

Research Article

Hybrid Biogeography Based Optimization for Constrained Numerical and Engineering Optimization

Zengqiang Mi,¹ Yikun Xu,¹ Yang Yu,¹ Tong Zhao,¹ Bo Zhao,² and Liqing Liu¹

¹State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources, North China Electric Power University, Baoding 071003, China

²Electronics-Controlling Lab of Construction Vehicle, Jilin University, Changchun 130022, China

Correspondence should be addressed to Yang Yu; ncepu.yy@163.com

Received 16 April 2014; Accepted 18 July 2014

Academic Editor: Baozhen Yao

Copyright © 2015 Zengqiang Mi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Biogeography based optimization (BBO) is a new competitive population-based algorithm inspired by biogeography. It simulates the migration of species in nature to share information. A new hybrid BBO (HBBO) is presented in the paper for constrained optimization. By combining differential evolution (DE) mutation operator with simulated binary crossover (SBX) of genetic algorithms (GAs) reasonably, a new mutation operator is proposed to generate promising solution instead of the random mutation in basic BBO. In addition, DE mutation is still integrated to update one half of population to further lead the evolution towards the global optimum and the chaotic search is introduced to improve the diversity of population. HBBO is tested on twelve benchmark functions and four engineering optimization problems. Experimental results demonstrate that HBBO is effective and efficient for constrained optimization and in contrast with other state-of-the-art evolutionary algorithms (EAs), the performance of HBBO is better, or at least comparable in terms of the quality of the final solutions and computational cost. Furthermore, the influence of the maximum mutation rate is also investigated.

1. Introduction

With the development of science and engineering, related optimization problems become more and more complex. Optimization methods are being confronted with great challenges brought by some undesirable but unavoidable characteristics of optimization problems such as being high-dimensional, nondifferentiable, nonconvex, noncontinuous, and so on. Efficient optimization methods are urgently required by the complicated optimization problems in the real world. Therefore, various evolutionary algorithms have been applied to solve difficult optimization problems in recent decades, which include GAs [1], particle swarm optimization approach (PSO) [2], DE [3, 4], ant colony optimization (ACO) [5], artificial bee colony strategy (ABC) [6, 7], and BBO [8, 9].

Biogeography based optimization (BBO) is a new population-based algorithm. It simulates the blossom and extinction of species in different habitats based on the mathematical model of biogeography. Decision variables of better solutions tend to be shared in the migration operation and decision

variables of each solution are probabilistically replaced to improve the diversity of population in mutation operation. Due to good search ability, BBO has been applied to PID parameter tuning [10], parameter estimation of chaotic system [11], complex system optimization [12], satellite image classification [13], and so forth.

In comparison with other EAs, owing to direct-copying-based migration and random mutation, exploration ability of BBO is not so efficient despite outstanding exploitation. In other words, BBO can be easily trapped into local optimum and suffer from premature convergence owing to lack of corresponding exploration to balance its exploitation.

In order to overcome the weakness of BBO, lots of improved BBO variants have been proposed. Ma [14] presented six different migration mathematical models and made basic improvements on the migration operator. Gong et al. [15] hybridized the DE with the migration operator of BBO to balance the exploration and exploitation of BBO. Motivated by blended crossover operator in GAs, Ma and Simon [16] got decision variables of an offspring by blending

corresponding decision variables of two parent individuals based on different weighting constants. Li and Yin [17] proposed multiparent migration in which three consecutive individuals are chosen to generate three new individuals by basic BBO and then the new individuals are modified like multiparent crossover in GA; besides, the mutation was improved based on Gaussian operator. Li et al. [18] updated the decision variables not selected in migration by generating a perturbation from the neighborhood solutions and Gaussian operator was integrated into mutation. Wang and Xu [11] integrated DE mutation operator into migration operator of BBO and simplex search was introduced to improve the searching accuracy. Sayed et al. [10] formed new decision variables of an offspring by combining corresponding decision variables from two different parents with weighted constants related to the rank of their fitness in migration. Boussaïd et al. [19] proposed a new hybrid BBO in which new solutions are first generated by DE mutation and then modified by migration of original BBO. Xiong et al. [20] utilized four individuals' features to construct a new solution in proposed polyphyletic migration and orthogonal learning was introduced to further enhance converge speed toward global optimum.

In order to balance the exploration and exploitation of BBO, a new hybrid BBO called as HBBO is proposed in the paper. The unique points of HBBO are shown as the following. On one hand, a new hybrid mutation operator combining DE mutation and SBX is presented in HBBO while operators of EAs are often hybridized with migration operator in most of BBO variants. On the other hand, HBBO provides a new method to extend BBO to optimize constrained problems well due to only a few BBO variants available for constrained optimization in previous literatures. In addition, DE is applied to evolve one half of population to improve convergence speed further and chaotic search is introduced to enhance the diversity of population. Experiments have been conducted on twelve benchmark functions and four engineering optimization problems, and HBBO is compared with many other state-of-the-art algorithms from the quality of solutions obtained and computational cost. Furthermore, the influence of maximum mutation rate on HBBO is studied.

The rest of the paper is organized as follows. Constrained optimization, basic BBO, mutation strategies of DE, and SBX are briefly introduced in Section 2. In Section 3, the HBBO method proposed in the paper is specifically depicted. The comparison with six state-of-the-art algorithms on twelve benchmark functions is presented in Section 4. In Section 5, HBBO is compared with other methods on four well known engineering optimization problems. Section 6 further demonstrates the efficiency of HBBO and presents the investigation on the influence of maximum mutation rate. Finally, the work is concluded in Section 7.

2. Preliminary

2.1. Constrained Optimization. Constrained optimizations are always inevitable in scientific study and engineering

design. A general constrained optimization problem can be written as the following

$$\begin{aligned} & \text{Minimize} && f(\mathbf{x}) \\ & \text{subject to} && g_i(\mathbf{x}) \leq 0, \quad i = 1, 2, \dots, p \\ & && h_j(\mathbf{x}) = 0, \quad j = 1, 2, \dots, q, \end{aligned} \quad (1)$$

where \mathbf{x} represents the solution vector $\mathbf{x} = [x_1, x_2, \dots, x_D]^T$, D is the dimensionality of a solution in the paper, p is the number of inequality constraints, and q is the number of equality constraints. In common practice, equality constraints are often transformed to inequality constraints with a given small tolerance δ . For example, the equality constraint above can be converted to $|h_i(\mathbf{x})| \leq \delta$. In the paper, the feasible-based rule by Deb [21] is applied to handle constraint. In the constraint handling mechanism, fitness value and constraint violation are considered separately based on the following criterions: (a) any feasible solution is preferred to any infeasible solution; (b) between two feasible solutions, the one having smaller objective function value is preferred; (c) between two infeasible solutions, the one having smaller constraint violation is preferred.

2.2. Biogeography Based Optimization. Biogeography is the study of the distribution of species on earth surface over time. BBO is proposed based on the mathematical model of biogeography by Simon in 2008 [8]. In BBO, every solution is analogous to a habit; habit suitability index (HSI) is utilized to measure habits just like fitness function in other EAs; the elements that characterize habitability are called suitability index variables (SIVs) which are identical to the decision variables in other EAs. A good solution is similar to a habitat with high HSI which have a large number of species and vice versa. The species in habitats with high HSI tend to emigrate to habitats with low HSI. That is, habitats with high HSI tend to share their features while habitats with low HSI are inclined to accept the features from good habitats.

In BBO, each individual evolves by immigration and mutation operator. The SIVs of individuals are probabilistically shared in migration operator as shown in Algorithm 1, where X_{ij} is the j th SIV of i th individual in the population, λ_i here represents the immigration rate of X_i , and μ_k is the emigration rate of X_k , which are related to the number of species in the corresponding habitat; NP is the population size in the paper.

The following mathematical model is applied to calculate immigration rate and emigration rate owing to its outstanding performance in [22]:

$$\begin{aligned} \lambda_i &= I \left(1 - \frac{S_i}{n} \right) \\ \mu_i &= E \left(\frac{S_i}{n} \right)^2, \end{aligned} \quad (2)$$

where S_i is the number of species in habitat X_i , I and E are, respectively, the maximum value of immigration and emigration rate; n is equal to S_{\max} .

```

Target individual  $X_i$  for migration
For  $j = 1$  to  $D$  do
    Select  $X_{ij}$  with probability  $\propto \lambda_i$ 
    If  $X_{ij}$  is selected
        For  $k = 1$  to NP do
            Select  $X_k$  with probability  $\propto \mu_k$ 
            If  $X_k$  is selected
                Replace  $X_{ij}$  with  $X_{kj}$ 
            End if
        End for
    End if
End for
    
```

ALGORITHM 1: Migration operator.

```

Target individual  $X_i$  for mutation
For  $j = 1$  to  $D$ 
    Select  $X_{ij}$  based on  $m_i$ 
    If  $X_{ij}$  is selected
        Replace it with a randomly generated SIV
    End if
End for
    
```

ALGORITHM 2: Mutation operator.

