Computers & Operations Research 50 (2014) 115–127

Contents lists available at ScienceDirect

Computers & Operations Research

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

Ecogeography-based optimization: Enhancing biogeography-based optimization with ecogeographic barriers and differentiations

Yu-Jun Zheng [a,*], Hai-Feng Ling [b], Jin-Yun Xue [c]

[a] College of Computer Science & Technology, Zhejiang University of Technology, Hangzhou 310023, China
[b] College of Field Engineering, PLA University of Science & Technology, Nanjing 210007, China
[c] Jiangxi Provincial Lab of High-Performance Computing, Jiangxi Normal University, Nanchang 330022, China

ARTICLE INFO

Available online 2 May 2014

Keywords:
Global optimization
Evolutionary algorithms (EAs)
Biogeography-based optimization (BBO)
Emergency airlift

ABSTRACT

Biogeography-based optimization (BBO) is a bio-inspired metaheuristic based on the mathematics of island biogeography. The paper proposes a new variation of BBO, named ecogeography-based optimization (EBO), which regards the population of islands (solutions) as an ecological system with a local topology. Two novel migration operators are designed to perform effective exploration and exploitation in the solution space, mimicking the species dispersal under ecogeographic barriers and differentiations. Experimental results show that the EBO outperforms the basic BBO and several other popular evolutionary algorithms (EAs) on a set of well-known benchmark problems. We also present a real-world application of the proposed EBO to an emergency airlift problem in the 2013 Ya'an–Lushan Earthquake, China.

1. Introduction

Nature-inspired computing has been fascinating computer scientists for a long time, giving rise to popular areas such as artificial neural networks [1], cellular automata [2], molecular computing [3], and evolutionary computation. Taking inspiration from natural evolution processes, evolutionary algorithms (EAs) are a class of heuristic methods for solving complex optimization problems which typically have non-convex and highly nonlinear solution spaces, and which are otherwise computationally difficult to solve by conventional mathematical programming methods [4].

Biogeography-based optimization (BBO) [5] is a relatively new EA borrowing ideas from biogeographic evolution for global optimization. As most EAs, BBO maintains a population of solutions (called "habitats" or "islands" in the metaheuristic) to the optimization problem, and uses a fitness function for evaluating the solutions. A distinct feature of BBO is its migration operator, which works on the principle of immigration and emigration of the species from one island to another, and therefore evolves the islands to find better solutions to the problem. BBO has proven itself to be a competitive heuristic to other well-known EAs on a wide set of problems (e.g. [5–9]). Moreover, several variations of BBO [10–14] and hybridizations of BBO with other metaheuristics [15–18] have been proposed to improve the performance of the optimization method.

In this paper we propose a new variation of BBO, named ecogeography-based optimization (EBO), which designs two novel migration operators that mimic the species dispersal under ecogeographic barriers and differentiations, and thus achieves a much better balance between exploration (global search) and exploitation (local search). Experiments on a set of well-known benchmark problems show that EBO is highly competitive with several state-of-the-art EAs. Also, the proposed EBO has been successfully applied to an emergency airlift problem in the 2013 Ya'an–Lushan Earthquake, Sichuan Province, China.

The rest of this paper is organized as follows: Section 2 introduces the basic BBO algorithm. Section 3 describes our EBO method in detail. Section 4 presents the experimental results, Section 5 depicts the application of EBO to the real-world emergency airlift problem, and finally Section 6 concludes with discussion.

2. Biogeography-based optimization (BBO)

Biogeography is the science of the geographical distribution of biological organisms over space and time. It was first studied by Wallace and Darwin as early as the 19th century. In 1960s, MacArthur and Wilson [19] worked together on mathematical models of island biogeography, which shows that the species richness of an island could be predicted in terms of such factors as habitat area, immigration rate, and extinction rate.

* Corresponding author. Tel.: +86 571 85290085.
E-mail address: yujun.zheng@computer.org (Y.-J. Zheng).

http://dx.doi.org/10.1016/j.cor.2014.04.013
0305-0548/

Based on the mathematics of island biogeography, Simon [5] developed the BBO algorithm, where a solution is analogous to an island (which can refer to any habitat that is geographically isolated from others), the solution components are analogous to a set of suitability index variables (SIVs), and the fitness of the solution is analogous to the species richness or habitat suitability index (HSI) of the island. Central to the algorithm is the equilibrium theory of island biogeography, which indicates that high HSI islands have a high species emigration rate and low HSI islands have a high species immigration rate. Fig. 1 illustrates a simple linear model of species richness in a single island, where the immigration rate λ and the emigration rate μ are functions of the HSI value of the island. Note that BBO can also use other nonlinear migration models, which are more complicated but may produce better optimization results [5,20].

BBO uses two operators: migration and mutation. At each time, the migration operator migrates an SIV from an emigrating island to an immigrating island, which are probabilistically selected according to the rates μ and λ of the islands. The mutation operator randomly modifies an SIV according to the island's steady-state probability p of species count. Algorithm 1 presents the basic procedure of BBO, where *rand*() generates a random real number in the range of [0,1] and $rand_d()$ generates a random value in the range of the dth SIV.

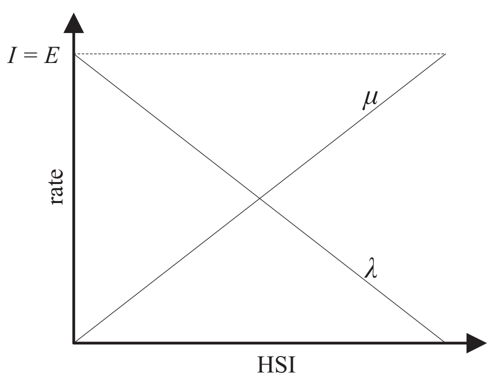

Fig. 1. A linear model of emigration and immigration rates.

Algorithm 1. The basic BBO algorithm.

1 Randomly initialize a population P of n islands (solutions) to the problem;
2 **while** stop criterion is not satisfied **do**
3   Calculate λ_i, μ_i, and p_i for each island X_i;
4   **for each** $X_i \in P$ **do**
5   **for each** SIV $X_{i,d}$ of the island **do**
6   **if** $rand() < \lambda_i$ **then**
7   Select an emigrating island X_j with probability $\propto \mu_j$;
8         $X_{i,d} \leftarrow X_{j,d}$; *//migration*
9   **for each** $X_i \in P$ **do**
10   **for each** SIV $X_{i,d}$ of the island **do**
11   **if** $rand() < p_i$ **then**
12         $X_{i,d} \leftarrow rand_d()$; *//mutation*
13   Evaluate the fitness values of the habitats;
14 **return** the best known solution.

It should be noted that, in recent BBO code released by Simon [21], it is suggested to perform another round of fitness evaluation and solution sorting before the mutations (Line 9 of Algorithm 1) at each generation. Such an implementation can improve the algorithm performance; however, it doubles the number of function evaluations (NFE) of each generation. As we will see in the next section, given a population size of n, our new algorithm can achieve considerable performance improvement with only n function evaluations at a generation.

The migration operator of the basic BBO directly clones an SIV from one island to another, and thus limits the component diversity of new islands. Ma and Simon [11] developed the blended BBO (B-BBO), which uses the blended migration to replace the line 8 of Algorithm 1 as follows:

$$X_{i,d} = \alpha X_{i,d} + (1-\alpha) X_{j,d} \tag{1}$$

where α is a real number between 0 and 1. Though the B-BBO method was proposed for constrained optimization in [11], it also outperforms the basic BBO on many other optimization problems.

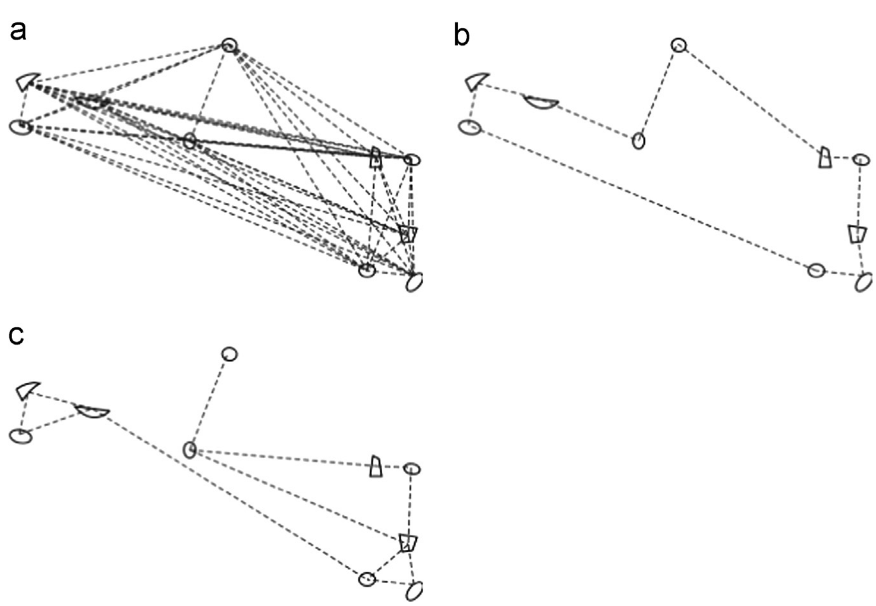

Fig. 2. Different topologies of the islands. (a) Fully connected, (b) local ring, and (c) local random.

