



## Blended biogeography-based optimization for constrained optimization

Haiping Ma<sup>a,\*</sup>, Dan Simon<sup>b</sup>

<sup>a</sup> Department of Electrical Engineering, Shaoxing University, Shaoxing, Zhejiang, China

<sup>b</sup> Department of Electrical and Computer Engineering, Cleveland State University, Cleveland, Ohio, USA

### ARTICLE INFO

#### Article history:

Received 4 May 2010

Received in revised form

4 August 2010

Accepted 6 August 2010

Available online 16 September 2010

#### Keywords:

Evolutionary algorithm

Biogeography-based optimization

Constrained optimization

### ABSTRACT

Biogeography-based optimization (BBO) is a new evolutionary optimization method that is based on the science of biogeography. We propose two extensions to BBO. First, we propose a blended migration operator. Benchmark results show that blended BBO outperforms standard BBO. Second, we employ blended BBO to solve constrained optimization problems. Constraints are handled by modifying the BBO immigration and emigration procedures. The approach that we use does not require any additional tuning parameters beyond those that are required for unconstrained problems. The constrained blended BBO algorithm is compared with solutions based on a standard genetic algorithm (SGA) and standard particle swarm optimization 2007 (SPSO 07). The numerical results demonstrate that constrained blended BBO outperforms SGA and performs similarly to SPSO 07 for constrained single-objective optimization problems.

© 2010 Elsevier Ltd. All rights reserved.

### 1. Introduction

Many optimization problems in science and engineering have constraints (Schwefel, 1995; Runarsson and Yao, 2000) which can be linear or nonlinear. Evolutionary algorithms (EAs) have been successful for a wide range of constrained optimization problems (Coello, 2002; Zbigniew and Marc, 1996). In the last few years, for example, evolutionary strategy (ES) (McTavish and Restrepo, 2008; Mezura-Montes and Coello, 2005), genetic algorithms (GAs) (Deb, 2000; Eshelman and Schaffer, 1993; Yao et al., 1999), differential evolution (DE) (Huang et al., 2006; Mezura-Montes and Palomeque-Ortiz, 2009; Zielinske and Laur, 2006), and particle swarm optimization (PSO) (Clerc and Kennedy, 2002; Eberhart and Shi, 2004), have arisen as attractive optimization algorithms due to their competitive results.

Biogeography-based optimization (BBO) (Simon, 2008) is a new evolutionary algorithm for global optimization that was introduced in 2008. It is modeled after the immigration and emigration of species between habitats. The application of this idea to optimization allows information sharing between candidate solutions. One distinctive feature of BBO is that the original population is not discarded after each generation. It is rather modified by migration. Another distinctive feature is that, for each generation, BBO uses the fitness of each solution to determine its immigration and emigration rate. BBO has demonstrated good performance on various unconstrained benchmark functions

(Du et al., 2009; Ma et al., 2009; Simon, 2008). It has also been applied to real-world optimization problems, including sensor selection (Simon, 2008), power system optimization (Rarick et al., 2009), groundwater detection (Kundra et al., 2009), and satellite image classification (Panchal et al., 2009). See Ref. Gardner and Simon (2009) for a web-based BBO graphical user interface.

This paper proposes two extensions to BBO. First, we take the blended crossover operator of the GA (Mühlenbein and Schlierkamp-Voosen, 1993) and use it to derive a blended migration operator for BBO. We show that blended BBO generally outperforms standard BBO on a set of benchmark problems. Second, we generalize BBO to constrained optimization problems. BBO has been used for unconstrained optimization, but up to now there have not been any reports in the literature of its use for constrained optimization. In this paper blended BBO is used for the optimization of constrained benchmark problems.

The most common approach to deal with constraints is the use of penalty functions. However, penalty functions have several limitations and problems which are difficult to deal with (Smith and Coit, 1997; Yeniyay, 2005), including the difficulty of tuning the penalty parameters. We use a recently-developed method for constrained optimization, based on the superiority of feasible solutions over infeasible solutions, proposed by Deb (2000). Its performance has been demonstrated in Zielinske and Laur (2006) which deals with constraints in differential evolution. In this paper we handle the constraints by modifying the BBO migration procedure between solutions based on their feasibility.

The remainder of this paper is organized as follows. Section 2 reviews the BBO algorithm and proposes blended BBO. Section 3 describes the constraint-handling method for single-objective

\* Corresponding author. Tel.: +86 575 88345673.

E-mail addresses: Mahp@usx.edu.cn (H. Ma), d.j.simon@csuohio.edu (D. Simon).

optimization problems and shows how it can be applied to BBO. Simulation results and performance analysis of the blended BBO algorithms are presented in Section 4, and conclusions and directions for future investigations are given in Section 5. A preliminary version of this paper was presented in (Ma and Simon, 2010).

## 2. Blended biogeography-based optimization (B-BBO)

BBO is a new population-based global optimization algorithm (Simon, 2008). As its name implies, BBO is based on the study of the distribution of species over time and space. This study, which is a subset of biology, is called biogeography (Lomolino et al., 2009; Whittaker, 1998). Suppose that we have a global optimization problem and a population of candidate solutions (individuals). Each individual is considered to be analogous to a habitat and is characterized by a habitat suitability index (HSI). The value of HSI, which is the same as fitness in other population-based optimization algorithms, and which measures the goodness of the solution, depends on many features of the habitat. A habitat with a high HSI is a good solution, and a habitat with a low HSI is a poor solution. High-HSI solutions tend to share their features with low-HSI solutions by emigrating solution features to other habitats. Low-HSI solutions accept a lot of new features from high-HSI solutions by immigration from other habitats. Immigration and emigration tend to improve the solutions and thus evolve a solution to the optimization problem. Namely, the BBO algorithm views the value of HSI as the objective function, and the evolutionary procedure of BBO is to determine the solutions which maximize the HSI by immigrating and emigrating features of the habitats. In BBO, there are two main operators: migration (which includes both emigration and immigration) and mutation. Several options can be used for migration and mutation, but one option for implementing migration is described in Section 2.1. We introduce a blended migration mechanism to BBO, which is called blended BBO. One option for implementing the mutation operator is described in Section 2.2.

### 2.1. Blended migration

In biogeography, migration is the movement of species between different habitats. In BBO, migration is a probabilistic operator that adjusts each solution  $H_i$  by sharing features between solutions. In the original BBO paper (Simon, 2008), the probability that the solution  $H_i$  is selected as the immigrating habitat is proportional to its immigration rate  $\lambda_i$ , and the probability that the solution  $H_j$  is selected as the emigrating habitat is proportional to the emigration rate  $\mu_j$ . Migration can be expressed as

$$H_i(\text{SIV}) \leftarrow H_j(\text{SIV}). \quad (1)$$

In biogeography, an SIV is a suitability index variable which characterizes the habitability of a habitat (Simon, 2008); that is, the HSI is determined by many SIVs. In BBO, an SIV is a solution feature, equivalent to a gene in a GA. In other words, an SIV is a search variable and the set of all possible SIVs is the search space from which an optimal solution will be determined. Eq. (1) means that a solution feature of solution  $H_i$  is replaced by a feature from solution  $H_j$ .