In mutation operator, it is probabilistically decided whether or not to replace each SIV in a solution by a randomly generated SIV in the light of mutation rate. The detail of mutation operator is shown in Algorithm 2. The mutation rate m can be calculated as follows:

$$m_i = m_{\max} \left(1 - \frac{P_i}{P_{\max}} \right), \quad (3)$$

where $P_{\max} = \max(P_i)$, P_i represents the priori probability of existence for i th individual; m_{\max} is a user-defined parameter which represents the maximum mutation rate.

More details about basic BBO can be found in [8, 23].

2.3. Differential Evolution. DE algorithm is a population-based stochastic search method proposed by Storn and Price in 1997 [3]. Due to simple structure, few parameters, easy use, and fast convergence speed, DE has obtained wide application in various regions. DE generates new individuals by perturbing a randomly chosen individual with weighted differences for some couples of different individuals. Only when the offspring outperforms corresponding parent, the offspring survives as the parent for next generation. Mutation operator is the most important part in DE. In this part, only three widely applied mutation strategies are briefly introduced as follows:

(1) *rand/1*:

$$Y_{ij} = X_{r_1j} + F(X_{r_2j} - X_{r_3j}), \quad (4)$$

(2) *best/1*:

$$Y_{ij} = X_{g_j} + F(X_{r_1j} - X_{r_2j}), \quad (5)$$

(3) *rand to best/1*:

$$Y_{ij} = X_{r_1j} + F(X_{g_j} - X_{r_1j}) + F(X_{r_2j} - X_{r_3j}), \quad (6)$$

where X_g is the best individual in population; r_1, r_2, r_3 are uniformly distributed different numbers in the range $[1, NP]$; F is mutation scaling factor; Y represents the new individuals generated by mutation operator.

2.4. SBX of GA. Genetic algorithms simulate the evolutionary process in nature to solve optimization problems. In GA, some good individuals are chosen based on Deb's feasible-based rule. Different individuals can share information in crossover operator. SBX is one of the most popular crossover operators which can explore the neighborhood region of parent individual as follows:

$$C_{1j} = \frac{1}{2} \left[(1 - \beta_j) Parent_{1j} + (1 + \beta_j) Parent_{2j} \right] \quad (7)$$

$$C_{2j} = \frac{1}{2} \left[(1 + \beta_j) Parent_{1j} + (1 - \beta_j) Parent_{2j} \right], \quad (8)$$

where C_{ij} is the j th decision variable of the i th offspring individual; $Parent_{ij}$ is the j th decision variable of i th parent individual selected. β can be obtained by the random number u in $[0, 1]$ based on (9), where η is the distribution index for crossover. The detail of SBX can be found in [24]. Consider the following:

$$\beta(u) = \begin{cases} (2u)^{1/(\eta+1)} & \text{if } u \leq 0.5 \\ \frac{1}{[2(1-u)]^{1/(\eta+1)}} & \text{if } u > 0.5. \end{cases} \quad (9)$$

3. Proposed Algorithm HBBO

In mutation operator of basic BBO, SIVs are replaced probabilistically by new SIVs randomly generated. Although the mutation of BBO can improve the diversity of population, the random operation brings blindness to search. To modify the defect, a new hybrid mutation operator is proposed, in which DE mutation operator and SBX are mixed to generate promising SIV as shown in Algorithm 3. From Algorithm 3, it can be seen that two candidate SIVs are generated for each SIV mutated: one is gotten by DE *rand/1* mutation and the other by SBX. One point should be stated specially: X_{r_1} in DE *rand/1* mutation and $Parent_1$ and $Parent_2$ in SBX are all randomly selected from the first half of parent population which is sorted based on Deb's feasible rule (better one in front); X_{r_2}, X_{r_3} in DE *rand/1* mutation are randomly selected from the whole population. The core idea of hybrid mutation is based on the following considerations. First, owing to well-known performance in locating the region of global optimum, DE mutation can explore new search space with more clear direction towards global optimum instead of the random mutation in the original BBO. Second, SBX can explore the neighbor region of parent individual so that it can be combined with DE to explore search space efficiently. Third, the combination of DE mutation and SBX can balance the exploitation ability of BBO.

```

Target individual  $X_i$  for mutation;
For  $j = 1$  to  $D$ 
  If  $X_{ij}$  is selected for mutation as basic BBO
    Get two candidate SIVs of offspring;
    (1) Get a temp SIV by DE rand/1 mutation;
    (2) If  $rand < 0.5$  ( $rand$  is random number in  $[0, 1]$ )
      Get another temp SIV by (7);
    Else
      Get another temp SIV by (8);
    End if
  Else
    the  $j$ th SIV of  $i$ th individual in population Island survives as SIV of offspring;
    (population Island contains new individuals gotten by migration operator)
  End if
End for
Two temp offspring individuals are gotten for  $X_i$ , and the better one survives as offspring;

```

ALGORITHM 3: Hybrid mutation operator.

In order to speed up convergence, DE is further hybridized with BBO. The first half of parent population also evolves by two DE mutation strategies (*rand/1* and *rand to best/1*) and two new individuals are generated for each one in the first half. The best one among these two new individuals and corresponding parent individual survives to replace corresponding one in the second half of parent population.

For convenience and easy use, self-adaption mechanism for mutation scaling factor of DE proposed in [25] is applied, in which each individual is given an independent mutation scaling factor. The self-adaption mechanism is written as follows:

$$F_i^{G+1} = \begin{cases} F_i + rand_1 \cdot F_u & \text{if } rand_2 < 0.3 \\ F_i^G & \text{otherwise,} \end{cases} \quad (10)$$

where F_i^{G+1} represents mutation scaling factor for i th individual in $(G+1)$ th generation, F_i^G represents mutation scaling factor for i th individual in G th generation; F_l, F_u are the lower boundary and the change range of mutation scaling factor, respectively, and $rand_1$ and $rand_2$ are uniformly distributed numbers in $[0, 1]$.

Based on unique ergodicity, inherent stochastic property, and irregular chaos, chaotic search can reach each situation in given space so that it can contribute to the escape from the local optimum and is often integrated into EAs to enhance global search ability. Hence, the chaotic search is brought in for the first half of population. In the paper, logistic maps are used to generate chaotic sequences as follows:

$$W_{i+1,j} = \varepsilon \cdot W_{ij} \cdot (1 - W_{ij}) \quad 0 < W_{1j} < 1, \quad (11)$$

where $W_{1,j}, W_{i,j}, W_{i+1,j}$ are, respectively, the first i th, $(i+1)$ th element of sequence $\{W_{1,j}, W_{2,j}, \dots, W_{N,j}\}$; N represents the size of population for chaotic search; $j = 1, 2, \dots, D$; ε is the control parameter; sequence $\{W_{1,j}, W_{2,j}, \dots, W_{N,j}\}$ is chaotic when $\varepsilon = 4$ and $W_{1j} \neq 0.25, 0.5, 0.75$. We can apply the following equation to perform chaotic search for

i -th individual X_i in parent population by vector $W_i = [W_{i1}, W_{i2}, \dots, W_{iD}]^T$:

$$X'_i = X_i + R^G \cdot * (2W_i - 1), \quad (12)$$

where X'_i is the new individual generated by chaotic search for X_i ; R^G represents the search radius vector in G th generation; each dimension in R^G represents the search radius for the variables in corresponding dimension of solutions in parent population.

In the initial phase, large chaotic search radius is helpful for escape from the local optimum; small chaotic search radius can improve the accuracy of search at the later stage of evolution. The search radius R_j^G for j th decision variable is adapted as follows:

$$R_j^G = Bound_j \cdot \left(0.005 - 0.004 \frac{G}{G_{\max}} \right), \quad (13)$$

where $Bound_j$ is the initial search radius for j th decision variable; G_{\max} is the maximum number of generations.