3. Ecogeography-based optimization

BBO uses a global topology where all the islands interconnect with each other: if a given island is chosen to be immigrated, any other island has a chance to be an emigrating habitat, as shown in Fig. 2(a). In other words, the basic BBO does not consider ecogeographic isolation in migration, and the biological and ecological features can be shared between any two islands. From the algorithmic point of view, such a fully connected topology often makes most solutions to be strongly attracted by one or several high HSI solutions, thereby causing premature convergence; the situation is aggravated by the fact that the clone-based migration operator cannot produce any new SIVs. In consequence, the population diversity and search performance heavily rely on the mutations, but the simple random mutation operator of the basic BBO is not very effective in exploring the solution space. That is why studies in [16,17,12] seek for new mutation operators to enhance the BBO.

In historical biogeography, Wallace and Darwin considered that species originate in one center of origin, from which some individuals subsequently disperse by chance, and then change through natural selection [22]. From this viewpoint, species richness is correlated with many biological, ecological and geographical factors, and speciation is characterized by the evolution of barriers to genetic exchange between previously interbreeding populations [23].

Embedding this idea into the algorithm, we suggest using a local topology of isolated islands, where each island is connected to a subset of islands in the population: migratory routes between adjacent islands can be considered as corridors, while those between nonadjacent islands can be considered as filter bridges or sweepstakes routes [24]. One of the simplest local topologies is the ring topology, which connects each island to just two other islands, as shown in Fig. 2(b). Another idea is use a random topology, i.e., the neighbors for each island are chosen at random, as shown in Fig. 2(c). Typically, the neighborhood structure in a ring topology is fixed in one run of the algorithm, but that in a random topology may be subject to change during the evolution.

In contrast to the global topology, the local topology can effectively maintain the diversity of the population and avoid premature convergence [14], but it may also slow the convergence speed of the algorithm. In order to preserve the fast convergence rate of BBO, we design a new "global" migration operator to replace the line 8 of Algorithm 1. That is, based on the emigration rates of the islands, we select a neighbor X_{nb} and a non-neighbor X_{far} of X_i, and update the component of X_i as follows:

$$X_{i,d} = \begin{cases} X_{far,d} + \alpha(X_{nb,d} - X_{i,d}) & fit(X_{far}) > fit(X_{nb}) \\ X_{nb,d} + \alpha(X_{far,d} - X_{i,d}) & fit(X_{far}) \le fit(X_{nb}) \end{cases} \tag{2}$$

where a is a coefficient in the range from 0 to 1. Here $(X_{nb,d} - X_{i,d})$ or $(X_{far,d} - X_{i,d})$ is regraded as the "ecological differentiation" between the two islands, and a is called the "evolutionary force" coefficient.

Eq. (2) indicates that the island accepts immigrants from both neighboring and non-neighboring islands: the fitter one between X_{far} and X_{nb} acts as the "primary" immigrant, while the other acts as the "secondary" immigrant that needs to compete with the "original inhabitants" of X_i.

Global migration is preferred in early stages of evolution, where species are easier to disperse to a wide range of new habitats. With the increase of species richness, the ecological system has more stability or more resistance to invasions. The increase of invasion resistance facilitates divergence, which increases invasion resistance even further. Such a feedback loop, given enough time, leads to a stable connection between the islands. Thereby, in later stages of evolution, the islands are more likely to accept immigrants from their neighbors. This is mimicked by the following "local" migration operator:

$$X_{i,d} = X_{i,d} + \alpha(X_{nb,d} - X_{i,d}) \tag{3}$$

In general, global migration facilitates exploration (global search) and local migration facilitates exploitation (local search). We introduce a parameter η, named the immaturity index, to represent the island immaturity of the ecological system (population), which is inversely proportional to the invasion resistance of the system. During the search process of EBO, the value of η can be fixed, e.g. $\eta = 0.5$ gives equal chances for global and local migration. We can also use a dynamically changing value of η, such as

$$\eta = \eta^{max} - \frac{t}{t^{max}}(\eta^{max} - \eta^{min}) \tag{4}$$

where t is the current generation number and t^{max} is the total generation number of the algorithm, and η^{max} and η^{min} are respectively the upper limit and the lower limit of η.

Whenever an island is to be immigrated, we generate a random number uniformly distributed in [0,1]. If the number is less than η, the global migration (2) is used; otherwise the local migration (3) is used.

Besides, there are two other differences in comparison with the basic BBO:

- The migration operator of EBO does not directly modify an existing island X_i. Instead, it produces a new island X_i', and keeps the better one of X_i and X_i' to the next generation.
- The random mutation operator of the basic BBO is no longer needed, because the two new migration operators can provide enough diversity to the population.

From another perspective, Eqs. (2) and (3) can be seen as two variant differential evolution (DE) operators [25], but they are different from those popular DE methods in that the current solution X_i is included into the differential part, and the two other solutions X_{far} and X_{nb} are selected based on the migration model. In the most widely used DE/rand/1/bin method, the three solutions on the right side of the equations are all randomly selected. The combination of the two elaborately designed operators in EBO is expected to enhance the exploitation ability of the DE operator without harming its exploration capability very much.

Algorithm 2 presents the procedure of our EBO method. In comparison with the basic BBO, EBO adds up to three parameters, namely α, η^{max} and η^{min}, the fine-tuning of which may be beneficial but costly. However, we find that using a random value in (0,1) for the coefficient α is very competitive on most test functions; empirically, we also suggest to set η^{max} between 0.7 and 0.8, and η^{min} between 0.2 and 0.4. Moreover, the maximum immigration rate and the maximum emigration rate in BBO are typically set to 1, and the EBO algorithm does not need to use the mutation rate. Thereby, the parameter selection of EBO is not very difficult in most cases.

Algorithm 2. The proposed EBO algorithm.

```
Randomly initialize a population P of n islands to the
problem;
while stop criterion is not satisfied do
  Calculate η according to Eq. (4);
  Calculate λ_i and μ_i for each island X_i;
  for each X_i ∈ P do
    Clone X_i to a new island X_i';
    for each SIV X_{i,d} of the island do
      if rand() < λ_i then
```

Select a neighboring island X_{nb} with probability $\propto \mu$;
if $rand() < \eta$ **then**
Select a non-neighbor X_{far} with probability $\propto \mu$;
if X_{far} is fitter than X_{nb} **then**
$X_{i,d} \leftarrow X_{far,d} + \alpha(X_{nb,d} - X_{i,d})$;
else
$X_{i,d} \leftarrow X_{nb,d} + \alpha(X_{far,d} - X_{i,d})$;
else
$X_{i,d} \leftarrow X_{i,d} + \alpha(X_{nb,d} - X_{i,d})$;
if at least one dimensionof X_i' is changed **then**
Evaluate the new X_i;
if X_i' is better than X_i **then**
$X_i \leftarrow X_i'$;
return the best known solution.

4. Numerical experiments

4.1. Experimental settings

For evaluating the proposed EBO algorithm, we compare it with the basic BBO, the B-BBO, the DE algorithm with the DE/rand/1/bin scheme [25], and the hybrid DE/BBO method [16] on a set of well-known benchmark functions. The 13 test functions, denoted as f_1–f_{13}, are scalable high-dimensional problems taken from Yao et al. [26] and briefly described in Table 1. In this paper, we conduct experiments on 10, 30, and 50 dimensional functions. For a fair comparison, on each test problem we set a maximum number of function evaluations (MNFE) as $5000D$ which is the same for all the algorithms, where D is the dimension of the function. In addition, we record the required number of function evaluations (RNFE) of the algorithms to reach the required function error value, which is set to 10^{-8} for all the functions.

We implement two versions of EBO, one with the ring topology and the other with the local random topology, denoted by EBO1 and EBO2 respectively. For EBO2, the topology is randomly generated such that each island has probably K neighbors, i.e., the probability of any two islands being connected is $K/(n-1)$, as used by the neighborhood topology of the standard PSO 2007 [27]. Moreover, the random topology will be reset after every non-improvement generation, i.e., when no new best solution has been found in the generation. Empirically, we set the neighborhood size $K=2$ in the experiments.

All the six EAs use the same population size of 50. We also set both the maximum immigration rate and the maximum emigration rate to 1 for BBO, DE/BBO, and EBO. For BBO and DE/BBO, we set the mutation rate to 0.01. The coefficient α in B-BBO could be random or deterministic, and here we set it to 0.5 which is preferable on most problems [11]. The other control parameters of BBO, B-BBO, DE and DE/BBO are set as suggested in the literature [25,5,11,16].

The experiments are conducted on a computer of Intel Core i5-2520M processor and 4 GB DDR3 memory. All the algorithms have been run for 60 times (with different random seeds) on every problem, and the resulting function values are averaged over the 60 runs.

4.2. Impact of the immaturity index η

First we test the impact of different η values on the performance of EBO. We select 9 functions including f_1–f_4, f_6, f_7, and f_{10}–f_{12}, and on each function respectively set η to 7 fixed values including 0, 0.2, 0.4, 0.5, 0.6, 0.8, and 1.0, in addition with a linearly decreasing value with $\eta^{\max} = 0.7$ and $\eta^{\min} = 0.4$. When η is fixed to 0, the algorithm only uses the local migration operator, and when $\eta = 1$ it only uses the global migration operator.

