1016/j.camwa.2012.04.015)
- Lohokare M, Pattnaik S, Panigrahi B, Das S (2013) Accelerated biogeography-based optimization with neighborhood search for optimization. *Appl Soft Comput* 13(5):2318–2342. doi:[10.1016/j.asoc.2013.01.020](https://doi.org/10.1016/j.asoc.2013.01.020)
- Ma H, Simon D (2011) Blended biogeography-based optimization for constrained optimization. *Eng Appl Artif Intell* 24(3):517–525. doi:[10.1016/j.engappai.2010.08.005](https://doi.org/10.1016/j.engappai.2010.08.005)
- Ma H, Fei M, Ding Z, Jin J (2012a) Biogeography-based optimization with ensemble of migration models for global numerical optimization. In: *IEEE congress on evolutionary computation*, pp 1–8. doi:[10.1109/CEC.2012.6252930](https://doi.org/10.1109/CEC.2012.6252930)
- Ma HP, Ruan XY, Pan ZX (2012b) Handling multiple objectives with biogeography-based optimization. *Int J Autom Comput* 9(1):30–36. doi:[10.1007/s11633-012-0613-9](https://doi.org/10.1007/s11633-012-0613-9)
- MacArthur R, Wilson E (1967) *The theory of biogeography*. Princeton University Press, New Jersey
- Mendes R, Kennedy J, Neves J (2004) The fully informed particle swarm: simpler, may be better. *IEEE Trans Evol Comput* 8(3):204–210. doi:[10.1109/TEVC.2004.826074](https://doi.org/10.1109/TEVC.2004.826074)
- Mladenović N, Hansen P (1997) Variable neighborhood search. *Comput Oper Res* 24(11):1097–1100. doi:[10.1016/S0305-0548\(97\)00031-2](https://doi.org/10.1016/S0305-0548(97)00031-2)
- Mo H, Xu Z (2011) Research of biogeography-based multi-objective evolutionary algorithm. *J Inform Technol Res* 4(2):70–80. doi:[10.4018/jitr.2011040106](https://doi.org/10.4018/jitr.2011040106)
- Mühlenbein H, Schlierkamp-Voosen D (1993) Predictive models for the breeder genetic algorithm i. Continuous parameter optimization. *Evol Comput* 1(1):25–49. doi:[10.1162/evco.1993.1.1.25](https://doi.org/10.1162/evco.1993.1.1.25)
- Rahnamayan S, Tizhoosh H, Salama MMA (2008) Opposition-based differential evolution. *IEEE Trans Evol Comput* 12(1):64–79. doi:[10.1109/TEVC.2007.894200](https://doi.org/10.1109/TEVC.2007.894200)
- Costa e Silva M, Coelho L, Lebensztajn L (2012) Multiobjective biogeography-based optimization based on predator-prey approach. *IEEE Trans Magn* 48(2):951–954. doi:[10.1109/TMAG.2011.2174205](https://doi.org/10.1109/TMAG.2011.2174205)
- Simon D (2008) Biogeography-based optimization. *IEEE Trans Evol Comput* 12(6):702–713. doi:[10.1109/TEVC.2008.9190004](https://doi.org/10.1109/TEVC.2008.9190004)
- Simon D, Ergezer M, Du D, Rarick R (2011) Markov models for biogeography-based optimization. *IEEE Trans Syst Man Cybern Part B* 41(1):299–306. doi:[10.1109/TSMCB.2010.2051149](https://doi.org/10.1109/TSMCB.2010.2051149)
- Sinha A, Das S, Panigrahi B (2011) A linear state-space analysis of the migration model in an island biogeography system. *IEEE Trans Syst Man Cybern Part A* 41(2):331–337. doi:[10.1109/TSMCA.2010.2058100](https://doi.org/10.1109/TSMCA.2010.2058100)
- Song Y, Liu M, Wang Z (2010) Biogeography-based optimization for the traveling salesman problems. In: *Third international joint conference on computational science and optimization*, vol 1, pp 295–299. doi:[10.1109/CSO.2010.79](https://doi.org/10.1109/CSO.2010.79)
- Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Global Optim* 11(4):341–359. doi:[10.1023/A:1008202821328](https://doi.org/10.1023/A:1008202821328)
- Suganthan PN, Hansen N, Liang JJ, Deb K, Chen Y, Auger A, Tiwari S (2005) Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization. *Tech. Rep. 2005005*, KanGAL, <http://www.ntu.edu.sg/home/EPNSugan>
- Tizhoosh H (2005) Opposition-based learning: A new scheme for machine intelligence. In: *Computational Intelligence for Modelling, Control and Automation*, vol 1, pp 695–701. doi:[10.1109/CIMCA.2005.1631345](https://doi.org/10.1109/CIMCA.2005.1631345)
- Wang H, Wu Z, Rahnamayan S (2011) Enhanced opposition-based differential evolution for solving high-dimensional continuous optimization problems. *Soft Comput* 15(11):2127–2140. doi:[10.1007/s00500-010-0642-7](https://doi.org/10.1007/s00500-010-0642-7)
- Yao X, Liu Y, Lin G (1999) Evolutionary programming made faster. *IEEE Trans Evol Comput* 3(2):82–102. doi:[10.1109/4235.771163](https://doi.org/10.1109/4235.771163)
- Zhu W (2010) Parallel biogeography-based optimization with gpu acceleration for nonlinear optimization. In: *36th design automation conference*, ASME, vol 1, pp 315–323. doi:[10.1115/DETC2010-28102](https://doi.org/10.1115/DETC2010-28102)