In BBO, each  $H_i$  has its own immigration rate  $\lambda_i$  and emigration rate  $\mu_i$ . A good solution has relatively high  $\mu$  and low  $\lambda$ , while the converse is true for a poor solution. The immigration rate and the emigration rate are functions of the fitness of the solution. They can be calculated as

$$\lambda_i = I \left( 1 - \frac{k(i)}{n} \right)$$

$$\mu_i = E \left( \frac{k(i)}{n} \right), \quad (2)$$

where  $I$  is the maximum possible immigration rate;  $E$  is the maximum possible emigration rate;  $k(i)$  is the fitness rank of the  $i$ th individual (1 is worst and  $n$  is best); and  $n$  is the number of candidate solutions in the population.  $I$  and  $E$  are often set equal to 1, or slightly less than 1. Note that this migration function is a linear curve, but in general it might be a more complicated curve.

We propose a new migration operator called blended migration, which is a generalization of the standard BBO migration operator, and which is motivated by blended crossover in GAs. Blended crossover is frequently used in GAs (Mühlenbein and Schlierkamp-Voosen, 1993). In blended crossover, instead of copying a parent's gene to a child chromosome, the offspring are obtained by combining parents' genes. In blended migration in BBO, a solution feature of solution  $H_i$  is not simply replaced by a feature from solution  $H_j$ . Instead, a new solution feature in a BBO solution is comprised of two components: the migration of a feature from another solution, and the migration of a feature from itself. Blended migration is defined as

$$H_i(\text{SIV}) \leftarrow \alpha H_i(\text{SIV}) + (1 - \alpha) H_j(\text{SIV}), \quad (3)$$

where  $\alpha$  is a real number between 0 and 1. It could be random, or deterministic, or it could be proportional to the relative fitness of the solutions  $H_i$  and  $H_j$ . Eq. (3) means that the new solution feature (SIV) of  $H_i$  comes from a combination of its own SIV and the emigrating solution's SIV.

The core idea of the proposed blended migration operator is based on two considerations. First, the operator is easily used with continuous-domain optimization problems. Second, blended combination operators have been widely and successfully used in other population-based optimization algorithms. Blended migration is an attractive BBO modification from a couple of different viewpoints. On the one hand, good solutions will be less likely to be degraded due to migration. On the other hand, poor solutions can still accept a lot of new features from good solutions. That is, if solution  $H_i$  is much more fit than solution  $H_j$ , it would make sense to have  $\alpha$  close to 1; but if solution  $H_i$  is much less fit than the solution  $H_j$ , it would make sense to have  $\alpha$  close to 0. Blended migration is similar to the blended crossover approach of the breeder GA (Mühlenbein and Schlierkamp-Voosen, 1993) and ES (McTavish and Restrepo, 2008) in which both of the parents can contribute characteristics to a single feature of an offspring.

### 2.2. Mutation

Mutation is a probabilistic operator that randomly modifies a solution's SIV based on its a priori probability of existence. Namely, a randomly generated SIV replaces a selected SIV in the solution  $H_i$  according to a mutation probability. Although mutation is not the most important factor in BBO, the improvement of solutions is obtained by perturbing the solution after the migration operation. For classic BBO, the mutation probability is inversely proportional to the solution probability (Simon, 2008), and is defined by

$$m_i = m_{\max} \left( 1 - \frac{P_i}{P_{\max}} \right) \quad (4)$$

where  $m_{\max}$  is the user-defined maximum mutation probability,  $P_{\max} = \text{argmax} P_i$ ,  $i = 1, \dots, n$  ( $n$  is population size), and  $P_i$  is the solution probability. For more details see Simon (2008). This mutation scheme tends to increase diversity among the population.

---

```

Create a random set of habitats (population)  $H_1, H_2, \dots, H_n$ ;
Compute HSI values;
While the halting criterion is not satisfied do
  Compute immigration rate  $\lambda$  and emigration rate  $\mu$  for each habitat based on HSI;
  For each habitat (solution)
    For each SIV (solution feature)
      Select habitat  $H_i$  with probability  $\propto \lambda_i$ ;
      If  $H_i$  is selected then
        Select  $H_j$  with probability  $\propto \mu_j$ ;
        If  $H_j$  is selected then
           $H_i(\text{SIV}) \leftarrow \alpha H_i(\text{SIV}) + (1-\alpha)H_j(\text{SIV})$ ;
        End if
      End if
      Select  $H_i(\text{SIV})$  based on mutation probability  $m_i$ ;
      If  $H_i(\text{SIV})$  is selected then
        Replace  $H_i(\text{SIV})$  with a randomly generated SIV;
      End if
    Next for
  Recompute HSI values;
Next for
End while

```

---

**Fig. 1.** BBO algorithm, where  $\alpha=0$  for standard BBO, and  $\alpha > 0$  for blended BBO.

The BBO algorithm, generalized for blended migration, is shown in Fig. 1. If  $\alpha=0$  in Fig. 1, then we have the standard BBO algorithm. If  $\alpha$  is random, or is proportional to relative solution fitnesses, then we have blended BBO.

### 3. Constrained optimization

Constrained optimization problems frequently arise in every field of science, engineering, and industry. Without loss of generality, the single-objective constrained optimization problem can be formalized as follows:

find  $x$  to minimize  $f(x)$   
 such that  $g_i(x) \leq 0$  for  $i = 1, \dots, q$   
 and  $h_i(x) = 0$  for  $i = q+1, \dots, m$

where  $x=(x_1, x_2, \dots, x_n)$  is the vector of decision variables and each  $x_i$  is bounded by lower and upper limits ( $L_i \leq x_i \leq U_i$ ). The set of all decision variables which satisfy the constraints of the problem is called the feasible region.  $q$  is the number of inequality constraints and  $m-q$  is the number of equality constraints. Note that constraints could be linear or nonlinear. The objective of an optimization algorithm is to minimize the cost function  $f(x)$ , while at the same time satisfying the constraints  $g_i(x)$  and  $h_i(x)$ . Among the various constraint-handling methods, the use of penalty functions is the most common technique (Smith and Coit, 1997). In this paper we incorporate into BBO a method for constraint-handling that is based on feasibility rules (Deb, 2000), which have demonstrated promising performance in dealing with constraints. We incorporate this constraint-handling technique into our blended BBO algorithm as described below.

During the migration procedure, each candidate solution has been modified. We then check each solution to see if it is better than the solution of its original version (i.e., its version before migration). The modified solution will enter the population of the next generation only if it is better than its previous version, otherwise the previous solution will enter the population of the

next generation. Note that this is similar to a  $(\mu+\lambda)$  ES where the next generation comes from both parents and children.

For constrained problems, when a candidate solution  $H_1$  is compared to a candidate solution  $H_2$ , solution  $H_1$  is considered better if:

- 1) both solutions are feasible, but  $H_1$  has a cost that is less than or equal to that of  $H_2$ ; or,
- 2)  $H_1$  is feasible and  $H_2$  is not; or,
- 3) both solutions are infeasible, but  $H_1$  has a smaller overall constraint violation.

Note that no additional tuning parameters beyond those that are required for unconstrained problems are needed for this constraint-handling technique. For unconstrained problems, this method is identical to the original unconstrained BBO algorithm.

### 4. Simulation results

In this section we look at the performance of blended BBO based on the constraint-handling approach discussed above. A representative set of benchmark functions with constraints, including linear, nonlinear, and polynomial constraints, have been used for performance verification of the proposed approach. These functions are briefly summarized in Table 1. A more detailed description of these functions can be found in the literature Liang et al. (2005).