Fig. 3(a) and (b) presents the mean function error values obtained by EBO1 and EBO2 respectively. As we can see, for both

Table 1
Summary of the 13 benchmark functions.

Name	Type	Expression	Search range
Sphere	Unimodal	$f_1(x) = \sum_{i=1}^{D} x_i^2$	$[-100, 100]^D$
Schwefel 2.22	Unimodal	$f_2(x) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	$[-10, 10]^D$
Schwefel 1.2	Unimodal	$f_3(x) = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} x_j \right)^2$	$[-100, 100]^D$
Schwefel 2.21	Unimodal	$f_4(x) = \max_i \{ \lvert x_i \rvert, 1 \le i \le D \}$	$[-100, 100]^D$
Rosenbrock	Unimodal	$f_5(x) = \sum_{i=2}^{D-1} (100(x_i^2 - x_{i-1})^2 + (x_i - 1)^2)$	$[-30, 30]^D$
Step	Discrete	$f_6(x) = \sum_{i=1}^{D} (\lfloor x_i + 0.5 \rfloor)^2$	$[-100, 100]^D$
Quartic	Noisy	$f_7(x) = \sum_{i=1}^{D} i x_i^4 + rand[0, 1)$	$[-1.28, 1.28]^D$
Schwefel	Multimodal	$f_8(x) = 418.9829 \times D - \sum_{i=1}^{D} x_i \sin(\lvert x_i \rvert^{1/2})$	$[-500, 500]^D$
Rastrigin	Multimodal	$f_9(x) = \sum_{i=1}^{D} (x_i^2 - 10\cos(2\pi x_i) + 10)$	$[-5.12, 5.12]^D$
Ackley	Multimodal	$f_{10}(x) = -20 \exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}) - \exp\left(\frac{1}{D}\sum_{i=1}^{D} \cos(2\pi x_i)\right) + 20 + e$	$[-32, 32]^D$
Griewank	Multimodal	$f_{11}(x) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600, 600]^D$
Penalized1	Multimodal	$f_{12}(x) = \frac{\pi}{30}(10\sin^2(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2 (1 + 10\sin^2(\pi y_{i+1})) + (y_D - 1)^2) + \sum_{i=1}^{D-1} u(x_i, 10, 100, 4)$ where $y_i = 1 + \frac{1}{4}(x_i + 1)$	$[-50, 50]^D$
Penalized2	Multimodal	$f_{13}(x) = 0.1\left(\sin^2(3\pi x_1) + \sum_{i=1}^{D-1} (x_i - 1)\right)^2 (1 + \sin^2(3\pi x_{i+1})) + (x_D - 1)^2 (1 + \sin^2(2\pi x_D))) + \sum_{i=1}^{D-1} u(x_i, 5, 100, 4)$ where $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \le x_i \le a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	$[-50, 50]^D$

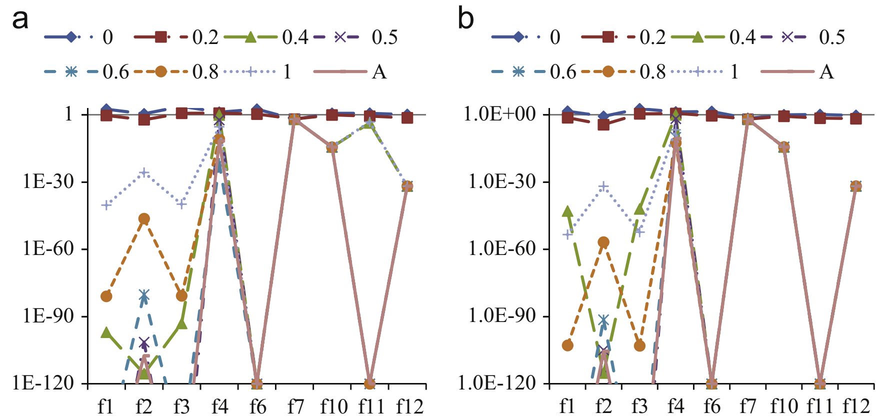

Fig. 3. Mean function values obtained by (a) EBO1 and (b) EBO2 with different η values, where 'A' denotes the linearly decreasing value of η.

the methods, $\eta = 0$ leads to the worst results on all the functions. In general, the EBO methods with $\eta = 0.4$ or 0.5 perform well on most functions, but on f_4 and f_7 a large η between 0.8 and 1.0 is preferable. However, no single fixed value of η can always be superior to other fixed values. In comparison, when η linearly decreases from 0.7 to 0.4, the overall performance of EBO on the 9 functions is more competitive and robust. Thus we use this dynamic strategy in the following comparative experiments.

4.3. Comparison on the 10-D functions

Table 2 presents the experimental results on the 10-D functions, where "mean" denotes the result function error value averaged over the 60 runs, "std" is the corresponding standard deviation, and "RNFE" is the average NFE used to reach the required function error value. The best mean error and RNFE values are shown in bold. As we can see,

- All the six algorithms reach the same optimum on function f_6, where EBO1 uses the minimum RNFE.
- DE, DE/BBO, EBO1 and EBO2 also reach the same optimum on f_{12} and f_{13}, where EBO2 uses the minimum RNFE on both the functions.
- DE/BBO, EBO1 and EBO2 also reach the same optimum on f_8 and f_9, where EBO1 uses the minimum RNFE on f_8, and DE does so on f_9.

On the remaining 8 test functions, EBO1 has both the best mean error and RNFE values on f_1–f_3, and EBO2 uniquely achieves the best mean error values on f_4, f_7 and f_{10}. DE achieves the best mean error value on f_5, where none of the algorithms can reach the required accuracy. DE/BBO does so on f_{11}, but its RNFE is larger than the two EBO methods.

We have also conducted paired t-tests on the differences between the mean error values of EBO1/EBO2 and the other comparative algorithms, the resulting p-values of which are shown in Table 3. The results indicate that, EBO1 has statistically significant improvement over the BBO and B-BBO on 11 functions, over DE on 8 functions, and over DE/BBO on 7 functions; EBO2 has significant improvement over the BBO and B-BBO on 11 functions, over DE 8 functions, and over DE/BBO on 7 functions.

The above results show that the performance of EBO methods is significantly better than the other four EAs. By comparing the two EBO versions, in terms of statistical tests, EBO1 outperforms EBO2 on two functions, EBO2 outperforms EBO1 on one function, and on the remaining 10 functions there is no significant difference between them. In general, EBO2 has more advantage in the mean error values, while EBO1 is more preferable in terms of RNFE.

4.4. Comparison on the 30-D functions

Table 4 presents the mean function error and RNFE values of the six algorithms on the 30-D functions, and Table 5 presents their statistical test results. On this group,

- B-BBO, DE/BBO, EBO1 and EBO2 reach the same optimum on function f_6, where EBO1 uses the minimum RNFE.
- DE/BBO, EBO1 and EBO2 also reach the same optimum on f_8 and f_{11}, where EBO1 and EBO2 uses the minimum RNFE.
- DE/BBO achieves the best mean error value on f_5, where none of the algorithms can reach the required accuracy.
- On the remaining 9 test functions, both the mean error and RNFE values of the EBO methods are better than all the other four EAs. Individually, EBO1 has the best mean error values on 3 functions, EBO2 does so on 2 functions, and the two methods achieve the same best results on 4 functions.

According to the statistical test results, both EBO1 and EBO2 have significant performance improvement over BBO on all the 13 functions, and over B-BBO on 12 functions. EBO1 also has significant improvement over DE and DE/BBO on 10 functions and 8 functions respectively, and EBO2 does so on 11 functions and 8 functions respectively.

We can find that the EBO methods have much performance advantage over the other four EAs on the 30-D functions, and the advantage is more obvious than that on the 10-D functions. By comparing EBO1 and EBO2, they outperform the counterpart on 3 functions, and there is no significant difference between them on the remaining 7 functions.