In order to maintain solutions feasible, any new decision variable generated in evolution process should be repaired if it violates boundary. Suppose that t_j is the j th decisions variable in certain new individual generated during evolution process. If t_j violates given boundaries, it can be modified as follows:

$$t_j = lb_j + (ub_j - lb_j) \cdot randnum_j, \quad (14)$$

where lb_j, ub_j are, respectively, the lower boundary and upper boundary of j th decision variable; $randnum_j$ is uniformly distributed number in $[0, 1]$ in each dimension.

The whole procedure of HBBO is described in Algorithm 4 in detail. From Algorithm 4, it can be seen that operations are mainly concentrated on the first half of population. It can be explained as follows. First, the convergence speed can be improved by focusing operations on the first half. Second, the information of the second half

```

Generate the initial population POP and vector F of mutation scaling factors;
Evaluate the fitness and constraint violations of each individual in POP;
For each generation do
  Sort the individuals in POP based on Deb's feasibility-based rule (better in front);
  For each one in POP's first half
    Get two new individuals by two DE mutation strategies (rand/1, rand to best/1);
    Evaluate the fitness value and constraint violations of these two new individuals;
    Among these two new individuals and corresponding parent individual, the best one is stored into population
    Tempbest;
  End for
  Update the vector P of prior probability;
  For each one in the first half of POP
    Generate a new individual by Algorithm 1, and store it into population Island;
  End for
  For each one in the first half of POP
    Get one offspring by Algorithm 3 and replace the corresponding individual in population Island with it;
  End for
  Go on chaotic search for the first half of POP and the new individuals generated are stored into population tempIsland;
  Make a contrast between the corresponding ones in Island and tempIsland, and the first half of POP, the best one survives
  as the corresponding one in POP for next generation;
  The population Tempbest replace the second half of POP as the parent ones for next generation;
  Update F, R by (10), (13) respectively;
End for

```

ALGORITHM 4: Hybrid biogeography based optimization.

TABLE 1: Main characteristics of the twelve selected benchmark functions.

Benchmark function	D	Type of objective function	ρ	LI	NI	NE	a
G01	13	Quadratic	0.0003%	9	0	0	6
G02	20	Nonlinear	99.9970%	1	1	0	1
G03	10	Nonlinear	0.0000%	0	0	1	1
G04	5	Quadratic	26.9668%	0	6	0	2
G05	4	Nonlinear	0.0000%	2	0	3	3
G06	2	Nonlinear	0.0064%	0	2	0	2
G07	10	Quadratic	0.0002%	3	5	0	6
G08	2	Nonlinear	0.8575%	0	2	0	2
G09	7	Nonlinear	0.5235%	0	4	0	2
G10	8	Linear	0.0007%	3	3	0	3
G11	2	Quadratic	0.0000%	0	0	1	1
G12	3	Quadratic	4.774%	0	9 ³	0	0

is also utilized in migration and hybrid mutation operator to generate promising solutions. Third, the risk of trapping into stagnation brought by concentration of operations can be relieved by chaotic search. Consequently, the focus of operations can make the search of HBBO efficient.

4. Simulation Tests on Benchmark Functions

4.1. Parameter Setting and Statistical Results Obtained by HBBO. In order to validate the performance of the proposed HBBO on numerical optimization, twelve benchmark test functions are adopted. The selected benchmark problems propose a good challenge and measure for constrained optimization techniques. Main characteristics of the selected benchmark functions are shown in detail in Table 1 where

D is the dimensionality of a solution for test function, ρ represents the ratio of feasible region to search space, NI is the number of nonlinear inequality constraints, LI is the number of linear inequality constraints, NE is the number of nonlinear equality constraints, and a is the number of constraints active at the optimal solution. The ρ metric can be computed as the following:

$$\rho = \frac{|F|}{|S|}, \quad (15)$$

where $|S|$ is the number of solutions generated randomly ($|S| = 1,000,000$ in the paper), and $|F|$ is the number of feasible solutions found in all the solutions randomly generated. All the benchmark functions selected are depicted explicitly in Appendix A.

TABLE 2: Statistic results for twelve benchmark functions obtained by HBBO.

Function	Optimal	<i>Best</i>	<i>Mean</i>	<i>Median</i>	<i>Worst</i>	SD	FFE _s
G01	-15	-15	-14.799953	-15	-13	$6.10E - 1$	50,100
G02	-0.803619	-0.8036179	-0.7805965	-0.7852652	-0.7330360	$1.870E - 2$	75,200
G03	-1	-1.0050100	-1.0050100	-1.0050100	-1.0050100	$2.88E - 12$	150,100
G04	-30665.539	-30665.53867	-30665.5387	-30665.5387	-30665.5387	$1.11E - 11$	37,600
G05	5126.4981	5126.4842	5126.4842	5126.4842	5126.4842	$9.23E - 4$	375,100
G06	-6961.81388	-6961.81388	-6961.81388	-6961.81388	-6961.81388	$3.70E - 12$	50,100
G07	24.3062091	24.3062091	24.3062091	24.3062091	24.3062091	$6.49E - 10$	125,400
G08	-0.095825	-0.095825	-0.095825	-0.095825	-0.095825	$1.06E - 17$	10,100
G09	680.630057	680.630057	680.630057	680.630057	680.630057	$5.16E - 13$	75,100
G10	7049.248021	7049.248021	7049.248021	7049.248021	7049.248021	$1.08E - 7$	137,600
G11	0.75	0.74990	0.74990	0.74990	0.74990	$1.13E - 16$	75,100
G12	-1	-1	-1	-1	-1	0	12,600

For each test function, we performed 30 independent runs in matlab 7.0. The parameters of HBBO for experiments are set as follows: $E = I = 1$ is chosen as recommended in [8]; m_{\max} is set to be 0.8 which is much bigger than the corresponding value in basic BBO because big m_{\max} can improve mutation probabilities of individuals in population and enhance population diversity; based on the suggestions of mutation factor in DE in [3] and numerous experiments, $F_l = 0.75$ and $F_u = 0.15$ are chosen; $\eta = 20$ is chosen in the light of the effect of η on the search ability of SBX [24].

Through various tests, an appropriate set of population size NP for all the selected functions is found with which HBBO can present desirable performance. In the set found, population size NP for each benchmark function is given as the following: 200 for G02, 150 for G07, and 100 for the rest of benchmark functions. In each run, the maximum generations are given as the following: 200 for G01 and G06, 150 for G02 and G04, 600 for G03, 1500 for G05, 334 for G07, 40 for G08, 300 for G09 and G11, 550 for G10, and 50 for G12. In G03 and G05, the toleration value for equation constraint equals 0.001 as recommended in [19]; the toleration value for equation constraint of G11 is set to be 0.0001 as suggested in [26].

Table 2 summarizes the statistical features of results for twelve test functions obtained by HBBO and number of fitness function evaluations (FFE_s) required. From Table 2, we can see that HBBO can get optimal solution in all 30 runs for seven benchmark functions (G04, G06, G07, G08, G09, G10, and G12); for G01, HBBO can get the optimal solutions in some runs; the *best* results obtained by HBBO are very close to the known best solution for G02; for three benchmark functions (G03, G05, and G11), the results gained by HBBO are very close to the optimal solutions or the known best solution.

4.2. Comparison with Other State-of-the-Art Methods. In this part, the proposed approach HBBO is compared with other six state-of-the-art optimization technologies.

The following are the six state-of-the-art optimization technologies: conventional BBO with DE mutation technology (CBO-DM) [19], hybrid PSO with DE strategy (PSO-DE) [27], coevolutionary DE algorithm (CDE) [28], changing

range genetic algorithm (CRGA) [26], self-adaptive penalty function based algorithm (SAPF) [29], and simple multimembered evolution strategy (SMES) [30]. The statistic results of other six approaches are compared with that of HBBO in Table 3, which are gotten from the original references. The “NA” in tables of the paper indicates the results of compared algorithms are not available. It should be noted that the best results obtained by algorithms are marked in boldface in the following tables. As far as computational cost is concerned, CBO-DM, SAPF, CDE, SMES, respectively, need 350,000, 500,000, 248,000, and 240,000 FFE_s for all the test functions; PSO-DE needs 70,100 FFE_s for G04, 17,600 FFE_s for G12, and 140,100 FFE_s for the rest of test functions; 1,350 to 68,000 FFE_s are required for CRGA; the computational cost for CRGA is given in detail in [26].