#### 4.1. Blended BBO performance

In our first experiment we compare the original BBO with the blended BBO (which we call B-BBO) to solve the constrained benchmark functions. The aim of this experiment is to investigate how BBO solves constrained problems and how blended migration affects BBO optimization performance for constrained

**Table 1**

Benchmark problems.  $L_I$  and  $N_I$  are the number of linear and nonlinear inequality constraints,  $L_E$  and  $N_E$  are the number of linear and nonlinear equality constraints,  $D$  is the number of dimensions, and  $A$  is the number of active constraints at the solution.

Function	$D$	Function type	$L_I$	$N_I$	$L_E$	$N_E$	$A$
g01	13	Quadratic	9	0	0	0	6
g02	20	Nonlinear	0	2	0	0	1
g03	10	Polynomial	0	0	0	1	1
g04	5	Quadratic	0	6	0	0	2
g05	4	Cubic	2	0	0	3	3
g06	2	Cubic	0	2	0	0	2
g07	10	Quadratic	3	5	0	0	6
g08	2	Nonlinear	0	2	0	0	0
g09	7	Polynomial	0	4	0	0	2
g10	8	Linear	3	3	0	0	6
g11	2	Quadratic	0	0	0	1	1
g12	3	Quadratic	0	1	0	0	0
g13	5	Nonlinear	0	0	0	3	3
g14	10	Nonlinear	0	0	3	0	3
g15	3	Quadratic	0	0	1	1	2
g16	5	Nonlinear	4	34	0	0	4
g17	6	Nonlinear	0	0	0	4	4
g18	9	Quadratic	0	13	0	0	6
g19	15	Nonlinear	0	5	0	0	0
g20	24	Linear	0	6	2	12	16
g21	7	Linear	0	1	0	5	6
g22	22	Linear	0	1	8	11	19
g23	9	Linear	0	2	3	1	6
g24	2	Linear	0	2	0	0	2

problems. For both BBO methods, the following parameters have to be examined: number of habitats (population size), maximum migration rates, mutation rate, and migration curve shapes. In Ma (2010) these parameters have been discussed in detailed, and we have obtained a reasonable set of tuning parameters. So we have not made any effort in finding the best parameter settings in this paper. For this experiment, the parameters used in the two methods are the same, which are recommended as follows: population size of 50, maximum immigration rate and maximum emigration rate of 1, and maximum mutation rate of 0.01. In addition, we use linear migration curves as suggested in Simon (2008). For maximum number of fitness function evaluations (Max\_NFFEs), we use 50,000.

Moreover, in our experiments, each function is optimized over 25 independent runs. We also use the same set of 25 initial random populations to evaluate the algorithms, and we set the parameter  $\alpha=0, 0.5, \text{ or } 0.8$  in Eq. (3) to investigate its influence on B-BBO performance (when  $\alpha=0$ , B-BBO reduces to standard BBO). In addition, we mention that in the original BBO paper a discrete version of BBO was used to minimize benchmark functions. In this paper we adopt a continuous version of BBO. Beside the best, mean, and worst value of benchmark functions, two performance criteria are selected from the literature (Liang et al., 2005) to evaluate the performance of the algorithms.

- 1) Number of feasible runs (NF): The number of feasible runs is the number of runs during which at least one feasible solution is found before the Max\_NFFEs condition terminates the run.
- 2) Number of successful runs (NS): The number of successful runs is the number of runs during which the algorithm finds a feasible solution  $x$  satisfying  $f(x) - f(x^*) \leq 0.0001$  before the Max\_NFFEs condition terminates the run.

In addition, in order to conclude that one method is “better” than another, we consider all three performance criteria.

- 1) If the algorithm has a higher NS than another, then it is better.
- 2) If two algorithms have the same NS, then the one with the higher NF is better.
- 3) If two algorithms have the same NS and NF, then the one with mean objective function value closer to the best known solution is better.

Tables 2 and 3 summarize the B-BBO performance on the constrained benchmark functions. We observe that, for most of the functions, B-BBO with three different values of  $\alpha$  gives good performance except function g20, g21, and g22, for which no feasible solution at all could be found. For 9 (g01, g02, g03, g05, g06, g08, g09, g16, g17) of the 24 functions, B-BBO ( $\alpha=0.5$ ) has a higher NS than BBO ( $\alpha=0$ ) and B-BBO ( $\alpha=0.8$ ). For 3 functions (g10, g15, g23), the three algorithms have the same NS, but B-BBO ( $\alpha=0.5$ ) has the higher NF. For 3 functions (g11, g18, g24), they all have the same NS and NF, but B-BBO ( $\alpha=0.5$ ) has the mean values closer to the best known solutions.

In summary, B-BBO ( $\alpha=0.5$ ) performs the best on 15 functions, B-BBO ( $\alpha=0.8$ ) performs the best on 3 functions (g07, g12, g19), and BBO ( $\alpha=0$ ) performs the best for 1 function (g13). For function g14, all three algorithms attain the global optimum every run, and for function g04, both B-BBO ( $\alpha=0.5$ ) and B-BBO ( $\alpha=0.8$ ) attain the global optimum every run.

From Table 3 we also obtain statistically significant differences of two pairs of BBOs (BBO ( $\alpha=0$ ) vs. B-BBO ( $\alpha=0.5$ ), and BBO ( $\alpha=0$ ) vs. B-BBO ( $\alpha=0.8$ )) based on the  $p$  value, which is the probability that the two sets of data come from the same distribution. From  $p$  value comparison between BBO ( $\alpha=0$ ) vs. B-BBO ( $\alpha=0.5$ ), and BBO ( $\alpha=0$ ) vs. B-BBO ( $\alpha=0.8$ ), we find that there are fourteen  $p$  values smaller than 0.05 (which is often used as the significance level or critical  $p$  value) for BBO ( $\alpha=0$ ) vs. B-BBO ( $\alpha=0.5$ ), and there are thirteen  $p$  values smaller than 0.05 for BBO ( $\alpha=0$ ) vs. B-BBO ( $\alpha=0.8$ ). Based on this result, the probability that the results of BBO ( $\alpha=0$ ), B-BBO ( $\alpha=0.5$ ), and B-BBO ( $\alpha=0.8$ ) are from the same distribution is low. It indicates that the value of  $\alpha$  is influential on BBO performance. B-BBO with  $\alpha > 0$  generally outperforms BBO ( $\alpha=0$ ), which means that blended migration can significantly improve the optimization ability of BBO. Also, B-BBO ( $\alpha=0.5$ ) is better than B-BBO ( $\alpha=0.8$ ) which indicates that when a new solution feature is contributed equally from itself and the selected emigrating solution, B-BBO performs best. Lastly, we note that the successfully optimized benchmark functions include quadratic, cubic, nonlinear, polynomial, and linear objective functions, thus the type of objective function is not of importance for successful optimization for B-BBO.

#### 4.2. Comparison between BBO and other EAs

The next experiment compares BBO against other popular evolutionary algorithms. We choose to compare the proposed B-BBO with stud GA (which we call SGA) (Khatib and Fleming, 1998) and standard PSO 2007. We compare with an SGA because the SGA is an improvement of the classic GA which uses the best individual at each generation for crossover. We compare with PSO because it often offers good performance and is itself a relatively new evolutionary algorithm. We use the current standard PSO (SPSO 07) (Bratton and Kennedy, 2007), obtainable from Particle Swarm Central (Particle Swarm Central). Here, the proposed constraint-handling method is adopted in an identical way, as described in Section 3, for B-BBO, SGA, and SPSO 07.