We have also present the convergence curves of algorithms on the 30-D functions f_1–f_{12} in Fig. 4(a)–(l) respectively. The convergence curves on f_{13} are very similar to f_{12} and thus are omitted here. As we can see from the curves, the overall convergence

Table 2
The experimental results of the six EAs on the 10-*D* problems.

f	Metrics	BBO	B-BBO	DE	DE/BBO	EBO1	EBO2
f_1	Mean	1.35E−01	1.06E−03	1.52E−41	3.46E−49	**1.16E−127**	9.25E−124
	std	8.82E−02	1.03E−03	2.60E−41	8.46E−49	3.46E−127	4.93E−123
	RNFE	–	–	11,272 ± 398	10,667 ± 260	**7143** ± 174	7193 ± 199
f_2	Mean	7.61E−02	1.86E−03	3.32E−25	1.11E−24	**4.62E−68**	5.78E−66
	std	2.34E−02	1.24E−03	2.86E−25	8.66E−25	8.91E−68	1.23E−65
	RNFE	–	–	18212 ± 518	14,581 ± 292	**9863** ± 223	9979 ± 214
f_3	Mean	6.27E−01	6.55E−03	1.03E−48	4.18E−41	**2.13E−126**	1.78E−122
	std	4.08E−01	8.12E−03	1.71E−48	7.42E−41	8.76E−126	5.47E−122
	RNFE	–	–	11,933 ± 4.36	11,325 ± 271	**7563** ± 172	7629 ± 201
f_4	Mean	8.54E−01	2.18E−01	8.53E−06	8.52E−13	4.93E−30	**1.11E−34**
	std	2.64E−01	7.61E−02	3.95E−05	6.58E−13	3.76E−29	7.64E−34
	RNFE	–	–	28,186 ± 7394	35,110 ± 865	**18,728** ± 496	18,888 ± 526
f_5	Mean	4.27E+01	1.92E+02	**1.99E+00**	3.91E+00	2.06E+00	2.32E+00
	std	2.96E+01	8.05E+01	1.33E+00	8.80E−01	2.19E+00	2.22E+00
	RNFE	–	–	–	–	–	–
f_6	Mean	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	std	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	RNFE	27,605 ± 8042	10,902 ± 2641	4094 ± 316	3834 ± 216	**2534** ± 135	2566 ± 119
f_7	Mean	3.84E−02	3.37E−02	8.01E−02	6.67E−02	3.23E−02	**2.66E−02**
	std	2.67E−02	2.22E−02	5.88E−02	4.77E−02	2.23E−02	1.73E−02
	RNFE	–	–	–	–	–	–
f_8	Mean	4.00E−01	1.09E−02	2.63E−07	**0.00E+00**	**0.00E+00**	**0.00E+00**
	std	3.40E−01	1.06E−02	2.00E−06	0.00E+00	0.00E+00	0.00E+00
	RNFE	–	–	37,353 ± 574	13,503 ± 451	**13,273** ± 1913	13,347 ± 1561
f_9	Mean	6.18E−02	4.18E−04	9.80E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**
	std	3.85E−02	7.57E−04	6.58E+00	0.00E+00	0.00E+00	0.00E+00
	RNFE	–	–	–	**16,541** ± 771	19,087 ± 1119	19,259 ± 1178
f_{10}	Mean	2.12E−01	1.12E−01	4.00E−15	4.00E−15	3.70E−15	**3.35E−15**
	std	8.80E−02	2.78E−02	0.00E+00	0.00E+00	9.90E−16	1.39E−15
	RNFE	–	–	18,437 ± 535	16,743 ± 403	**10,118** ± 185	10,300 ± 204
f_{11}	Mean	2.45E−01	6.21E−02	2.10E−02	**0.00E+00**	1.32E−10	4.46E−12
	std	8.86E−02	1.14E−01	3.12E−02	0.00E+00	6.56E−10	2.16E−11
	RNFE	–	–	–	30,922 ± 3687	30,596 ± 8049	**29,827** ± 7706
f_{12}	Mean	5.11E−03	4.33E−01	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	std	4.82E−03	9.50E−02	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	RNFE	–	–	10,783 ± 495	10,604 ± 311	7968 ± 369	**7951** ± 301
f_{13}	Mean	2.20E−02	3.64E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	std	1.66E−02	1.12E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	RNFE	–	–	11,827 ± 533	11,326 ± 402	8456 ± 395	**8425** ± 371

Table 3
Statistical test results on the 10-*D* problems.

f	EBO1 vs.					EBO2 vs.				
	BBO	B-BBO	DE	DE/BBO	EBO2	BBO	B-BBO	DE	DE/BBO	EBO1
f_1	†4.56E−22	†5.46E−13	†9.87E−04	†7.35E−06	7.46E−02	†4.56E−22	†5.46E−13	†9.87E−04	†7.35E−06	9.25E−01
f_2	†1.17E−49	†1.68E−21	†2.59E−15	†1.49E−17	†2.15E−04	†1.17E−49	†1.68E−21	†2.59E−15	†1.49E−17	1.00E+00
f_3	†2.79E−22	†3.37E−09	†4.56E−06	†1.38E−05	†6.60E−03	†2.79E−22	†3.37E−09	†4.56E−06	†1.38E−05	9.93E−01
f_4	†2.51E−49	†3.59E−44	†4.86E−02	†8.43E−18	8.44E−01	†2.51E−49	†3.59E−44	†4.86E−02	†8.43E−18	1.56E−01
f_5	†3.88E−19	†2.01E−36	5.83E−01	†8.47E−09	2.62E−01	†5.62E−19	†2.26E−36	8.36E−01	†4.99E−07	7.38E−01
f_6	–	–	–	–	–	–	–	–	–	–
f_7	8.92E−02	3.62E−01	†1.37E−06	†2.39E−05	9.38E−01	†2.50E−03	†2.63E−02	†3.04E−08	†3.19E−07	6.17E−02
f_8	†1.19E−15	†4.77E−13	1.60E−01	–	–	†1.19E−15	†4.77E−13	1.60E−01	–	–
f_9	†1.75E−23	†1.95E−05	†2.24E−21	–	–	†1.75E−23	†1.95E−05	†2.24E−21	–	–
f_{10}	1.74E−32	†5.17E−19	†9.00E−03	†9.00E−03	9.46E−01	†1.74E−32	†5.17E−19	†9.00E−03	†9.00E−03	5.45E−02
f_{11}	†8.94E−43	†2.36E−05	†3.92E−07	9.39E−01	9.33E−01	†8.94E−43	†2.36E−05	†3.92E−07	9.44E−01	†6.75E−02
f_{12}	†4.10E−03	†8.03E−65	–	–	–	†4.10E−03	†8.03E−65	–	–	–
f_{13}	†1.68E−04	†1.31E−37	–	–	–	†1.68E−04	†1.31E−37	–	–	–

The symbol † indicates that the EBO method has statistically significant improvement over the corresponding algorithms at 95% confidence level.

speeds of the EBO methods are much better than the other algorithms on almost all of the problems. On some problems (such as f_5–f_8), BBO and B-BBO converge fast at the very early stage, but their curves soon become flat. DE/BBO typically converges faster than BBO and slower than DE, but it often converges longer and thus reaches better results than DE. In comparison, the

Table 4
The experimental results of the six EAs on the 30-*D* problems.

f	Metrics	BBO	B-BBO	DE	DE/BBO	EBO1	EBO2
f_1	Mean	1.19E+00	1.08E+00	5.66E−51	7.04E−31	**1.46E−187**	3.34E−174
	std	4.65−01	2.42E−01	2.95E−50	1.08E−30	0.00E+00	0.00E+00
	RNFE	–	–	33,823 ± 1188	37,799 ± 571	**16,328** ± 294	17,017 ± 285
f_2	Mean	3.09E−01	6.10E−02	7.44E−28	1.73E−19	**4.17E−102**	9.82E−95
	std	5.57E−02	1.24E−02	7.10E−28	9.34E−20	6.81E−102	1.14E−94
	RNFE	–	–	51,844 ± 1187	52,597 ± 796	**23,720** ± 294	24,728 ± 331
f_3	Mean	2.46E+01	2.86E+01	6.85E−49	2.94E−30	**1.07E−183**	3.25E−173
	std	8.75E+00	6.76E+01	4.66E−48	4.62E−30	0.00E+00	0.00E+00
	RNFE	–	–	36,770 ± 1319	41,112 ± 711	**17,752** ± 349	18,603 ± 317
f_4	Mean	3.05E+00	1.40E+00	8.92E+00	8.03E−04	4.17E−12	**1.63E−13**
	std	5.80E−01	1.57E−01	4.14E+00	2.83E−04	1.12E−11	3.76E−13
	RNFE	–	–	–	–	**71,563** ± 2154	73,312 ± 1797
f_5	Mean	2.60E+02	9.02E+03	2.67E+01	**2.11E+01**	2.24E+01	2.15E+01
	std	2.64E+02	2.49E+03	1.82E+01	4.31E−01	6.25E−01	7.01E−01
	RNFE	–	–	–	–	–	–
f_6	Mean	8.50E−01	**0.00E+00**	1.83E−01	**0.00E+00**	**0.00E+00**	**0.00E+00**
	std	8.20E−01	0.00E+00	4.31E−01	0.00E+00	0.00E+00	0.00E+00
	RNFE	–	126,361 ± 13529	37,506 ± 51725	13,997 ± 511	**6116** ± 196	6364 ± 189
f_7	Mean	1.74E−02	1.37E−02	2.56E−02	2.11E−02	9.80E−03	**7.09E−03**
	std	1.14E−02	8.83E−03	1.70E−02	1.41E−02	7.60E−03	5.02E−03
	RNFE	–	–	–	–	–	–
f_8	Mean	1.92E+00	1.76E+02	2.40E+03	**0.00E+00**	**0.00E+00**	**0.00E+00**
	std	7.83E−01	7.44E−01	1.80E+03	0.00E+00	0.00E+00	0.00E+00
	RNFE	–	–	–	67,811 ± 5147	**43,335** ± 6631	44,783 ± 5673
f_9	Mean	3.91E−01	4.32E−02	2.09E+01	2.96E−11	**0.00E+00**	**0.00E+00**
	std	1.58E−01	2.50E−02	1.38E+01	2.18E−10	0.00E+00	0.00E+00
	RNFE	–	–	–	124,391 ± 10,831	**90,955** ± 9068	10,3282 ± 10,443
f_{10}	Mean	2.69E−01	2.70E−02	6.78E−15	6.90E−15	**4.00E−15**	**4.00E−15**
	std	7.02E−02	7.54E−03	1.86E−15	1.39E−15	0.00E+00	0.00E+00
	RNFE	–	–	52,873 ± 1778	60,562 ± 966	**23,125** ± 388	24,348 ± 389
f_{11}	Mean	8.16E−01	8.44E−01	2.87E−03	**0.00E+00**	**0.00E+00**	**0.00E+00**
	std	1.44E−01	9.47E−02	8.23E−03	0.00E+00	0.00E+00	0.00E+00
	RNFE	–	–	61,631 ± 49,174	41,591 ± 2752	18,793 ± 5083	**18,485** ± 3252
f_{12}	Mean	8.82E−02	5.65E+00	1.74E−30	1.02E−31	**1.57E−32**	**1.57E−32**
	std	3.07E−02	1.05E+00	2.19E−30	3.38E−31	2.21E−47	2.21E−47
	RNFE	–	–	40,132 ± 983	36,401 ± 3053	**18,745** ± 412	19,420 ± 536
f_{13}	Mean	4.08E−01	3.91E+00	1.64E−24	4.16E−30	**1.35E−32**	**1.35E−32**
	std	1.35E−01	1.30E+00	1.27E−23	4.77E−30	8.28E−48	8.28E−48
	RNFE	–	–	46,936 ± 14,924	42,785 ± 933	**19,755** ± 501	20,804 ± 527