With respect to CBO-DM, a variant of BBO, similar results are obtained by HBBO for five functions (G04, G06, G08, G09, and G12); in two functions (G07, G10), HBBO has better performance in the respect of considered metrics (*best*, *mean*, and *worst*); in G02, HBBO gets better *best* value with greater variability; the results of HBBO are obviously inferior but comparable for G01; the results obtained by HBBO are only lightly inferior for three test functions (G03, G05, and G11). In addition, the computational cost is far less than that of CBO-DM for all selected benchmark functions except G05. Consequently, HBBO is powerful competitor for CBO-DM on constrained optimization.

In contrast with other five state-of-the-art methods, the performance of HBBO is obviously inferior for function G01; HBBO can get better or similar solutions for the selected test functions except for G01, G03, and G11. In G03 and G11, the results obtained by HBBO are only lightly inferior to those of SMES. Furthermore, the computational cost is very competitive with respect to other methods for all selected test functions except G05.

5. Simulation Tests on Engineering Optimization Problems

In this part, four well-known engineering optimization problems are utilized to validate the performance of HBBO on

TABLE 3: Statistical features of results for twelve benchmark functions obtained by HBBO and other six state-of-the-art algorithms.

Function	Metrics	HBBO	CBO-DM	PSO-DE	CRGA	SAPF	SMES	CDE
G01	Best	-15	-15.000	15.000000	-14.9977	-15.000	-15.000	-15.0000
	Mean	-14.799953	-15.000	-15.000000	-14.9850	-14.552	-15.000	-15.0000
	Worst	-13	-15.000	-15.000000	-14.9467	-13.097	-15.000	-15.0000
G02	Best	-0.8036179	-0.803557	-0.8036145	-0.802959	-0.803202	-0.803601	-0.794669
	Mean	-0.7805965	-0.802774	-0.756678	-0.764494	-0.755798	-0.785238	-0.785480
	Worst	-0.7330360	-0.792576	-0.6367995	-0.722109	-0.745712	-0.751322	-0.779837
G03	Best	-1.0050100	-1.000	-1.0050100	-0.9997	-1.000	-1.000	NA
	Mean	-1.0050100	-1.000	-1.0050100	-0.9972	-0.964	-1.000	NA
	Worst	-1.0050100	-1.000	-1.0050100	-0.9931	-0.887	-1.000	NA
G04	Best	-30665.53867	-30665.539	-30665.539	-30665.520	-30665.401	-30665.539	-30665.539
	Mean	-30665.5387	-30665.539	-30665.539	-30664.398	-30665.922	-30665.539	-30665.536
	Worst	-30665.5387	-30665.539	-30665.539	-30660.313	-30656.471	-30665.539	-30665.509
G05	Best	5126.4842	5126.498	NA	5126.500	5126.907	5126.599	NA
	Mean	5126.4842	5126.498	NA	5507.041	5214.232	5174.492	NA
	Worst	5126.4842	5126.498	NA	6112.075	5564.642	5304.167	NA
G06	Best	-6961.81388	-6961.814	-6961.8139	-6956.251	-6961.046	-6961.814	-6961.814
	Mean	-6961.81388	-6961.814	-6961.8139	-6740.288	-6953.061	-6961.284	-6960.603
	Worst	-6961.81388	-6961.814	-6961.8139	-6077.123	-6943.304	-6952.482	-6901.285
G07	Best	24.3062091	24.326	24.3062091	24.882	24.838	24.327	NA
	Mean	24.3062091	24.345	24.306210	25.746	27.328	24.475	NA
	Worst	24.3062091	24.378	24.3062172	27.381	33.095	24.843	NA
G08	Best	-0.095825	-0.095825	-0.095826	-0.095825	-0.095825	-0.095825	NA
	Mean	-0.095825	-0.095825	-0.095826	-0.095819	-0.095635	-0.095825	NA
	Worst	-0.095825	-0.095825	-0.095826	-0.095808	-0.092697	-0.095825	NA
G09	Best	680.630057	680.630	680.63006	680.726	680.773	680.632	680.771
	Mean	680.630057	680.630	680.63006	681.347	681.246	680.643	681.503
	Worst	680.630057	680.630	680.63006	682.965	682.081	680.719	685.144
G10	Best	7049.248021	7059.802	7049.2480	7114.743	7069.981	7051.903	NA
	Mean	7049.248021	7075.832	7049.2480	8785.149	7238.964	7253.047	NA
	Worst	7049.248021	7098.254	7049.2482	10826.09	7489.406	7638.366	NA
G11	Best	0.74990	0.75	0.749999	0.750	0.749	0.75	NA
	Mean	0.74990	0.75	0.749999	0.752	0.751	0.75	NA
	Worst	0.74990	0.75	0.749999	0.757	0.757	0.75	NA
G12	Best	-1	-1.000000	1.000000	-1.000000	-1.000000	-1.000	-1.000000
	Mean	-1	-1.000000	-1.000000	-1.000000	-0.99994	-1.000	-1.000000
	Worst	-1	-1.000000	-1.000000	-1.000000	-0.999548	-1.000	-1.000000

solving real-world optimization problems. The four engineering optimization problems contain welded beam design problem, tension/compression spring design problem, speed reducer design problem, and three-bar truss design problem, which are listed in Appendix B. Parameters in HBBO for these four engineering optimization problems are as follows: population size and maximum generations are, respectively, 50 and 200 for welded beam design problem, 50 and 350 for tension/compression spring design problem, 100 and 100 for speed reducer design problem, and 50 and 60 for three-bar truss design problem; other parameters for HBBO are set in the same way as Section 4. For each engineering optimization problem, 30 independent runs are performed. Table 4 showed the statistic results for the four engineering optimization

problems solved by HBBO. We will evaluate performance of HBBO in respect of the quality of results and computational cost.

In order to demonstrate the superiority of HBBO, it is compared with other state-of-the-art algorithms on the four engineering problems. Welded beam and tension/compression spring design problems are also attempted by PSO-DE [27], CDE [28], coevolutionary particle swarm optimization (CPSO) [31], $(\mu + \lambda)$ -evolutionary strategy $((\mu + \lambda)$ -ES) [32], unified particle swarm optimization (UPSO) [33], and ABC [7]. PSO-DE [27], $(\mu + \lambda)$ -ES [32], and ABC [7] have also already performed on speed reducer design problem. PSO-DE [27] and Ray and Liew [34] have also been applied to solve three-bar truss design problem. The comparison of

TABLE 4: Statistic results for four engineering optimization problems solved by HBBO.

Engineering optimization problem	<i>Best</i>	<i>Mean</i>	<i>Median</i>	<i>Worst</i>	SD	FFEs
Welded beam design	1.724852309	1.724852309	1.724852309	1.724852309	1.14E – 15	25,050
Tension/compression spring design	0.012665233	0.012665393	0.012665234	0.012666698	3.18E – 07	43,800
Speed reducer design	2996.348165	2996.348165	2996.348165	2996.348165	5.52E – 12	25,100
Three-bar truss design	263.89584338	263.89584338	263.89584338	263.89584338	7.08E – 14	7,550

TABLE 5: Statistic results for welded beam design obtained by HBBO and other six state-of-the-art methods.

Method	<i>Best</i>	<i>Mean</i>	<i>Worst</i>	FFEs
HBBO	1.7248523	1.7248523	1.7248523	25,050
PSO-DE	1.7248531	1.7248579	1.7248811	33,000
CDE	1.733461	1.768158	1.824105	240,000
CPSO	1.728024	1.748831	1.782143	200,000
($\mu + \lambda$)-ES	1.724852	1.777692	NA	30,000
UPSO	1.92199	2.83721	NA	100,000
ABC	1.724852	1.741913	NA	30,000

TABLE 6: Statistic results of HBBO and other six state-of-the-art methods for tension/compression spring design.

Method	<i>Best</i>	<i>Mean</i>	<i>Worst</i>	FFEs
HBBO	0.012665233	0.012665393	0.012666698	43,800
PSO-DE	0.012665233	0.012665233	0.012665233	42,100
CDE	0.0126702	0.012703	0.012790	240,000
CPSO	0.0126747	0.01273	0.012924	200,000
($\mu + \lambda$)-ES	0.012689	0.013165	NA	30,000
UPSO	0.01312	0.02294	NA	100,000
ABC	0.012665	0.012709	NA	30,000

TABLE 7: Statistic results of HBBO and other six state-of-the-art methods for speed reducer design.