The parameters used in B-BBO are the same as those described in the previous section, and  $\alpha=0.5$ . For the SGA we use real coding, roulette wheel selection, single point crossover with a crossover probability of 1, and random mutation with a mutation



**Table 2**

Constrained optimization results obtained by BBO ( $\alpha=0$ ), B-BBO ( $\alpha=0.5$ ), and B-BBO ( $\alpha=0.8$ ) on 24 constrained functions over 25 independent runs. The table shows the best, mean, and worst values. Best results for each benchmark function are shown in boldface.

Fun.	Best known solution	BBO ( $\alpha=0$ )			B-BBO ( $\alpha=0.5$ )			B-BBO ( $\alpha=0.8$ )		
		Best	Mean	Worst	Best	Mean	Worst	Best	Mean	Worst
g01	-15.000	<b>-15.000</b>	-14.982	-14.791	<b>-15.000</b>	<b>-14.998</b>	<b>-14.991</b>	<b>-15.000</b>	-14.987	-14.842
g02	-0.803619	<b>-0.803619</b>	-0.792461	-0.783462	<b>-0.803619</b>	<b>-0.802154</b>	<b>-0.801924</b>	<b>-0.803619</b>	-0.798445	-0.781189
g03	-1.0005	<b>-1.0005</b>	-1.0000	-0.9992	<b>-1.0005</b>	<b>-1.0004</b>	<b>-1.0001</b>	<b>-1.0005</b>	-1.0000	-0.9997
g04	-30665.54	<b>-30665.54</b>	-30665.44	-30665.17	<b>-30665.54</b>	<b>-30665.54</b>	<b>-30665.54</b>	<b>-30665.54</b>	<b>-30665.54</b>	<b>-30665.54</b>
g05	5126.497	<b>5126.497</b>	5127.048	5129.993	<b>5126.497</b>	<b>5127.046</b>	<b>5129.913</b>	<b>5126.497</b>	5127.053	5129.391
g06	-6961.814	<b>-6961.814</b>	-6957.078	-6942.652	<b>-6961.814</b>	<b>-6961.005</b>	<b>-6960.874</b>	<b>-6961.814</b>	-6959.034	-6957.391
g07	24.306	26.272	30.547	34.119	<b>24.306</b>	27.051	35.574	<b>24.306</b>	<b>26.011</b>	<b>28.562</b>
g08	-0.095825	<b>-0.095825</b>	-0.094256	-0.090063	<b>-0.095825</b>	<b>-0.095825</b>	<b>-0.095825</b>	<b>-0.095825</b>	-0.090836	-0.089174
g09	680.630	<b>680.630</b>	688.637	699.758	<b>680.630</b>	<b>680.677</b>	<b>681.328</b>	<b>680.630</b>	685.647	698.027
g10	7049.248	<b>7049.248</b>	7236.683	7377.902	<b>7049.248</b>	<b>7057.369</b>	<b>7139.645</b>	<b>7049.248</b>	7084.763	7147.586
g11	0.74990	<b>0.74990</b>	0.74992	0.74994	<b>0.74990</b>	<b>0.74990</b>	<b>0.74990</b>	<b>0.74990</b>	0.74993	0.74997
g12	-1.00000	<b>-1.00000</b>	-0.99514	-0.98725	<b>-1.000</b>	-0.99994	-0.99991	<b>-1.00000</b>	<b>-1.00000</b>	<b>-1.00000</b>
g13	0.053942	<b>0.053942</b>	<b>0.054264</b>	<b>0.054931</b>	<b>0.053942</b>	0.055046	0.056891	<b>0.053942</b>	0.054802	0.055284
g14	-47.765	<b>-47.765</b>	<b>-47.765</b>	<b>-47.765</b>	<b>-47.765</b>	<b>-47.765</b>	<b>-47.765</b>	<b>-47.765</b>	<b>-47.765</b>	<b>-47.765</b>
g15	961.715	<b>961.715</b>	968.648	972.114	<b>961.715</b>	<b>962.627</b>	<b>965.736</b>	<b>961.715</b>	963.683	967.728
g16	-1.90516	<b>-1.90516</b>	-1.90264	-1.90017	<b>-1.90516</b>	<b>-1.90511</b>	<b>-1.90508</b>	<b>-1.90516</b>	-1.90443	-1.90382
g17	8853.540	<b>8853.540</b>	8854.231	8855.703	<b>8853.540</b>	<b>8853.600</b>	<b>8853.692</b>	<b>8853.540</b>	8854.867	8855.376
g18	-0.866025	<b>-0.866025</b>	-0.862067	-0.857428	<b>-0.866025</b>	<b>-0.865521</b>	<b>-0.863036</b>	<b>-0.866025</b>	-0.863358	-0.859545
g19	32.656	<b>32.656</b>	32.661	32.678	<b>32.656</b>	32.659	32.665	<b>32.656</b>	<b>32.656</b>	<b>32.656</b>
g20	0.204979	-	-	-	-	-	-	-	-	-
g21	193.725	-	-	-	-	-	-	-	-	-
g22	236.431	-	-	-	-	-	-	-	-	-
g23	-400.055	-389.748	-381.794	-362.830	<b>-396.034</b>	<b>-391.319</b>	<b>-384.183</b>	-391.653	-378.407	-348.958
g24	-5.508013	<b>-5.508013</b>	-5.508008	-5.508001	<b>-5.508013</b>	<b>-5.508013</b>	<b>-5.508013</b>	<b>-5.508013</b>	-5.508012	-5.508010

**Table 3**

Constrained optimization results obtained by BBO and B-BBO on 24 constrained functions over 25 independent runs. NF and NS, respectively, stand for number of feasible runs and number of successful runs. "1 vs. 2" means "BBO ( $\alpha=0$ ) vs. B-BBO ( $\alpha=0.5$ )" and "1 vs. 3" means "BBO ( $\alpha=0$ ) vs. B-BBO ( $\alpha=0.8$ )." $p$  Values for each benchmark give the probability that the two sets of results came from the same distribution. Best results are shown in boldface.

Fun.	NF/NS			1 vs. 2	1 vs. 3	Fun.	NF/NS			1 vs. 2	1 vs. 3
	BBO ( $\alpha=0$ )	B-BBO ( $\alpha=0.5$ )	B-BBO ( $\alpha=0.8$ )				$p$ value	$p$ value	BBO ( $\alpha=0$ )		
g01	25/14	<b>25/20</b>	<b>25/18</b>	5.23E-03	7.19E-03	g13	<b>21/14</b>	13/04	09/03	2.1E-03	9.5E-04
g02	25/12	<b>25/19</b>	<b>25/14</b>	0.017	0.32	g14	<b>25/25</b>	<b>25/25</b>	<b>25/25</b>	0.66	0.58
g03	25/10	<b>25/20</b>	<b>25/13</b>	0.34	0.35	g15	12/06	<b>14/06</b>	11/06	0.42	0.38
g04	25/19	<b>25/25</b>	<b>25/25</b>	1.25E-05	4.89E-05	g16	25/12	<b>25/24</b>	25/21	2.9E-04	6.84E-04
g05	17/06	<b>20/14</b>	12/02	0.17	0.19	g17	08/04	<b>15/12</b>	07/03	7.2E-03	0.31
g06	20/08	<b>25/14</b>	<b>25/12</b>	1.34E-03	3.81E-03	g18	<b>25/18</b>	<b>25/18</b>	<b>25/18</b>	0.14	8.9E-03
g07	04/00	08/02	<b>09/05</b>	0.06	0.05	g19	25/21	<b>25/16</b>	<b>25/25</b>	0.04	2.1E-03
g08	25/11	<b>25/25</b>	<b>25/15</b>	4.72E-04	1.83E-03	g20	00/00	00/00	00/00	-	-
g09	20/02	<b>25/24</b>	<b>25/17</b>	8.92E-05	3.87E-04	g21	00/00	00/00	00/00	-	-
g10	05/03	<b>09/03</b>	<b>07/03</b>	4.31E-03	0.02	g22	00/00	00/00	00/00	-	-
g11	25/25	<b>25/25</b>	<b>25/25</b>	1.36E-05	1.03E-03	g23	07/00	<b>12/00</b>	<b>08/00</b>	0.28	0.31
g12	25/20	<b>25/25</b>	<b>25/25</b>	3.78E-04	1.85E-04	g24	<b>25/25</b>	<b>25/25</b>	<b>25/25</b>	4.17E-04	7.52E-04