Table 5
Statistical test results on the 30-*D* problems.

f	EBO1 vs.					EBO2 vs.				
	BBO	B-BBO	DE	DE/BBO	EBO2	BBO	B-BBO	DE	DE/BBO	EBO1
f_1	†1.03E−39	†4.17E−64	6.99E−02	†8.71E−07	†0.00E+00	†1.03E−39	†4.17E−64	6.99E−02	†8.71E−07	1.00E+00
f_2	†3.60E−74	†2.22E−68	†2.70E−13	†5.42E−28	†4.16E−10	†3.60E−74	†2.22E−68	†2.70E−13	†5.42E−28	1.00E+00
f_3	†2.20E−43	†1.99E−61	1.28E−01	†1.39E−06	†0.00E+	†2.20E−43	†1.99E−61	†1.28E−01	†1.39E−06	1.00E+00
f_4	†1.23E−71	†2.03E−97	†4.25E−33	†7.29E−44	9.97E−01	†1.23E−71	†2.03E−97	†4.25E−33	†7.29E−44	†3.30E−03
f_5	†9.06E−11	†3.69E−54	†3.59E−02	1.00E+00	1.00E+00	†7.92E−11	†3.66E−54	†1.47E−02	1.00E+00	†6.15E−12
f_6	†4.07E−13	–	†6.58E−04	–	–	†4.07E−13	–	†6.58E−04	–	–
f_7	†1.69E−05	†5.50E−03	†7.03E−10	†1.14E−07	9.89E−01	†1.38E−09	†8.72E−07	†3.04E−13	†1.94E−11	†1.15E−02
f_8	†5.82E−38	†1.22E−36	†1.59E−18	–	–	†5.82E−38	†1.22E−36	†1.59E−18	–	–
f_9	†2.62E−38	†9.91E−26	†6.72E−22	1.47E−01	–	†2.62E−38	†9.91e-26	†6.72E−22	1.47E−01	–
f_{10}	†7.80E−57	†7.02E−54	†1.71E−21	†4.18E−32	5.00E−01	†7.80E−57	†7.02E−54	†1.71E−21	†4.18E−32	5.00E−01
f_{11}	†2.81E−75	†1.35E−97	†3.90E−03	–	–	†2.81E−75	†1.35E−97	†3.90E−03	–	–
f_{12}	†2.18E−44	†5.93E−73	†6.96E−09	†2.59E−02	5.00E−01	†2.18E−44	†5.93E−73	†6.96E−09	†2.59E−02	5.00E−01
f_{13}	†1.50E−46	†3.70E−46	1.59E−01	†3.16E−10	5.00E−01	†1.50E−46	†3.70E−46	1.59E−01	†3.16E−10	5.00E−01

The symbol "†" indicates that the EBO method has statistically significant improvement over the corresponding algorithms at 95% confidence level.

curves of EBO fall not only fast but also deep, which demonstrates that they achieve a much better balance between exploration and exploitation.

The shapes of convergence curves of the 10-*D* and 50-*D* functions are also similar to those of the 30-*D* functions, and thus we do not present them here.

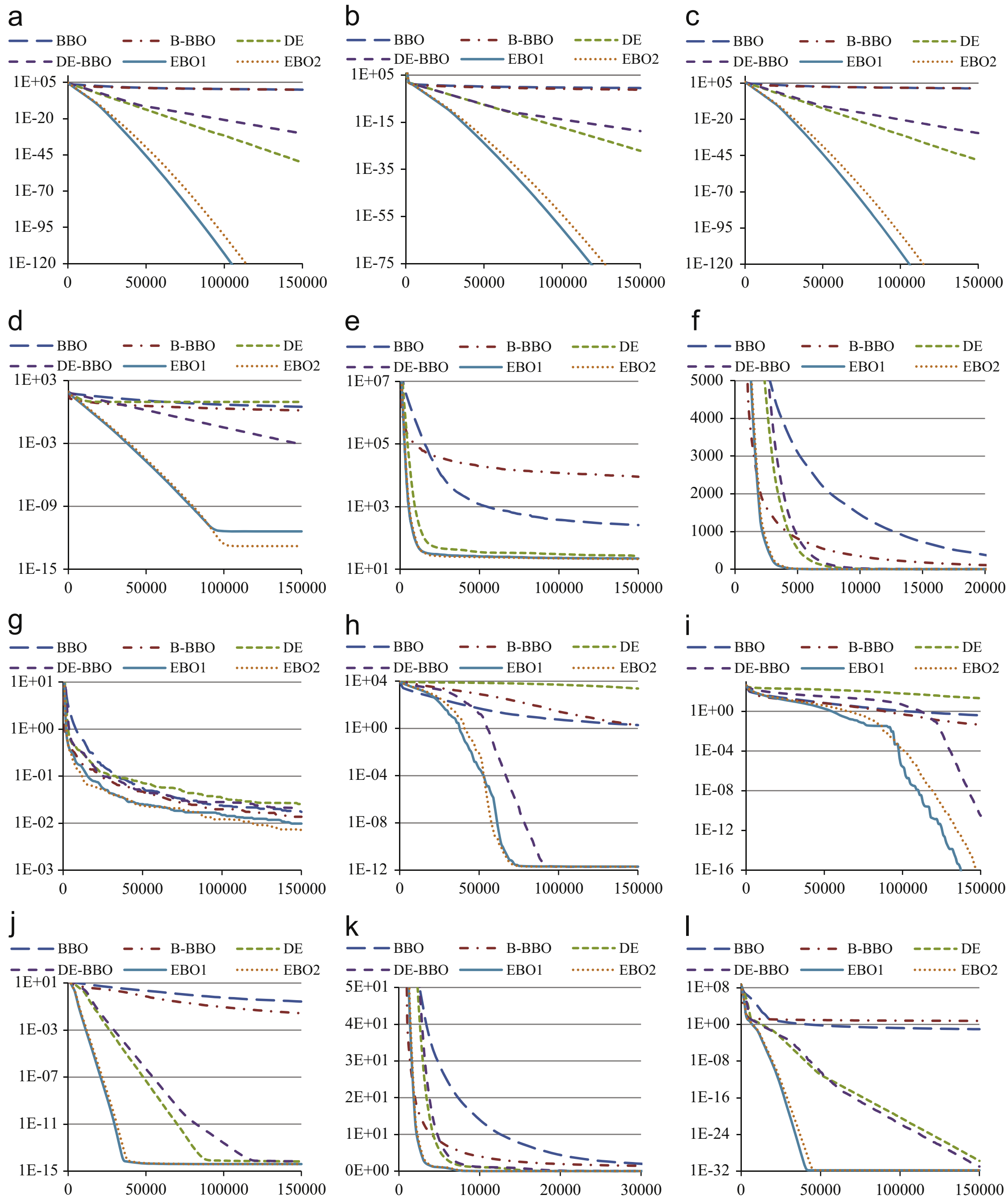

Fig. 4. Convergence curves of the comparative algorithms on the 30-D functions. (a) Sphere. (b) Schwefel 2.22. (c) Schwefel 1.2. (d) Schwefel 2.21. (e) Rosenbrock. (f) Step. (g) Quartic. (h) Schwefel. (i) Rastrigin. (j) Ackley. (k) Griewank. (l) Penalized1.

4.5. Comparison on the 50-D functions

Table 6 presents the mean function error and RNFE values of the six algorithms on the 50-*D* functions, and Table 7 presents their statistical test results. On this group,

- DE/BBO, EBO1 and EBO2 reach the same optimum on function f_6, where EBO1 uses the minimum RNFE.
- BBO achieves the best mean error value on f_8, where none of the algorithms can reach the required accuracy.
- On the remaining 11 test functions, the EBO methods outperform all the other four EAs. Individually, EBO1 has the best mean error values on 2 functions, EBO2 does so on 6 functions, and the two methods achieve the same best results on 3 functions.