Method	<i>Best</i>	<i>Mean</i>	<i>Worst</i>	FFEs
HBBO	2996.348165	2996.348165	2996.348165	25,100
PSO-DE	2996.348165	2996.348165	2996.348166	70,100
($\mu + \lambda$)-ES	2996.348	2996.348	NA	30,000
ABC	2997.058	2997.058	NA	30,000

statistical results and computational cost for four engineering optimization problems between HBBO and other algorithms is shown in Tables 5, 6, 7, and 8.

From Tables 5, 6, 7, and 8, it can be seen that HBBO outperforms other compared algorithms for the given engineering optimization problems except tension/compression spring design problem for which PSO-DE has best performance. For tension/compression spring design problem, HBBO get similar *best* result and the *mean* and *worst* results of it are just lightly inferior in contrast with PSO-DE.

TABLE 8: Statistic results of HBBO and other six state-of-the-art methods for three-bar truss design.

Method	<i>Best</i>	<i>Mean</i>	<i>Worst</i>	FFEs
HBBO	263.89584338	263.89584338	263.89584338	7,550
PSO-DE	263.89584338	263.89584338	263.89584338	17,600
Ray and Liew	263.89584654	263.90335672	263.96975638	17,610

6. Discussions

In this part, HBBO is compared with the original BBO and self-adapting DE (SADE) to demonstrate the searching efficiency of HBBO further. In addition, the influence of maximum mutation rate on searching efficiency of HBBO is investigated.

6.1. Comparison with the Original BBO and SADE. The detail of the original BBO can be gotten from [8] and SADE proposed in [25] is compared. Specific parameter setting of these two algorithms is the same as the original references while the parameters related to test functions such as population size and constraint tolerance are in accordance with Section 4. Deb's feasible rule is applied in BBO and SADE to handle constraint. Here, HBBO is adopted in an identical way as described in Section 4.

Figure 1 illustrates typical evolution processes of objective function value of best solution in population when four benchmark functions (G02, G03, G07, and G09) are, respectively, solved by HBBO, BBO, and SADE. From Figure 1, it can be seen that HBBO have fastest convergence speed while BBO is often trapped into stagnation. So it can be concluded that the exploration and exploitation of BBO are well enhanced and balanced by new mutation operator and further hybridization with DE and chaotic search.

6.2. Influence of Maximum Mutation Rate on HBBO. The maximum value of mutation rate m_{\max} is related to the probability that individuals mutate by new hybrid mutation operator so that it affects the balance degree of exploration and exploitation of HBBO. In order to investigate the effect of m_{\max} on search efficiency of HBBO, m_{\max} is set to be different values including 0.05, 0.1, 0.4, 0.8, and 1. The investigation experiments are performed on five benchmark functions (G01, G02, G03, G07, and G10).

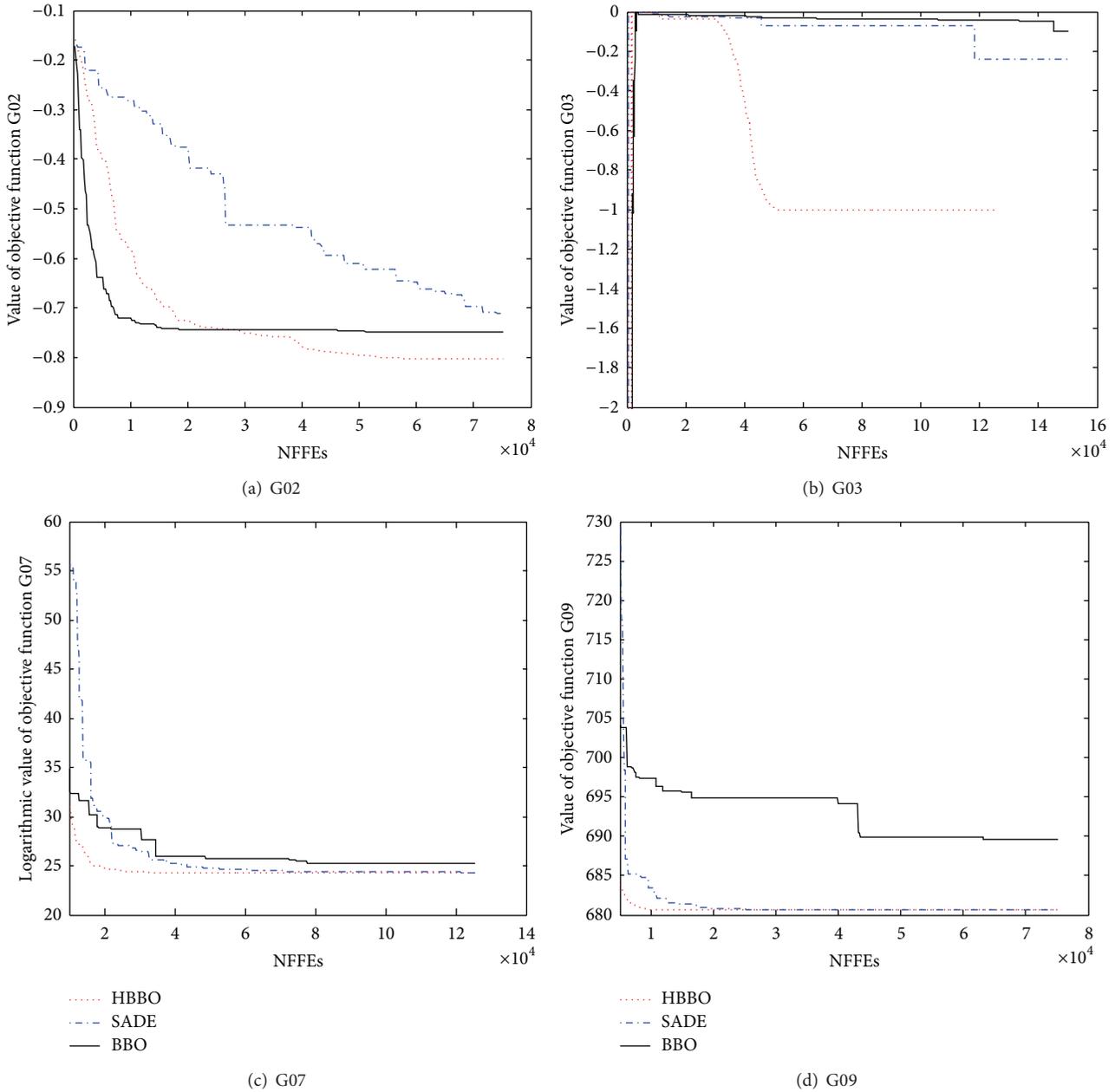


FIGURE 1: Objective function value curves of four test functions solved by HBBO, SADE, and BBO.

The other parameters are in accordance with description in Section 4. For each value of m_{max} and test function, we perform 30 independent runs. The statistical features of results obtained by HBBO with different m_{max} are summarized in Table 9.

From Table 9, we can see that the value of m_{max} has significant influence on search efficiency of HBBO. The influence can be stated from three respects as the following. First, HBBO with different value of m_{max} has different performance on the five test functions. Second, HBBO with too small or too big m_{max} could not solve the selected test functions well while HBBO with middle value of m_{max} has better comprehensive performance. Third, the fittest m_{max} for each test function is different. The influence above may

be explained as follows. Too small m_{max} cannot balance the exploitation of BBO well while too big m_{max} will destroy the exploitation of BBO; different test functions have different characteristics so that they have different requirements of exploration and exploitation of optimization algorithm.

7. Conclusions

The paper proposes a new hybrid biogeography based optimization (HBBO) for constrained optimization. For the presented algorithm HBBO, a new mutation operator was proposed to generate promising solutions by merging DE mutation with SBX; a half of the population also evolved by two mutation strategies of DE. Chaotic search was introduced

TABLE 9: Statistical features of the results obtained by HBBO with different m_{\max} .