probability of 0.01. For SPSO 07, we use an inertia weight of 0.8, a cognitive constant of 0.5, a social constant for swarm interaction of 1.0, and a social constant for neighborhood interaction of 1.0. Each algorithm has a population size of 50, and a maximum number of fitness function evaluations (Max\_NFFEs) of 50,000. The results of solving the 24 constrained benchmark functions are given in Tables 4 and 5. All numbers are computed from 25 independent runs.

Tables 4 and 5 show the performance comparison on 24 constrained benchmark functions for B-BBO, SGA, and SPSO 07. We see that for functions g08, g11, g14, and g24, all three algorithms attain the global optimum, but for functions g20, g21, and g22 no feasible solution could be found by any of the algorithms. According to Tables 4 and 5 and the criteria discussed in Section 4.1, B-BBO performs the best on 7 of the 24 benchmark functions, SGA performs the best on 2 functions, and SPSO 07 performs the best on 7 functions. The algorithms

are tied on the rest of the functions. The results indicate that B-BBO is significantly better than SGA, and performs similarly to SPSO 07. If we adopt more state-of-the-art PSO and GA, they could probably perform better than the results shown here (Clerc and Kennedy, 2002; Eberhart and Shi, 2004; Zielinske and Laur, 2006). However, the same could be said for recently proposed improvements for BBO (Du et al., 2009; Ergezer et al., 2009), and the purpose of these comparisons is to show that BBO is a competitive algorithm for solving constrained optimization problems.

In addition, according to Tables 4 and 5, we find that 4 of the functions (g08, g11, g12, g24) for which all 25 runs found the minimum have a rather low dimensionality of 2 or 3, with the exception being g14, for which the dimensionality is 10. We also see that the dimensionality of 2 of the functions (g20 and g22) with no successful runs is large ( $D \geq 20$ ). We see that high dimensionality presents difficulties for EAs.

**Table 4**  
Results obtained by B-BBO ( $\alpha=0.5$ ), SGA, and SPSO 07 on 24 constrained benchmark functions over 25 independent runs. The table shows the best, mean and worst values. Best results for each benchmark function are shown in boldface.

Fun.	Best known solution	B-BBO			SGA			SPSO 07		
		Best	Mean	Worst	Best	Mean	Worst	Best	Mean	Worst
g01	-15.000	<b>-15.000</b>	<b>-14.998</b>	<b>-14.991</b>	<b>-15.000</b>	-14.372	-14.011	<b>-15.000</b>	-14.997	-14.990
g02	-0.803619	<b>-0.803619</b>	-0.802154	-0.801924	<b>-0.803619</b>	-0.76341	-0.70378	<b>-0.803619</b>	<b>-0.803619</b>	<b>-0.803619</b>
g03	-1.0005	<b>-1.0005</b>	<b>-1.0004</b>	<b>-1.0001</b>	<b>-1.0005</b>	-0.99346	-0.99017	<b>-1.0005</b>	-0.9525	-0.9167
g04	-30665.54	<b>-30665.54</b>	<b>-30665.54</b>	<b>-30665.54</b>	-30602.16	-30322.57	-30106.44	<b>-30665.54</b>	-30641.13	-30637.18
g05	5126.497	<b>5126.497</b>	<b>5127.046</b>	<b>5129.913</b>	<b>5126.497</b>	5192.759	5230.718	<b>5126.497</b>	5135.693	5147.916
g06	-6961.814	<b>-6961.814</b>	-6961.005	-6960.874	<b>-6961.814</b>	-6944.947	-6912.362	<b>-6961.814</b>	<b>-6961.811</b>	<b>-6961.793</b>
g07	24.306	<b>24.306</b>	27.051	35.574	<b>24.306</b>	30.046	32.157	<b>24.306</b>	<b>26.035</b>	<b>28.472</b>
g08	-0.095825	<b>-0.095825</b>	<b>-0.095825</b>	<b>-0.095825</b>	<b>-0.095825</b>	<b>-0.095825</b>	<b>-0.095825</b>	<b>-0.095825</b>	<b>-0.095825</b>	<b>-0.095825</b>
g09	680.630	<b>680.630</b>	680.677	681.328	<b>680.630</b>	683.274	693.361	<b>680.630</b>	<b>680.630</b>	<b>680.630</b>
g10	7049.248	<b>7049.248</b>	7057.369	7139.645	<b>7049.248</b>	<b>7049.248</b>	<b>7049.248</b>	7104.802	7315.874	7542.567
g11	0.7499	<b>0.7499</b>	<b>0.7499</b>	<b>0.7499</b>	<b>0.7499</b>	<b>0.7499</b>	<b>0.7499</b>	<b>0.7499</b>	<b>0.7499</b>	<b>0.7499</b>
g12	-1.00000	<b>-1.00000</b>	-0.99994	-0.99991	<b>-1.00000</b>	<b>-1.00000</b>	<b>-1.00000</b>	<b>-1.00000</b>	<b>-1.00000</b>	<b>-1.00000</b>
g13	0.053942	<b>0.053942</b>	<b>0.054418</b>	<b>0.056891</b>	<b>0.053942</b>	0.057164	0.059821	<b>0.053942</b>	0.055046	0.058361
g14	-47.765	<b>-47.765</b>	<b>-47.765</b>	<b>-47.765</b>	<b>-47.765</b>	<b>-47.765</b>	<b>-47.765</b>	<b>-47.765</b>	<b>-47.765</b>	<b>-47.765</b>
g15	961.715	<b>961.715</b>	962.627	965.736	<b>961.715</b>	963.049	966.377	<b>961.715</b>	<b>962.003</b>	<b>965.024</b>
g16	-1.90516	<b>-1.90516</b>	-1.90511	-1.90508	<b>-1.90516</b>	-1.90373	-1.90045	<b>-1.90516</b>	<b>-1.90516</b>	<b>-1.90516</b>
g17	8853.540	<b>8853.540</b>	<b>8853.600</b>	<b>8853.692</b>	<b>8853.540</b>	8866.934	8879.824	<b>8853.540</b>	8854.276	8855.031
g18	-0.866025	<b>-0.866025</b>	-0.865521	-0.863036	<b>-0.866025</b>	<b>-0.866025</b>	<b>-0.866025</b>	<b>-0.866025</b>	-0.865841	-0.864033
g19	32.656	<b>32.656</b>	<b>32.659</b>	<b>32.665</b>	37.478	45.037	52.268	<b>32.656</b>	32.689	33.027
g20	0.204979	-	-	-	-	-	-	-	-	-
g21	193.725	-	-	-	-	-	-	-	-	-
g22	236.431	-	-	-	-	-	-	-	-	-
g23	-400.055	-396.034	-391.319	-384.183	-397.251	-395.432	-387.493	<b>-400.055</b>	<b>-399.0157</b>	<b>-397.393</b>
g24	-5.508013	<b>-5.508013</b>	<b>-5.508013</b>	<b>-5.508013</b>	<b>-5.508013</b>	<b>-5.508013</b>	<b>-5.508013</b>	<b>-5.508013</b>	<b>-5.508013</b>	<b>-5.508013</b>

**Table 5**  
Results obtained by B-BBO ( $\alpha=0.5$ ), SGA, and SPSO 07 on 24 constrained benchmark functions over 25 independent runs. NF and NS, respectively, stand for number of feasible runs and number of successful runs for B-BBO, SGA, and SPSO 07. Best results for each benchmark function are shown in boldface.