According to the statistical test results, EBO1 has significant performance improvement over BBO, B-BBO, DE and DE/BBO on 11, 12, 11 and 9 functions respectively, and EBO2 does so on 11, 12, 11 and 10 functions respectively. EBO1 outperforms EBO2 on 2 functions, while EBO2 outperforms EBO1 on 5 functions.

4.6. Discussion

In summary, our EBO methods exhibit much performance advantage over the other four comparative algorithms on the

Table 6
The experimental results of the six EAs on the 50-*D* problems.

Metrics	BBO	B-BBO	DE	DE/BBO	EBO1	EBO2
f_1						
Mean	2.67E+00	1.13E+01	2.97E−46	2.38E−27	**6.15E−184**	3.86E−182
std	7.14E−01	2.12E+00	1.83E−45	1.65E−27	0.00E+00	0.00E+00
RNFE	–	–	60,775 ± 4132	77,685 ± 1177	**24,793** ± 406	26,615 ± 423
f_2						
Mean	6.16E−01	3.79E−01	8.81E−29	3.14E−17	**5.47E−110**	1.57E−104
std	7.75E−02	5.35E−02	2.80E−28	1.63E−17	2.68E−109	3.51E−104
RNFE	–	–	81,840 ± 2472	107,976 ± 1524	**36,543** ± 366	39,006 ± 481
f_3						
Mean	1.11E+02	4.20E+02	1.46E−44	1.53E−26	3.69E−178	**2.47E−179**
std	3.06E+01	7.38E+01	7.95E−44	1.69E−26	0.00E+00	0.00E+00
RNFE	–	–	66,516 ± 3714	85,038 ± 1255	**27,198** ± 439	29,228 ± 392
f_4						
Mean	4.90E+00	1.16E+00	1.96E+01	3.75E−01	1.06E−05	**3.23E−07**
std	5.84E−01	1.30E−01	4.71E+00	7.78E−02	1.61E−05	6.00E−07
RNFE	–	–	–	–	–	215,916 ± 45246
f_5						
Mean	4.43E+02	1.49E+03	7.10E+01	4.10E+01	4.00E+01	**3.88E+01**
std	2.32E+02	3.23E+02	3.56E+01	9.73E+00	7.11E−01	5.39E−01
RNFE	–	–	–	–	–	–
f_6						
Mean	2.13E+00	1.44E+01	5.18E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**
std	1.40E+00	3.30E+00	1.16E+01	0.00E+00	0.00E+00	0.00E+00
RNFE	–	–	–	29,273 ± 1060	**9358** ± 277	10,138 ± 364
f_7						
Mean	1.38E−02	1.49E−02	0.015186991	1.33E−02	4.94E−03	**3.40E−03**
std	7.59E−03	6.75E−03	0.011797576	9.58E−03	3.98E−03	2.70E−03
RNFE	–	–	–	–	–	–
f_8						
Mean	**4.00E+00**	7.60E+01	4.21E+03	1.01E+02	4.25E+02	3.27E+02
std	1.16E+00	4.42E+01	2.69E+03	1.21E+02	1.75E+02	1.68E+02
RNFE	–	–	–	–	–	–
f_9						
Mean	8.05E−01	9.65E−01	3.12E+01	1.43E+01	7.30E−01	**6.09E−01**
std	2.69E−01	5.14E−01	7.83E+00	1.19E+01	7.74E−01	7.08E−01
RNFE	–	–	–	–	–	–
f_{10}						
Mean	2.80E−01	7.66E−02	9.74E−02	1.07E−14	6.96E−15	**6.72E−15**
std	5.23E−02	1.25E−02	2.97E−01	2.78E−15	1.34E−15	1.52E−15
RNFE	–	–	–	126,620 ± 1889	**34,884** ± 480	37,540 ± 498
f_{11}						
Mean	9.77E−01	1.10E+00	3.32E−03	1.85E−18	**0.00E+00**	**0.00E+00**
std	7.57E−02	2.00E−02	6.70E−03	1.43E−17	0.00E+00	0.00E+00
RNFE	–	–	104,991 ± 80,730	81,130 ± 2079	**25,187** ± 3691	27,703 ± 4315
f_{12}						
Mean	2.15E−01	1.51E+00	1.28E−02	1.46E−26	**1.57E−32**	**1.57E−32**
std	4.14E−02	3.41E−01	3.88E−02	2.21E−26	6.25E−35	6.25E−35
RNFE	–	–	105,568 ± 50,594	84,062 ± 1604	**28,798** ± 552	31,018 ± 695
f_{13}						
Mean	1.31E+00	2.82E+01	6.07E−01	6.86E−26	**1.35E−32**	**1.35E−32**
std	3.32E−01	6.69E+00	1.42E+00	8.28E−26	8.28E−48	8.28E−48
RNFE	–	–	213,305 ± 43,033	90,758 ± 2169	**30,763** ± 753	33,021 ± 858

benchmark problems, and the advantage becomes more obvious with the increase of problem dimension. The basic BBO and B-BBO sometimes converge fast at the very early stage, but they are easy to be trapped by local optima because of poor exploration abilities. DE/BBO combines the DE's ability in exploration and BBO's ability in exploitation, and thus keeps a fast convergency speed longer than BBO and exploits more precise optima than DE. The EBO methods can achieve a much better balance between exploration and exploitation due to the two new migration operators, and they are more capable of jumping out of local optima by using the immaturity index η.

By comparing EBO1 and EBO2, we find that the former generally converges faster, while the latter achieves better results on more test functions. That is, the local random topology is expected to produce results slightly better than the ring topology, but the ring topology is easier to implement and consumes less computational time in neighbor selection. In addition, the random topology exhibits more performance advantage on higher dimensional problems. In general, we prefer to use the random topology in EBO for most unknown global optimization problems, and favor the use of the ring topology for urgent problems.

Table 7
Statistical test results on the 50-*D* problems.

f	EBO1 vs.					EBO2 vs.				
	BBO	B-BBO	DE	DE/BBO	EBO2	BBO	B-BBO	DE	DE/BBO	EBO1
f_1	†9.82E−56	†1.98E−72	1.06E−01	†1.55E−20	†0.00E+00	†9.82E−56	†1.98E−72	1.06E−01	†1.55E−20	1.00E+00
f_2	†7.42E−92	†4.34E−86	†8.20E−03	†3.22E−29	†3.76E−04	†7.42E−92	†4.34E−86	†8.20E−03	†3.22E−29	1.00E+00
f_3	†2.73E−54	†1.75E−75	7.93E−02	†9.40E−11	1.00E+00	†2.73E−54	†1.75E−75	7.93E−02	†9.40E−11	†0.00E+00
f_4	†1.50E−94	†1.36E−97	†1.02E−60	†1.86E−67	1.00E+00	†1.50E−94	†1.36E−97	†1.02E−60	†1.85E−67	†1.26E−06
f_5	†5.90E−26	†4.19E−64	†2.98E−10	2.12E−01	1.00E+00	†4.77E−26	†3.84E−64	†8.07E−11	†4.09E−02	†1.22E−18
f_6	†4.28E−22	†7.77E−63	†3.60E−04	–	–	†4.28E−22	†7.77E−63	†3.60E−04	–	–
f_7	†4.01E−13	†2.54E−17	†1.84E−09	†3.66E−09	9.93E−01	†8.49E−18	†4.94E−23	†5.17E−12	†2.37E−12	†7.30E−03
f_8	1.00E+00	1.00E+00	†9.03E−20	1.00E+00	9.99E−01	1.00E+00	1.00E+00	†1.92E−20	1.00E+00	†1.10E−03
f_9	2.38E−01	†2.62E−02	†2.68E−57	†5.73E−15	8.12E−01	2.36E−01	†1.00E−03	†1.63E−57	†3.69E−15	1.88E−01
f_{10}	†1.73E−72	†7.01E−79	†6.10E−03	†3.22E−16	8.17E−01	†1.73E−72	†7.01E−79	†6.10E−03	†5.16E−17	1.83E−01
f_{11}	†3.45E−116	†3.58E−190	†9.92E−05	1.60E−01	–	†3.45E−116	†3.58E−190	†9.92E−05	1.60E−01	–
f_{12}	†5.11E−71	†1.31E−63	†5.90E−03	†6.20E−07	–	†5.11E−71	†1.31E−63	†5.90E−03	†6.20E−07	–
f_{13}	†4.87E−58	†3.76E−61	†6.10E−04	†1.56E−09	–	†4.87E−58	†3.76E−61	†6.10E−04	†1.56E−09	–

The symbol "†" indicates that the EBO method has statistically significant improvement over the corresponding algorithms at 95% confidence level.