Function	Metrics	HBBO				
		$m_{\max} = 0.05$	$m_{\max} = 0.1$	$m_{\max} = 0.4$	$m_{\max} = 0.8$	$m_{\max} = 1$
G01	<i>Best</i>	-15	-15	-15	-15	-15
	<i>Mean</i>	-14.666665	-14.633610	-14.600000	-14.799953	-14.632939
	<i>Worst</i>	-12.999949	-12.008289	-13	-13	-12
	SD	$7.58E - 1$	$8.49E - 1$	$8.14E - 1$	$6.10E - 1$	$8.50E - 1$
G02	<i>Best</i>	-0.8036071	-0.8036171	-0.8036185	-0.8036179	-0.8036173
	<i>Mean</i>	-0.7780832	-0.7787019	-0.7799297	-0.7805965	-0.7775819
	<i>Worst</i>	-0.71807066	-0.7501355	-0.7248475	-0.7330360	-0.7225584
	SD	$2.03E - 2$	$1.58E - 2$	$1.83E - 2$	$1.87E - 2$	$2.19E - 2$
G03	<i>Best</i>	-1.0044781	-1.0050100	1.0050100	1.0050100	-1.0050100
	<i>Mean</i>	-0.7515293	-1.0050100	1.0050100	1.0050100	-1.0050100
	<i>Worst</i>	-0.2971747	-1.0050100	1.0050100	1.0050100	-1.0050100
	SD	$2.3E - 1$	$2.53E - 10$	$4.10E - 10$	$2.88E - 12$	$5.39E - 12$
G07	<i>Best</i>	24.3062091	24.3062091	24.3062091	24.3062091	24.3062091
	<i>Mean</i>	24.3062100	24.3062091	24.3062091	24.3062091	24.3062091
	<i>Worst</i>	24.3062351	24.3062091	24.3062091	24.3062091	24.3062091
	SD	$4.74E - 6$	$7.00E - 10$	$2.35E - 10$	$6.49E - 10$	$7.97E - 10$
G10	<i>Best</i>	7049.248021	7049.248021	7049.248021	7049.248021	7049.248021
	<i>Mean</i>	7054.178611	7049.248022	7049.248021	7049.248021	7049.248027
	<i>Worst</i>	7194.608835	7049.248076	7049.248021	7049.248021	7049.248200
	SD	26.52	$1.02E - 5$	$4.20E - 8$	$1.08E - 7$	$3.27E - 5$

for escape from stagnation and Deb's feasibility-based rule was applied to handle constraints. Furthermore, self-adaption mechanism for the mutation scaling factor of DE was utilized to avoid bothering of choosing an appropriate parameter.

Simulation experiments were performed on twelve benchmark test functions and four well-known engineering optimization problems. HBBO can obtain better or comparable results in contrast with other state-of-the-art optimization technologies. At the same time, the low computation cost is the obvious advantage of our HBBO. In short, HBBO is an effective and efficient method for constrained optimization. In addition, the influence of maximum mutation rate was investigated and the results demonstrate HBBO with maximum mutation rate of middle value has better comprehensive performance.

Maximum immigration rate I and emigration rate E affect the migration probability and the prior probability of existence for individuals in population. How to select the fittest E and I for HBBO will be one possible focus of our research in the future. Besides, HBBO will be extended to multiobjectives optimization.

Appendices

A. Benchmark Test Functions

A.1. G01.

$$\begin{aligned} \text{Minimize} \quad & f(\vec{x}) = 5 \sum_{i=1}^4 x_i - 5 \sum_{i=1}^4 x_i^2 - \sum_{i=5}^{13} x_i \\ \text{subject to} \quad & g_1(\vec{x}) = 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \leq 0, \\ & g_2(\vec{x}) = 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \leq 0, \end{aligned}$$

$$\begin{aligned} g_3(\vec{x}) &= 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \leq 0, \\ g_4(\vec{x}) &= -8x_1 + x_{10} \leq 0, \\ g_5(\vec{x}) &= -8x_2 + x_{11} \leq 0, \\ g_6(\vec{x}) &= -8x_3 + x_{12} \leq 0, \\ g_7(\vec{x}) &= -2x_4 - x_5 + x_{10} \leq 0, \\ g_8(\vec{x}) &= -2x_6 - x_7 + x_{11} \leq 0, \\ g_9(\vec{x}) &= -2x_8 - x_9 + x_{12} \leq 0, \end{aligned} \quad (\text{A.1})$$

where the bounds are $0 \leq x_i \leq 1$ ($i = 1, \dots, 9$), $0 \leq x_i \leq 100$ ($i = 10, 11, 12$), and $0 \leq x_{13} \leq 1$. The global optimum is at $\vec{x}^* = (1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 1)$ with

$$f(\vec{x}^*) = -15. \quad (\text{A.2})$$

A.2. G02.

$$\begin{aligned} \text{Minimize} \quad & f(\vec{x}) = - \left| \frac{\sum_{i=1}^n \cos^4(x_i) - 2 \prod_{i=1}^n \cos^2(x_i)}{\sqrt{\sum_{i=1}^n ix_i^2}} \right| \\ \text{subject to} \quad & g_1(\vec{x}) = 0.75 - \prod_{i=1}^n x_i \leq 0, \\ & g_2(\vec{x}) = \sum_{i=1}^n x_i - 7.5n \leq 0, \end{aligned} \quad (\text{A.3})$$

where $n = 20$ and $0 \leq x_i \leq 10$ ($i = 1, \dots, n$). The global optimum is unknown; the best objective function value reported is $f(\vec{x}^*) = -0.803619$.

A.3. G03.

$$\begin{aligned} \text{Maximize } f(\vec{x}) &= (\sqrt{n})^n \prod_{i=1}^n x_i \\ \text{subject to } h(\vec{x}) &= \sum_{i=1}^n x_i^2 - 1 = 0, \end{aligned} \quad (\text{A.4})$$

where $n = 10$ and $0 \leq x_i \leq 1$ ($i = 1, 2, \dots, n$). The optimum solution is $x_i^* = 1/\sqrt{n}$ ($i = 1, 2, \dots, n$) with $f(\vec{x}^*) = 1$.

A.4. G04.

$$\begin{aligned} \text{Maximize } f(\vec{x}) &= 5.3578547x_3^2 + 0.8356891x_1x_5 \\ &\quad + 37.293239x_1 - 40792.141 \\ \text{subject to } g_1(\vec{x}) &= 85.334407 + 0.0056858x_2x_5 \\ &\quad + 0.0006262x_1x_4 \\ &\quad - 0.0022053x_3x_5 - 92 \leq 0, \\ g_2(\vec{x}) &= -85.334407 - 0.0056858x_2x_5 \\ &\quad - 0.0006262x_1x_4 \\ &\quad + 0.0022053x_3x_5 \leq 0, \\ g_3(\vec{x}) &= 80.51249 + 0.0071317x_2x_5 \\ &\quad + 0.0029955x_1x_2 \\ &\quad + 0.0021813x_3^2 - 110 \leq 0, \\ g_4(\vec{x}) &= -80.51249 - 0.0071317x_2x_5 \\ &\quad - 0.0029955x_1x_2 \\ &\quad - 0.0021813x_3^2 + 90 \leq 0, \\ g_5(\vec{x}) &= 9.300961 + 0.0047026x_3x_5 \\ &\quad + 0.0012547x_1x_3 \\ &\quad + 0.0019085x_3x_4 - 25 \leq 0, \\ g_6(\vec{x}) &= -9.300961 - 0.0047026x_3x_5 \\ &\quad - 0.0012547x_1x_3 \\ &\quad - 0.0019085x_3x_4 + 20 \leq 0, \end{aligned} \quad (\text{A.5})$$

where $78 \leq x_1 \leq 102$, $33 \leq x_2 \leq 45$, and $27 \leq x_i \leq 45$ ($i = 3, 4, 5$). The optimum solution is $\vec{x}^* = (78, 33, 29.995256025682, 45, 36.775812905788)$ with $f(\vec{x}^*) = 30665.539$.

A.5. G05.

$$\begin{aligned} \text{Minimize } f(\vec{x}) &= 3x_1 + 0.000001x_1^3 + 2x_2 \\ &\quad + \left(\frac{0.000002}{3}\right)x_2^3 \\ \text{subject to } g_1(\vec{x}) &= -x_4 + x_3 - 0.55 \leq 0, \\ g_2(\vec{x}) &= -x_3 + x_4 - 0.55 \leq 0, \\ h_3(\vec{x}) &= 1000 \sin(-x_3 - 0.25) \\ &\quad + 1000 \sin(-x_4 - 0.25) + 894.8 \\ &\quad - x_1 = 0, \\ h_4(\vec{x}) &= 1000 \sin(x_3 - 0.25) \\ &\quad + 1000 \sin(x_3 - x_4 - 0.25) + 894.8 \\ &\quad - x_2 = 0, \\ h_5(\vec{x}) &= 1000 \sin(x_4 - 0.25) \\ &\quad + 1000 \sin(x_4 - x_3 - 0.25) \\ &\quad + 1294.8 = 0, \end{aligned} \quad (\text{A.6})$$

where $0 \leq x_1 \leq 1200$, $0 \leq x_2 \leq 1200$, $-0.55 \leq x_3 \leq 0.55$, and $-0.55 \leq x_4 \leq 0.55$. The best known solution is $\vec{x}^* = (679.9453, 1026.067, 0.1188764, 0.3962336)$, where $f(\vec{x}^*) = 5126.4981$.