Fun.	NF/NS			Fun.	NF/NS			Fun.	NF/NS		
	B-BBO	SGA	SPSO 07		B-BBO	SGA	SPSO 07		B-BBO	SGA	SPSO 07
g01	<b>25/20</b>	<b>25/14</b>	<b>25/18</b>	g09	<b>25/24</b>	06/02	<b>25/25</b>	g17	<b>15/12</b>	13/04	10/08
g02	<b>25/19</b>	<b>25/11</b>	<b>25/25</b>	g10	09/03	<b>25/25</b>	02/00	g18	<b>25/22</b>	<b>25/25</b>	<b>25/21</b>
g03	<b>25/20</b>	<b>25/12</b>	<b>25/12</b>	g11	<b>25/25</b>	<b>25/25</b>	<b>25/25</b>	g19	<b>25/16</b>	07/00	13/01
g04	<b>25/25</b>	07/00	15/08	g12	<b>25/25</b>	<b>25/25</b>	<b>25/25</b>	g20	00/00	00/00	00/00
g05	<b>20/14</b>	17/10	18/12	g13	<b>13/03</b>	08/03	11/03	g21	00/00	00/00	00/00
g06	<b>25/14</b>	09/03	<b>25/22</b>	g14	<b>25/25</b>	<b>25/25</b>	<b>25/25</b>	g22	00/00	00/00	00/00
g07	08/05	06/06	<b>12/08</b>	g15	14/06	17/06	<b>21/06</b>	g23	12/00	07/00	<b>25/07</b>
g08	<b>25/25</b>	<b>25/25</b>	<b>25/25</b>	g16	<b>25/24</b>	<b>25/22</b>	<b>25/25</b>	g24	<b>25/25</b>	<b>25/25</b>	<b>25/25</b>

From Tables 4 and 5, we further find that the number of constraints for successfully optimized functions (g08, g11, g12, g14, g24) is mostly rather small, with the exception of g16, which contains 38 inequality constraints, and g18, which includes 13 inequality constraints. Therefore, it appears that the number of inequality constraints does not create difficulties for EAs in general.

For functions with a high number of equality constraints (g20 and g22) particularly bad results are obtained: neither a feasible run nor a successful run were accomplished for any of the three algorithms. The function with the next highest number of equality constraints is g21, for which neither a feasible run nor a successful run were accomplished. Therefore, it is concluded that a high number of equality constraints provides a challenge for the three algorithms.

Based on the above results, it appears that the dimensionality of the function and the number of equality constraints create difficulties for EAs, at least for the three algorithms applied here. Average convergence graphs are shown in Fig. 2. We see that for most of the benchmark functions considered in this paper, B-BBO, SGA, and SPSO 07 all converge fast and obtain good performance.

Note that for some of the graphs, we use a logarithmic scale for the x-axis or y-axis to distinguish the results of the three algorithms.

## 5. Conclusion

In this paper, we have generalized biogeography-based optimization (BBO) to handle single-objective optimization problems with multiple constraints. Instead of using penalty functions for constraint-handling, we used a method based on Deb's feasibility rules, which does not introduce any additional tuning parameters beyond those required for unconstrained optimization. In addition, we introduced a blended migration operator to BBO to obtain B-BBO, which is motivated by the blended crossover operator in GAs. The experiments that we performed: (1) verify that BBO is a competitive algorithm for solving constrained optimization problems; (2) show that blended migration further improves BBO's performance; (3) show that the parameter  $\alpha$  can affect B-BBO's optimization ability; and (4) show that BBO can solve constrained optimization problems

better than SGA and similar to SPSO 07 for the benchmark problems that we investigated.

We do not consider computational cost in this paper. According to the literature (Mezura-Montes and Coello, 2005), 3/4 of the computational cost of evolutionary search is typically consumed by the fitness function evaluation, and only 1/4 (often less) is consumed by the search process. The mechanism of BBO is simple, like that of GAs and PSO, therefore, the computational cost

of BBO and other EAs will be the same since it will be dominated by the difficulty of the landscape of the constrained objective function.

In addition, in this paper we use feasibility rules, which make a clear distinction between feasible and infeasible solutions, to solve the constrained problems. This approach indicates that a feasible solution is always preferred over an infeasible solution. In reality, a marginally infeasible solution with a good objective