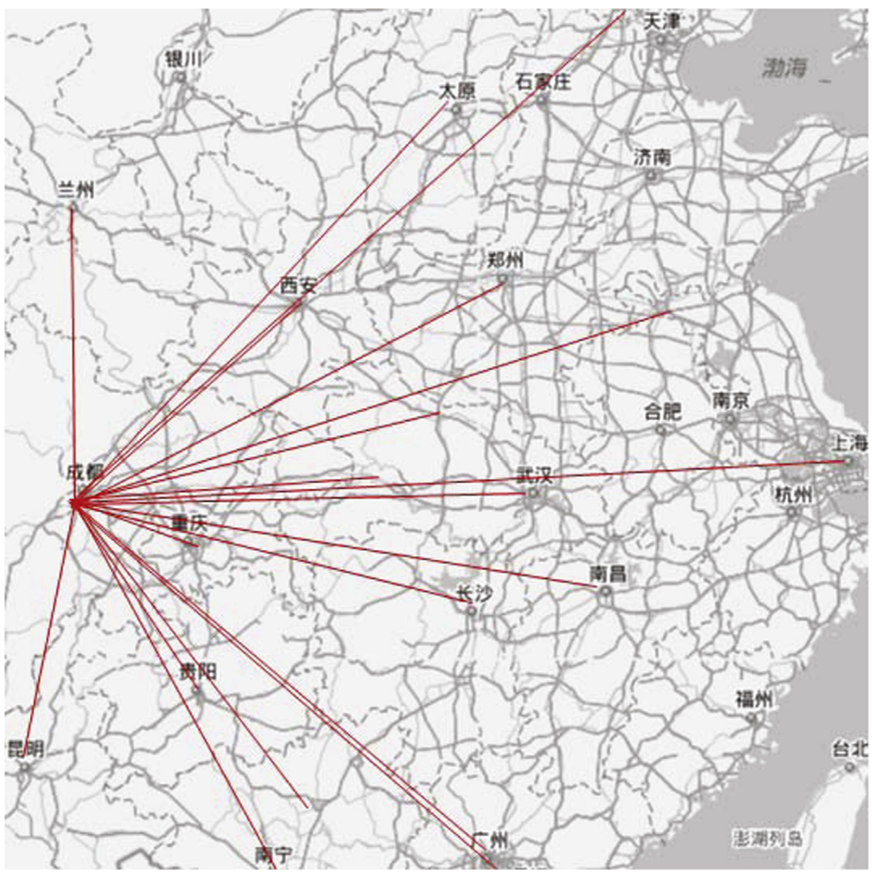

Fig. 5. The distribution of the air freight hubs.

5. Application to a real-world emergency airlift problem

In this section, we present an emergency airlift problem in the Ms 7.0 Ya'an–Lushan Earthquake, Sichuan Province, Southwest China, which has been successfully solved by the proposed EBO method.

The problem can be stated as follows. A certain amount of supplies is planned to be transported by air from a set of m (potential) air freight hubs to the airport closest to the disaster area. There are n types of supplies, each of which has a lower bound l_j and an upper bound u_j, and the amount of supply j available at hub i is s_{ij} $(1 \le i \le m, 1 \le j \le n)$. Within a given time period, hub i can arrange at most K_i flight batches, where batch k has a capacity c_{ik} and requires a preparation time τ_k $(1 \le k \le K_i)$. Let t_i be the travel time from hub i to the target, then the expected arrival time t_{ik} of batch k is $(t_i + \tau_k)$.

The problem is to determine the amounts x_{ijk} of supply j delivered in batch k of hub i, such that the supplies arrive as early as possible:

$$\min \quad f = \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{K_i} w_j x_{ijk} t_{ik} - \alpha \sum_{j=1}^{n} w_j \delta_j$$

$$\text{s.t.} \quad \sum_{i=1}^{m} \sum_{k=1}^{K_i} x_{ijk} \ge l_j, \quad j = 1, \ldots, n$$

$$\sum_{k=1}^{K_i} x_{ijk} \le s_{ij}, \quad i=1,\ldots,m;\ j=1,\ldots,n$$

$$\sum_{j=1}^{n} x_{ijk} \le c_{ik}, \quad i=1,\ldots,m;\ k=1,\ldots,K_i$$

$$x_{ijk} \in \mathbb{Z}^+, \quad i=1,\ldots,m; j=1,\ldots,n;\ k=1,\ldots,K_i \tag{5}$$

where w_j is the importance weight of supply j, α is an "award" coefficient, and δ_j is the amount of supply j over the lower bound:

$$\delta_j = \begin{cases} \sum_{i=1}^{m}\sum_{k=1}^{K_i} x_{ijk} - l_j & \text{if } \sum_{i=1}^{m}\sum_{k=1}^{K_i} x_{ijk} < u_j \\ u_j - l_j & \text{else} \end{cases} \tag{6}$$

It is not difficult to see that, if we have determined all the amounts x_{ij} of supply j provided by hub i, then in order to optimize the objective function, the supplies should be sent in decreasing order of their weights (until the capacities are exhausted), i.e., more important supplies should be arranged to earlier batches.

Table 9
The airlift solution of EBO implemented in the Ya'an–Lushan Earthquake.

Supply type	1	2	3	4	5	6	7	8	9
Hub 1	134	26	121	105	51	29	38	88	8
Hub 2	120	0	91	39	50	0	0	0	0
Hub 3	0	13	113	399	97	42	270	70	196
Hub 4	0	0	0	0	0	0	0	0	0
Hub 5	5	7	56	113	62	53	94	28	82
Hub 6	0	10	33	257	35	12	136	20	81
Hub 7	12	22	111	155	9	59	320	58	65
Hub 8	22	5	53	39	131	0	33	27	93
Hub 9	0	0	0	0	0	0	0	101	0
Hub 10	35	23	95	279	0	12	156	0	34
Hub 11	81	19	29	171	24	39	205	126	0
Hub 12	121	5	232	149	29	64	0	0	79
Hub 13	0	31	125	69	75	0	0	0	0
Hub 14	0	0	30	96	59	6	60	18	25
Hub 15	9	8	76	347	60	0	20	20	55
Hub 16	150	0	100	0	0	0	92	0	49
Hub 17	25	12	14	19	33	17	32	77	71
Hub 18	0	0	0	0	0	0	0	0	0
Hub 19	0	0	0	0	0	0	164	187	72

Table 8
The input parameters of the airlift problem in the Ya'an–Lushan Earthquake.

Supply type	1	2	3	4	5	6	7	8	9
Weight	1.6	1.3	1.2	1.0	0.96	0.87	0.65	0.48	0.35
Lower bound	660	160	1100	2000	700	330	1620	820	910
Upper bound	720	210	1280	2550	830	380	1860	950	1200
	Available supplies								
Hub 1	189	26	230	200	97	80	107	115	192
Hub 2	130	0	120	333	93	29	176	96	109
Hub 3	0	13	160	561	108	66	318	81	205
Hub 4	94	37	610	818	71	138	336	165	599
Hub 5	48	7	205	525	73	101	159	30	95
Hub 6	0	10	35	270	35	12	136	20	81
Hub 7	18	23	125	170	9	59	320	58	65
Hub 8	26	5	80	121	160	12	530	51	98
Hub 9	55	54	100	520	63	25	1055	130	80
Hub 10	39	24	120	291	0	46	318	106	160
Hub 11	83	21	50	274	24	39	205	126	0
Hub 12	124	5	246	196	50	92	350	0	109
Hub 13	0	31	125	135	92	41	51	0	72
Hub 14	0	0	30	96	59	6	60	18	25
Hub 15	10	8	90	356	76	40	270	62	88
Hub 16	151	0	120	155	100	37	184	0	50
Hub 17	127	12	210	290	91	43	540	79	129
Hub 18	107	68	350	1121	202	71	918	156	316
Hub 19	33	36	200	308	68	71	367	198	75
	Flight batches								
	c	τ	c	τ	c	τ	c	τ	
Hub 1	600	90	900	180					
Hub 2	300	60	300	180	600	270			
Hub 3	1200	60	300	150	300	240			
Hub 4	600	90	1200	120	600	180	850	285	
Hub 5	500	120	900	240					
Hub 6	300	90	300	210					
Hub 7	300	120	550	180					
Hub 8	250	90	500	180	325	225			
Hub 9	300	60	600	120	1200	240			
Hub 10	600	60	600	180					
Hub 11	300	75	550	180					
Hub 12	600	50	600	165					
Hub 13	300	120	300	300					
Hub 14	300	120							
Hub 15	500	90	500	195					
Hub 16	250	90	250	210	300	270			
Hub 17	300	90	600	165	800	300			
Hub 18	300	45	1200	90	800	150	1100	240	
Hub 19	800	105	550	210	300	315			

Table 10
The comparative results of the algorithms on the real-world emergency airlift problem (over 20 runs).

	5-min			15-min			30-min		
	min	max	mean (std)	min	max	Mean (std)	min	max	Mean (std)
EBO	1,711,721	1,716,373	**1,714,018** (1248)	1,711,616	1,715,702	**1,714,011** (1326)			
DE	1,726,477	1,752,178	1,736,778° (6381)	1,720,624	1,755,506	1,731,351° (9281)			
BBO	2,553,910	3,303,915	2,936,001° (208,540)	2,248,108	3,181,136	2,666,287°• (235,298)	2,313,378	3,262,338	2,621,279 (345,869)
DE/BBO	1,730,308	1,746,352	1,740,971° (7225)	1,722,972	1,746,352	1,733,969°• (7749)	1,724,639	1,747,231	1,732,502 (8584)

In columns 4 and 7, ° indicates that EBO have statistically significant improvement over the other algorithms, and in columns 7 and 10 • indicates that there is a statistically significant difference between the result obtained by running the same algorithm with 5 (15) min and that with 15 (30) min (at 99% confidence level).