A.6. G06.

$$\begin{aligned} \text{Minimize } f(\vec{x}) &= (x_1 - 10)^3 + (x_2 - 20)^3 \\ \text{subject to } g_1(\vec{x}) &= -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \leq 0, \\ g_2(\vec{x}) &= -(x_1 - 6)^2 - (x_2 - 5)^2 - 82.81 \leq 0, \end{aligned} \quad (\text{A.7})$$

where $13 \leq x_1 \leq 100$ and $0 \leq x_2 \leq 100$. The optimum solution is $\vec{x}^* = (14.095, 0.84296)$ with $f(\vec{x}^*) = -6961.81388$.

A.7. G07.

$$\begin{aligned} \text{Minimize } f(\vec{x}) &= x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 \\ &\quad + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 \\ &\quad + 2(x_6 - 1)^2 + 5x_7^2 \\ &\quad + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 \\ &\quad + (x_{10} - 7)^2 + 45 \end{aligned}$$

subject to $g_1(\vec{x}) = -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \leq 0$,
 $g_2(\vec{x}) = 10x_1 - 8x_2 - 17x_7 + 2x_8 \leq 0$,
 $g_3(\vec{x}) = -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \leq 0$,
 $g_4(\vec{x}) = 3(x_1 - 2)^2 + 4(x_2 - 3)^2$
 $+ 2x_3^2 - 7x_4 - 120 \leq 0$,
 $g_5(\vec{x}) = 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \leq 0$,
 $g_6(\vec{x}) = x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5$
 $- 6x_6 \leq 0$,
 $g_7(\vec{x}) = 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2$
 $+ 3x_5^2 - x_6 - 30 \leq 0$,
 $g_8(\vec{x}) = -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \leq 0$,
(A.8)

where $-10 \leq x_i \leq 10$ ($i = 1, 2, \dots, 10$). The optimum solution is $\vec{x}^* = (2.171996, 2.363683, 8.773926, 5.095984, 0.9906548, 1.430574, 1.321644, 9.828726, 8.280092, 8.375927)$ with $f(\vec{x}^*) = 24.3062091$.

A.8. G08.

Maximize $f(\vec{x}) = \frac{\sin^3(2\pi x_1) \sin(2\pi x_2)}{x_1^3(x_1 + x_2)}$
subject to $g_1(\vec{x}) = x_1^2 - x_2 + 1 \leq 0$,
 $g_2(\vec{x}) = 1 - x_1 + (x_2 - 4)^2 \leq 0$,
(A.9)

where $0 \leq x_1 \leq 10$ and $0 \leq x_2 \leq 10$. The optimal solution is $\vec{x}^* = (1.2279713, 4.2453733)$ where $f(\vec{x}^*) = 0.095825$.

A.9. G09.

Minimize $f(\vec{x}) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4$
 $+ 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4$
 $- 4x_6x_7 - 10x_6 - 8x_7$
subject to $g_1(\vec{x}) = -127 + 2x_1^2 + 3x_2^4 + x_3$
 $+ 4x_4^2 + 5x_5 \leq 0$,
 $g_2(\vec{x}) = -282 + 7x_1 + 3x_2 + 10x_3^2$
 $+ x_4 - x_5 \leq 0$,
 $g_3(\vec{x}) = -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \leq 0$,
 $g_4(\vec{x}) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2$
 $+ 5x_6 - 11x_7 \leq 0$,
(A.10)

where $-10 \leq x_i \leq 10$ ($i = 1, 2, \dots, 7$). The optimum solution is $\vec{x}^* = (2.330499, 1.951372, -0.4775414, 4.365726, -0.6244870, 1.038131, 1.594227)$ with $f(\vec{x}^*) = 680.630057$.

A.10. G10.

Minimize $f(\vec{x}) = x_1 + x_2 + x_3$
subject to $g_1(\vec{x}) = -1 + 0.0025(x_4 + x_6) \leq 0$,
 $g_2(\vec{x}) = -1 + 0.0025(x_5 + x_7 - x_4) \leq 0$,
 $g_3(\vec{x}) = -1 + 0.01(x_8 - x_5) \leq 0$,
 $g_4(\vec{x}) = -x_1x_6 + 833.33252x_4 + 100x_1$
 $- 83333.333 \leq 0$,
 $g_5(\vec{x}) = -x_2x_7 + 1250x_5 + x_2x_4$
 $- 1250x_4 \leq 0$,
 $g_6(\vec{x}) = -x_3x_8 + 1250000 + x_3x_5$
 $- 2500x_5 \leq 0$,
(A.11)

where $100 \leq x_1 \leq 10,000$, $1000 \leq x_i \leq 10,000$ ($i = 2, 3$), and $100 \leq x_i \leq 10,000$ ($i = 4, 5, \dots, 8$). The optimum solution is $\vec{x}^* = (579.3066, 1359.9707, 5109.9707, 182.0177, 295.601, 217.928, 286.165, 395.6012)$ with $f(\vec{x}^*) = 7049.248021$.

A.11. G11.

Minimize $f(\vec{x}) = x_1^2 + (x_2 - 1)^2$
subject to $h(\vec{x}) = x_2 - x_1^2 = 0$,
(A.12)

where $-1 \leq x_1 \leq 1$ and $-1 \leq x_2 \leq 1$. The optimum solution is $\vec{x}^* = (\pm 1/\sqrt{2}, 1/2)$ with $f(\vec{x}^*) = 0.75$.

A.12. G12.

Maximize $f(\vec{x}) = (100 - (x_1 - 5)^2 - (x_2 - 5)^2)$
 $- (x_3 - 5)^2 \times (100)^{-1}$
subject to $g(\vec{x}) = (x_1 - p)^2 + (x_2 - q)^2 + (x_3 - r)^2$
 $- 0.0625 \leq 0$,
(A.13)

where $0 \leq x_i \leq 10$ ($i = 1, 2, 3$) and $p, q, r = 1, 2, \dots, 9$. A point (x_1, x_2, x_3) is feasible if and only if there exist p, q, r such that the above inequality holds. The optimum solution is $\vec{x}^* = (5, 5, 5)$ with $f(\vec{x}^*) = 1$.

B. Engineering Design Problems

B.1. Welded Beam Design Problem. A welded beam is designed for the minimum cost subject to constraints on shear stress (τ); bending stress in the beam (θ); buckling load

on the bar (P_c); end deflection of the beam (δ); and side constraints. There are four design variables $h(x_1)$, $l(x_2)$, $t(x_3)$, and $b(x_4)$.

$$\begin{aligned}
 \text{Minimize } & f(\vec{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2) \\
 \text{subject to } & g_1(\vec{x}) = \tau(\vec{x}) - \tau_{\max} \leq 0, \\
 & g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{\max} \leq 0, \\
 & g_3(\vec{x}) = x_1 - x_4 \leq 0, \\
 & g_4(\vec{x}) = 0.1047x_1^2 + 0.04811x_3x_4(14 + x_2) \\
 & \quad - 5.0 \leq 0, \\
 & g_5(\vec{x}) = 0.125 - x_1 \leq 0, \\
 & g_6(\vec{x}) = \delta(\vec{x}) - \delta_{\max} \leq 0, \\
 & g_7(\vec{x}) = P - P_c(\vec{x}) \leq 0,
 \end{aligned} \tag{B.1}$$

where $\tau(\vec{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''(x_2/2R) + (\tau'')^2}$, $\tau' = P/\sqrt{2}x_1x_2$, $\tau'' = MR/J$, $M = P(L + x_2/2)$, $R = \sqrt{x_2^2/4 + ((x_1 + x_3)/2)^2}$, $\delta(\vec{x}) = 4PL^3/Ex_3^3x_4$, $J = 2\{\sqrt{2}x_1x_2[x_2^2/12 + ((x_1 + x_3)/2)^2]\}$, $\sigma(\vec{x}) = 6PL/x_4x_3^2$, $P_c(\vec{x}) = (4.013E\sqrt{x_3^2x_4^6/36/L^2})(1 - (x_3/2L)\sqrt{E/4G})$, $P = 6000$ lb, $L = 14$ in, $E = 30 \times 10^6$ psi, $G = 12 \times 10^6$ psi, $\tau_{\max} = 13,600$ psi, $\sigma_{\max} = 30,000$ psi, $\delta_{\max} = 0.25$ in, $0.1 \leq x_1 \leq 2$, $0.1 \leq x_2 \leq 10$, $0.1 \leq x_3 \leq 10$, $0.1 \leq x_4 \leq 2$.