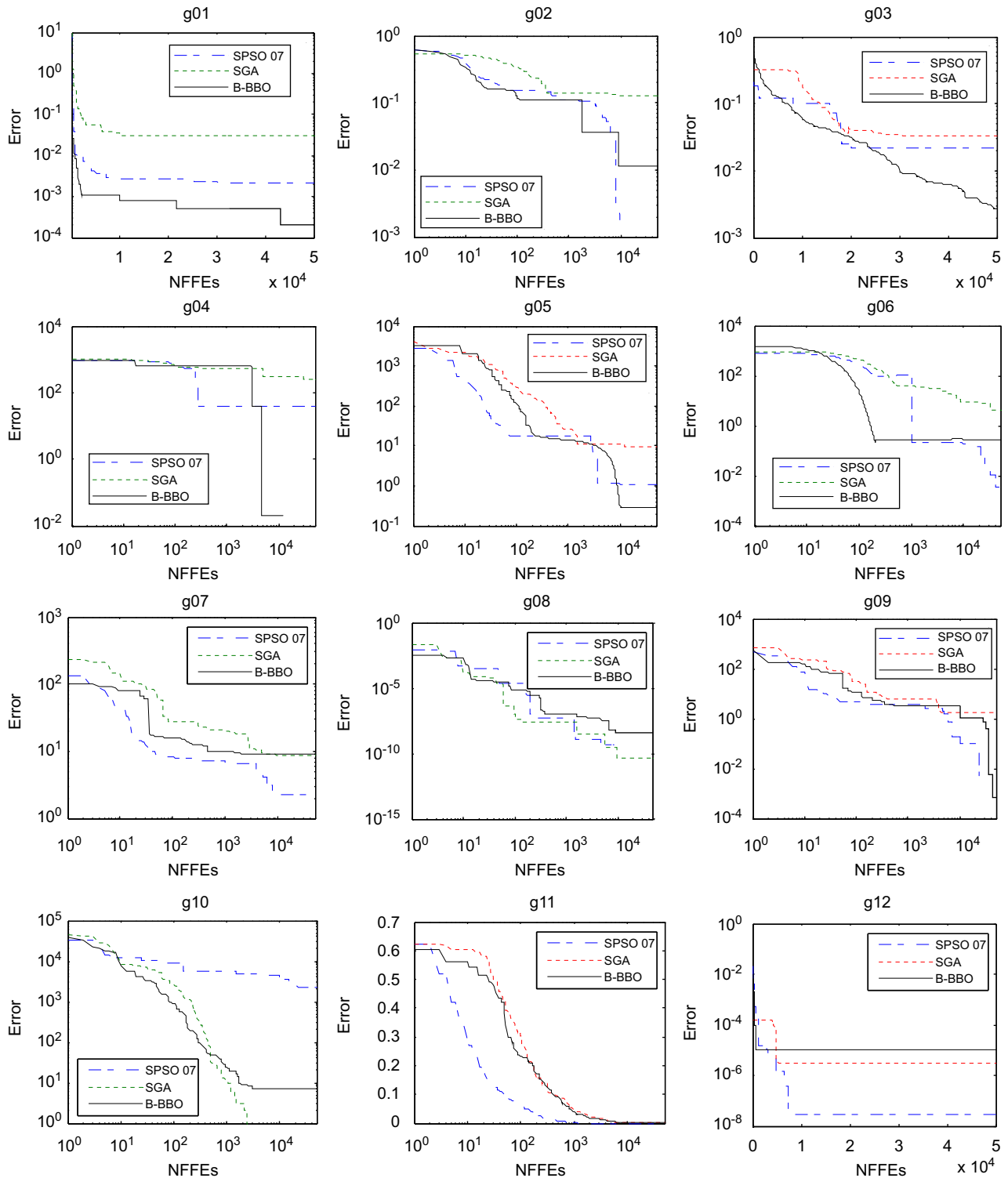


Fig. 2. Convergence curves of B-BBO ( $\alpha=0.5$ ), SGA, and SPSO 07 for 21 constrained benchmark functions (excluding functions g20, g21, and g22), averaged over 25 independent runs. NFFEs denote the number of fitness function evaluations.

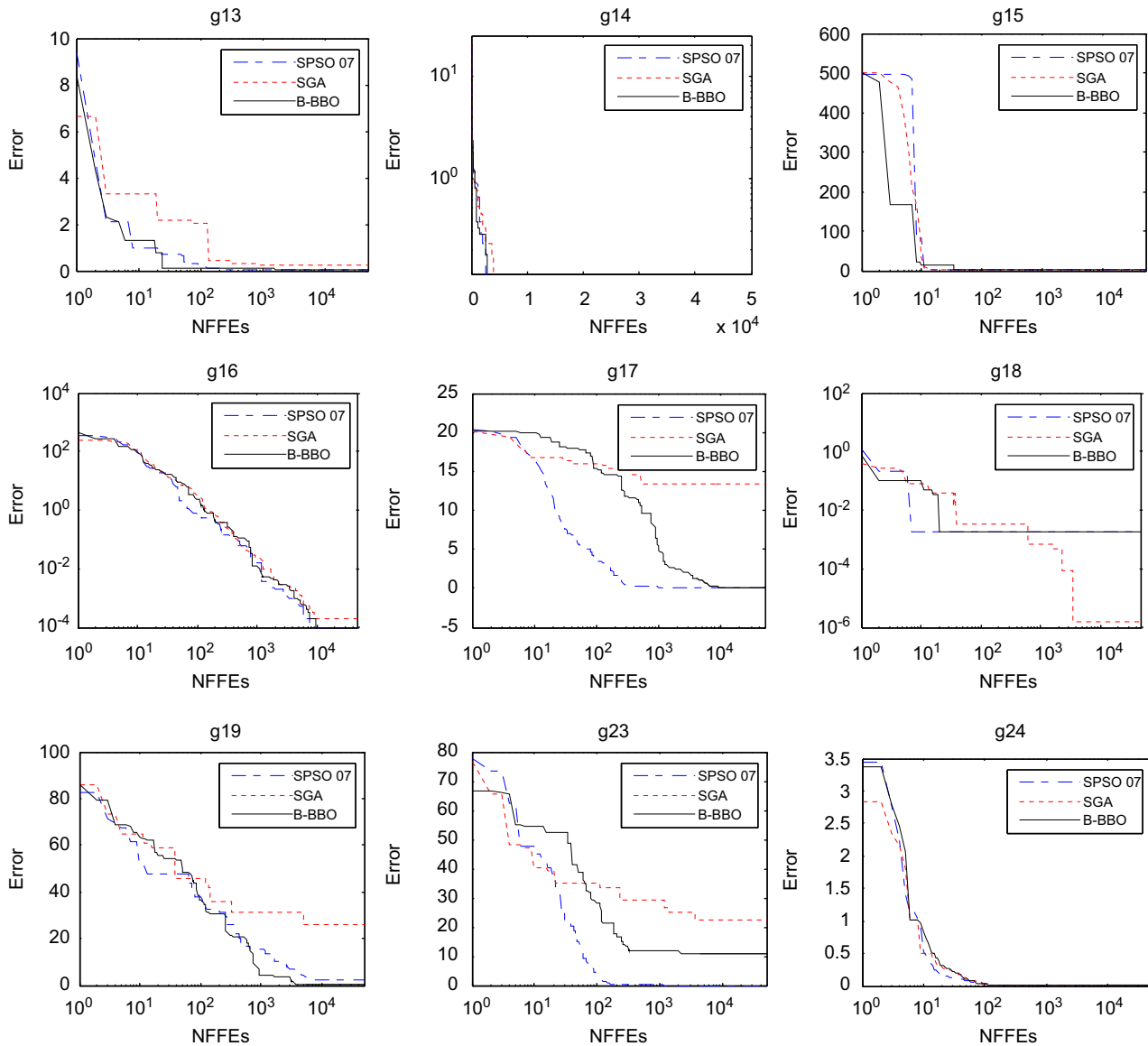


Fig. 2. (Continued)

value could be preferred over a feasible solution with a poor function value (Wang et al., 2008). Solutions to constrained optimization problems are likely to lie on constraint boundaries, and hence preferring a marginally feasible solution could be beneficial. For future work, we will investigate the application of more advanced constrained solution methods in BBO, such as the adaptive tradeoff model (Wang et al., 2008), to balance feasible and infeasible solutions.

BBO presents promising potential but still requires additional theoretical and empirical investigation. For future work, we can further tune the migration operator as inspired by other aspects of biogeography. For example, it has been shown that sigmoid migration curves like those found in nature, rather than the simple linear ones explored in this paper, generally give better BBO performance (Ma, 2010). It remains to be seen how this and other migration curve changes will affect constrained BBO performance. We intend to explore BBO for real-world constrained optimization problems. In this paper, 24 benchmark functions have been presented, including quadratic, cubic, non-linear, polynomial, and linear functions. It is a challenge for any algorithm to scale to high-dimension problems, so the exploration

of BBO for high-dimension real-world constrained optimization problems is an area for future work. Another area for future work is the development of theoretical results relating to blended migration, and constrained BBO convergence. Some theoretical work for unconstrained BBO has been performed using Markov theory (Simon et al., 2009), but it remains to be seen if that approach, or if other approaches, can give fruitful theoretical results for B-BBO or for constrained BBO. The use of oppositional learning has been incorporated into unconstrained BBO to yield oppositional BBO (OBBO) (Ergezer et al., 2009). Future work could see how OBBO could be modified to solve constrained optimization problems, or how oppositional learning could be incorporated into the BBO extensions proposed in this paper.

#### Acknowledgment

This work was partially supported by the Zhejiang Provincial Natural Science Foundation of China under Grant no. Y1090866, and by Grant 0826124 in the CMMI Division of the Engineering Directorate of the National Science Foundation.