Thus we transform the problem (5) into the following form:

$$\min \quad f=\sum_{i=1}^{m}\sum_{j=1}^{n} w_j x_{ij} t_{ij} - \alpha \sum_{j=1}^{n} w_j \delta_j + M \sum_{j=1}^{n} P(j)$$
$$\text{s.t.} \quad x_{ij} \le s_{ij}, \quad i=1,\ldots,m;\ j=1,\ldots,n$$
$$x_{ij} \in \mathbb{Z}^+, \quad i=1,\ldots,m;\ j=1,\ldots,n \tag{7}$$

where t_{ij} is the expected arrival time of supply j from hub i (if x_{ij} is divided into multiple batches, then an arithmetic mean time is used), M is a large positive constant, and $P(j)$ is the penalty function for handling the constraint on the lower bound of supply j:

$$P(j)=\begin{cases} l_j-\sum_{i=1}^{m} x_{ij} & \text{if } \sum_{i=1}^{m} x_{ij} < l_j \\ 0 & \text{else} \end{cases} \tag{8}$$

The earthquake occurred at 08:02 Beijing Time (00:02 UTC) on April 20, 2013. At about 10:30, the disaster relief command center (DRCC) began to plan the airlift task which involved 9 types of supplies and 19 air freight hubs (the distribution of which is shown in Fig. 5). At 11:35, the DRCC obtained the data from the disaster area and the air freight hubs, as summarized in Table 8. As we can see, there were 12 items of $s_{ij}=0$, and thus the problem had 159 dimensions. α and M were respectively set to 305 and 10,000 for the problem. It was required to work out the airlift solution before 11:55. Up to 11:48, we prepared the computational environment and initialized the problem and algorithm parameters. Thus we set the maximum running time of the algorithms to 5 min.

We simultaneously run three instances of the EBO algorithm and two instances of the DE algorithm on five computers with the same configuration (Intel Core i5-2430M processor and 4 GB DDR3 memory) for solving the given problem. Note that the algorithms were adapted for the integer programming problem by rounding the components of every new generated solution to the nearest integers, which would not affect significantly the algorithm performance [28]. The ring topology was employed in EBO for saving computational resources.

At 11:54, the algorithms produced five solutions, the objective values of which were 1,712,508 (EBO), 1,713,609 (EBO), 1,714,799 (EBO), 1,733,643 (DE), and 1,743,436 (DE). We submitted the solution with the best objective value, which was then accepted and put into implementation. Table 9 gives the detailed information about the solution. In general, the decision-maker was very satisfied with the implementation results of the solution produced by EBO.

Afterwards, we conducted a more comprehensive experiments to validate EBO, DE, BBO, and DE/BBO on the given problem (the results of which are given in Table 10):

1. We first run each algorithm with a maximum running time of 5 min. The results (averaged over 20 runs) show that EBO has the best performance among the four algorithms.
2. We then run each algorithm with a maximum running time of 15 min. The results (averaged over 20 runs) show that EBO still performs much better than the others, but EBO and DE cannot further improve the solutions of that of 5 min (in terms of statistical significance).
3. We further run BBO and DE/BBO with a maximum running time of 30 min. The results (averaged over 20 runs) show that the two algorithms also fail to improve the solutions of that of 15 min.

In summary, we can believe that EBO has found high quality solutions in 5 min, and it is most suitable for solving the emergency problem. In fact, during the experiments none of the other algorithms can find a solution better than the EBO's solution submitted to the DRCC.

6. Conclusions

BBO is a biogeography-inspired metaheuristic method that has received much attention in recent years. The basic BBO uses a global topology of population and a migration operator which has a good ability of exploitation, but it is not very efficient in exploration and often suffers from premature convergence. In this paper, we propose a new variation of BBO, named EBO, which employs a local topology to distinguish neighbors and non-neighbors of each island, and defines two new migration operators to enrich information sharing between the islands, and thus improves the exploration ability without harming the exploitation ability of BBO. Computational experiments demonstrate that the EBO is a very competitive method for global optimization. A real-world application has also validated the EBO on an emergency operational problem.

We believe that the main idea of EBO, i.e., integrating a global migration operator and a local migration operator to balance exploration and exploitation, can be applied to a variety of other problems, including many complex combinatorial optimization problems. We are currently developing discrete EBO for permutation-based optimization problems such as the traveling salesman and the flow-shop scheduling. Our ongoing work also includes extending the EBO for multiobjective optimization, and studying the parallelization of EBO, which can be much easier to conduct on local topologies than on a global one.

Acknowledgment

This work was supported by National Natural Science Foundation of China (Grant nos. 61020106009, 61105073 and 61272075).

References

[1] McCulloch WS, Pitts WH. A logical calculus for the ideas immanent in nervous activity. Bull Math Biophys 1943;5:115–33.
[2] Von Neumann J, Burks AW, et al. Theory of self-reproducing automata. Urbana: University of Illinois Press; 1966.

[3] Adleman L. Molecular computation of solutions to combinatorial problems. Science 1994;266:1021–4.
[4] De Jong KA. Evolutionary computation: a unified approach. Cambridge: MIT press; 2006.
[5] Simon D. Biogeography-based optimization. IEEE Trans Evol Comput 2008;12:702–13.
[6] Bhattacharya A, Chattopadhyay P. Biogeography-based optimization for different economic load dispatch problems. IEEE Trans Power Syst 2010;25:1064–77.
[7] Singh U, Kumar H, Kamal T. Design of Yagi-Uda antenna using biogeography based optimization. IEEE Trans Antennas Propag 2010;58:3375–9.
[8] Boussaïd I, Chatterjee A, Siarry P, Ahmed-Nacer M. Biogeography-based optimization for constrained optimization problems. Comput Oper Res 2012;39:3293–304.
[9] Zheng Y-J, Ling H-F, Shi H-H, Chen H-S, Chen S-Y. Emergency railway wagon scheduling by hybrid biogeography-based optimization. Comput Oper Res 2014;43:1–8.
[10] Ergezer M, Simon D, Du D. Oppositional biogeography-based optimization. In: IEEE international conference on systems, man and cybernetics; 2009. p. 1009–14. http://dx.doi.org/10.1109/ICSMC.2009.5346043.
[11] Ma H, Simon D. Blended biogeography-based optimization for constrained optimization. Eng Appl Artif Intell 2011;24:517–25.
[12] Li X, Yin M. Multi-operator based biogeography based optimization with mutation for global numerical optimization. Comput Math Appl 2012;64:2833–44.
[13] Lohokare M, Pattnaik S, Panigrahi B, Das S. Accelerated biogeography-based optimization with neighborhood search for optimization. Appl Soft Comput 2013;13:2318–42.
[14] Zheng Y-J, Ling H-F, Wu X-B, Xue J-Y. Localized biogeography-based optimization. Soft Comput 2014.
[15] Du D, Simon D, Ergezer M. Biogeography-based optimization combined with evolutionary strategy and immigration refusal. In: IEEE international conference on systems, man and cybernetics; 2009. p. 997–1002. http://dx.doi.org/10.1109/ICSMC.2009.5346055.
[16] Gong W, Cai Z, Ling CX. DE/BBO: a hybrid differential evolution with biogeography-based optimization for global numerical optimization. Soft Comput 2010;15:645–65.
[17] Boussaïd I, Chatterjee A, Siarry P, Ahmed-Nacer M. Two-stage update biogeography-based optimization using differential evolution algorithm (DBBO). Comput Oper Res 2011;38:1188–98.
[18] Ma H, Fei M, Ding Z, Jin J. Biogeography-based optimization with ensemble of migration models for global numerical optimization. In: IEEE congress on evolutionary computation; 2012. p. 1–8. http://dx.doi.org/10.1109/CEC.2012.6252930.
[19] MacArthur R, Wilson E. The theory of biogeography. NJ: Princeton University Press; 1967.
[20] Ma H. An analysis of the equilibrium of migration models for biogeography-based optimization. Inf Sci 2010;180:3444–64.
[21] Simon D. Biogeography-based optimization; 2009. URL: http://academic.csuohio.edu/simond/bbo/.
[22] Morrone JJ, Crisci JV. Historical biogeography: introduction to methods. Annu Rev Ecol Syst 1995;26:373–401.
[23] Rieseberg L, Willis J. Plant speciation. Science 2007;317:910–4.
[24] McKenna M. Sweepstakes, filters, corridors, noahs arks, and beached viking funeral ships in palaeogeography. In: Tarling D, Runcorn S, editors. Implications of continental drift to the earth sciences. London: Academic Press; 1973. p. 295–308.
[25] Storn R, Price K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 1997;11:341–59.
[26] Yao X, Liu Y, Lin G. Evolutionary programming made faster. IEEE Trans Evol Comput 1999;3:82–102.
[27] Clerc M. Standard PSO 2007; 2010. URL: http://www.particleswarm.info.
[28] Laskari EC, Parsopoulos KE, Vrahatis MN. Particle swarm optimization for integer programming. In: IEEE congress on evolutionary computation, vol. 2. IEEE; 2002. p. 1582–7. http://dx.doi.org/10.1109/CEC.2002.1004478.