## References

- Bratton, D., Kennedy, J., 2007. Defining a standard for particle swarm optimization. In: *Proceedings of the 2007 IEEE Swarm Intelligence Symposium*, April 2007, Honolulu, Hawaii. pp. 120–127.
- Coello, C., 2002. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering* 191 (11), 1245–1287.
- Clerc, M., Kennedy, J., 2002. The particle swarm—explosion, stability, and convergence in a multidimensional complex space. *IEEE Transaction on Evolutionary Computation* 6 (1), 58–73.
- Deb, K., 2000. An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering* 186 (2–4), 311–338.
- Du, D., Simon, D., Ergezer, M., 2009. Biogeography-based optimization combined with evolutionary strategy and immigration refusal. In: *Proceedings of the IEEE Conference on Systems, Man, and Cybernetics*, October 2009, San Antonio, Texas. pp. 1023–1028.
- Eberhart, R., Shi, Y., 2004. Guest editorial: special issue on particle swarm optimization. *IEEE Transaction on Evolutionary Computation* 8 (3), 201–203.
- Ergezer, M., Simon, D., Du, D., 2009. Oppositional biogeography-based optimization. In: *Proceedings of the IEEE Conference on Systems, Man, and Cybernetics*, October 2009, San Antonio, Texas. pp. 1035–1040.
- Eshelman, L.J., Schaffer, J.D., 1993. Real-coded genetic algorithms and interval-schemata. In: Whitley, D. (Ed.), *Foundations of Genetic Algorithms II*. pp. 187–202.
- Gardner, B., Simon, D., 2009. Evolutionary algorithm sandbox: a web-based graphical user interface for evolutionary algorithms. In: *Proceedings of the IEEE Conference on Systems, Man, and Cybernetics*, October 2009, San Antonio, Texas. pp. 583–588.
- Huang, V., Qin, A., Suganthan, P., 2006. Self-adaptive differential evolution algorithm for constrained real-parameter optimization. In: *Proceedings of the IEEE Congress on Evolutionary Computation*, July 2006, Vancouver, Canada. pp. 17–24.
- Khatib, W., Fleming, P., 1998. The stud GA: a mini revolution?. In: Eiben, A., Back, T., Schoenauer, M., Schwefel, H. (Eds.), *In Parallel Problem Solving from Nature*. Springer, New York, pp. 683–691.
- Kundra, H., Kaur, A., Panchal, V., 2009. An integrated approach to biogeography based optimization with case based reasoning for retrieving groundwater possibility. In: *Proceedings of the Eighth Annual Asian Conference and Exhibition on Geospatial Information, Technology and Applications*, August 2009, Singapore.
- Liang, J.J., Runarsson, T.P., Mezura-Montes, E., Clerc, M., Suganthan, P.N., Coello Coello, C.A.C., Deb, K., 2005. Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization. Technical Report, Nanyang Technological University, Singapore. <[http://www3.ntu.edu.sg/home/epnsugan/index\\_files/CEC-06/CEC06.htm](http://www3.ntu.edu.sg/home/epnsugan/index_files/CEC-06/CEC06.htm)>.
- Lomolino, M., Riddle, B., Brown, J., 2009. *Biogeography*. Sinauer Associates.
- Ma, H., 2010. An analysis of the equilibrium of migration models for biogeography-based optimization. *Information Sciences* 180 (18), 3444–3464.
- Ma, H., Ni, S., Sun, M., 2009. Equilibrium species counts and migration model tradeoffs for biogeography-based optimization. 2009. In: *Proceedings of the IEEE Conference on Decision and Control*, December 2009, Shanghai. pp. 3306–3310.
- Ma, H., Simon, D., 2010. Biogeography-based optimization with blended migration for constrained optimization problems. In: *Proceedings of the Genetic and Evolutionary Computation Conference*, July 2010, Portland, OR. pp. 417–418.
- McTavish, T., Restrepo, D., 2008. Evolving solutions: the genetic algorithm and evolution strategies for finding optimal parameters. In: Smolinski, T., Milanova, M., Hassani, A. (Eds.), *Applications of Computational Intelligence in Biology*. Springer, New York, pp. 55–78.
- Mezura-Montes, E., Coello, C., 2005. A simple multimembered evolution strategy to solve constrained optimization problems. *IEEE Transactions on Evolutionary Computation* 9 (1), 1–17.
- Mezura-Montes, E., Palomeque-Oritz, A., 2009. Parameter control in differential evolution for constrained optimization. In: *Proceedings of the IEEE Congress on Evolutionary Computation*, May 2009, Trondheim, Norway. pp. 1375–1382.
- Mühlenbein, H., Schlierkamp-Voosen, D., 1993. Predictive models for the breeder genetic algorithm I. Continuous parameter optimization. *Evolutionary Computation* 1 (1), 25–49.
- Panchal, V., Singh, P., Kaur, N., Kundra, H., 2009. Biogeography based satellite image classification. *International Journal of Computer Science and Information Security* 6 (2), 269–274.
- Particle Swarm Central, <<http://www.particleswarm.info/>>.
- Rarick, R., Simon, D., Villaseca, F.E., Vyakaranam, B., 2009. Biogeography-based optimization and the solution of the power flow problem. In: *Proceedings of the IEEE Conference on Systems, Man, and Cybernetics*, October 2009, San Antonio, Texas. pp. 1029–1034.
- Schwefel, H.P., 1995. *Evolution and Optimum Seeking*. Wiley Press.
- Simon, D., 2008. Biogeography-based optimization. *IEEE Transactions on Evolutionary Computation* 12 (6), 702–713.
- Simon, D., Ergezer, M., Du, D., 2009. Population distributions in biogeography-based optimization algorithms with elitism. In: *Proceedings of the IEEE Conference on Systems, Man, and Cybernetics*, October 2009, San Antonio, Texas. pp. 1017–1022.
- Smith, A., Coit, D., 1997. Penalty functions. In: Back, T., Fogel, D., Michalewicz, Z. (Eds.), *Handbook of Evolutionary Computation*. Taylor & Francis.
- Thomas, P., Runarsson, Y., Yao, X., 2000. Stochastic ranking for constrained evolutionary optimization. *IEEE Transaction on Evolutionary Computation* 4 (3), 284–294.
- Wang, Y., Cai, Z., Zhou, Y., Zeng, W., 2008. An adaptive tradeoff model for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation* 12 (1), 80–92.
- Whittaker, R., 1998. *Island Biogeography*. Oxford University Press.
- Yao, X., Liu, Y., Lin, G., 1999. Evolutionary programming made faster. *IEEE Transaction on Evolutionary Computation* 3 (2), 82–102.
- Yeniay, O., 2005. Penalty function methods for constrained optimization with genetic algorithms. *Mathematical and Computational Applications* 10 (1), 45–56.
- Zbigniew, M., Marc, S., 1996. Evolutionary algorithms for constrained parameter optimization problem. *Evolutionary Computation* 4 (1), 1–32.
- Zielinske, K., Laur, R., 2006. Constrained single-objective optimization using differential evolution. In: *Proceedings of the IEEE Congress on Evolutionary Computation*, July 2006, Vancouver, Canada. pp. 223–230.
- Zielinske, K., Laur, R., 2006. Constrained single-objective optimization using particle swarm optimization. In: *Proceedings of the IEEE Congress on Evolutionary Computation*, July 2006, Vancouver, Canada. pp. 443–450.