B.2. Tension/Compression String Design Problem. In this problem, the objective is to minimize the weight of a tension/compression spring subject to constraints on minimum deflection, shear stress, surge frequency, and limits on outside diameter and on design variables. The design variables are the mean coil diameter $D(x_2)$; the wire diameter $d(x_1)$; and the number of active coils $P(x_3)$.

$$\begin{aligned}
 \text{Minimize } & f(\vec{x}) = (x_3 + 2)x_2x_1^2 \\
 \text{subject to } & g_1(\vec{x}) = 1 - \frac{x_2^3x_3}{71785x_1^4} \leq 0, \\
 & g_2(\vec{x}) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0, \\
 & g_3(\vec{x}) = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0, \\
 & g_4(\vec{x}) = \frac{x_1 + x_2}{1.5} - 1 \leq 0,
 \end{aligned} \tag{B.2}$$

where $0.05 \leq x_1 \leq 2$, $0.25 \leq x_2 \leq 1.3$, and $2 \leq x_3 \leq 15$.

B.3. Speed Reducer Design Problem.

$$\begin{aligned}
 \text{Minimize } & f(\vec{x}) = 0.7854x_1x_2^2 \\
 & \quad \times (3.3333x_3^2 + 14.9334x_3 - 43.0934) \\
 & \quad - 1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3) \\
 & \quad + 0.7854(x_4x_6^2 + x_5x_7^2) \\
 \text{subject to } & g_1(\vec{x}) = \frac{27}{x_1x_2^2x_3} - 1 \leq 0, \\
 & g_2(\vec{x}) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \leq 0, \\
 & g_3(\vec{x}) = \frac{1.93x_4^3}{x_2x_6^4x_3} - 1 \leq 0, \\
 & g_4(\vec{x}) = \frac{1.93x_5^3}{x_2x_7^4x_3} - 1 \leq 0, \\
 & g_5(\vec{x}) = \frac{[(745x_4/x_2x_3)^2 + 16.9 \times 10^6]^{1/2}}{110x_6^3} \\
 & \quad - 1 \leq 0, \\
 & g_6(\vec{x}) = \frac{[(745x_5/x_2x_3)^2 + 157.5 \times 10^6]^{1/2}}{85x_7^3} \\
 & \quad - 1 \leq 0, \\
 & g_7(\vec{x}) = \frac{x_2x_3}{40} - 1 \leq 0, \\
 & g_8(\vec{x}) = \frac{5x_2}{x_1} - 1 \leq 0, \\
 & g_9(\vec{x}) = \frac{x_1}{12x_2} - 1 \leq 0, \\
 & g_{10}(\vec{x}) = \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0, \\
 & g_{11}(\vec{x}) = \frac{1.1x_7 + 1.9}{x_5} - 1 \leq 0,
 \end{aligned} \tag{B.3}$$

where $2.6 \leq x_1 \leq 3.6$, $0.7 \leq x_2 \leq 0.8$, $17 \leq x_3 \leq 28$, $7.3 \leq x_4 \leq 8.3$, $7.3 \leq x_5 \leq 8.3$, $2.9 \leq x_6 \leq 3.9$, and $5.0 \leq x_7 \leq 5.5$.

B.4. Three-Bar Truss Design Problem.

$$\begin{aligned}
 \text{Minimize } & f(\vec{x}) = (2\sqrt{2}x_1 + x_2) \times l \\
 \text{subject to } & g_1(\vec{x}) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leq 0,
 \end{aligned}$$

$$g_2(\vec{x}) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0,$$

$$g_3(\vec{x}) = \frac{1}{\sqrt{2}x_2 + x_1}P - \sigma \leq 0,$$

(B.4)

where $0 \leq x_1 \leq 1$ and $0 \leq x_2 \leq 1$; $l = 100$ cm, $P = 2$ kN/cm², and $\sigma = 2$ kN/cm².

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research is supported by the Specialized Research Fund for the Doctoral Program of Higher Education of China under Grant no. 20120036130001, the Fundamental Research Funds for the Central Universities of China under Grant no. 2014MS93, and the Independent Research Funds of State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources of China under Grant no. 201414.

References

- [1] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, Mass, USA, 1989.
- [2] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE Conference on Neural Networks*, vol. 4, pp. 1942–1948, November–December 1995.
- [3] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [4] R. Storn, "System design by constraint adaptation and differential evolution," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 1, pp. 22–34, 1999.
- [5] C. Blum, "Ant colony optimization: Introduction and recent trends," *Physics of Life Reviews*, vol. 2, no. 4, pp. 353–373, 2005.
- [6] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.
- [7] B. Akay and D. Karaboga, "Artificial bee colony algorithm for large-scale problems and engineering design optimization," *Journal of Intelligent Manufacturing*, vol. 23, no. 4, pp. 1001–1014, 2012.
- [8] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.
- [9] D. Simon, R. Rarick, M. Ergezer, and D. Du, "Analytical and numerical comparisons of biogeography-based optimization and genetic algorithms," *Information Sciences*, vol. 181, no. 7, pp. 1224–1248, 2011.
- [10] M. M. Sayed, M. S. Saad, H. M. Emara, and E. E. Abou El-Zahab, "A novel method for PID tuning using a modified biogeography-based optimization algorithm," in *Proceedings of the 24th Chinese Control and Decision Conference (CCDC '12)*, pp. 1642–1647, Taiyuan, China, May 2012.
- [11] L. Wang and Y. Xu, "An effective hybrid biogeography-based optimization algorithm for parameter estimation of chaotic systems," *Expert Systems with Applications*, vol. 38, no. 12, pp. 15103–15109, 2011.
- [12] D. Du and D. Simon, "Complex system optimization using biogeography-based optimization," *Mathematical Problems in Engineering*, vol. 2013, Article ID 456232, 17 pages, 2013.
- [13] V. K. Panchal, P. Singh, N. Kaur, and H. Kundra, "Biogeography based satellite image classification," *International Journal of Computer Science and Information Security*, vol. 6, no. 2, pp. 269–274, 2009.
- [14] H. Ma, "An analysis of the equilibrium of migration models for biogeography-based optimization," *Information Sciences*, vol. 180, no. 18, pp. 3444–3464, 2010.
- [15] W. Gong, Z. Cai, and C. X. Ling, "DE/BBO: a hybrid differential evolution with biogeography-based optimization for global numerical optimization," *Soft Computing*, vol. 15, no. 4, pp. 645–665, 2010.
- [16] H. Ma and D. Simon, "Blended biogeography-based optimization for constrained optimization," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 3, pp. 517–525, 2011.
- [17] X. Li and M. Yin, "Multi-operator based biogeography based optimization with mutation for global numerical optimization," *Computers and Mathematics with Applications*, vol. 64, no. 9, pp. 2833–2844, 2012.
- [18] X. Li, J. Wang, J. Zhou, and M. Yin, "A perturb biogeography based optimization with mutation for global numerical optimization," *Applied Mathematics and Computation*, vol. 218, no. 2, pp. 598–609, 2011.
- [19] I. Boussaid, A. Chatterjee, P. Siarry, and M. Ahmed-Nacer, "Biogeography-based optimization for constrained optimization problems," *Computers and Operations Research*, vol. 39, no. 12, pp. 3293–3304, 2012.
- [20] G. Xiong, D. Shi, and X. Duan, "Enhancing the performance of biogeography-based optimization using polyphyletic migration operator and orthogonal learning," *Computers and Operations Research*, vol. 41, pp. 125–139, 2014.
- [21] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2–4, pp. 311–338, 2000.
- [22] W. Guo, L. Wang, and Q. Wu, "An analysis of the migration rates for biogeography-based optimization," *Information Sciences*, vol. 254, pp. 111–140, 2014.
- [23] D. Simon, "The Matlab code of biogeography-based optimization," 2008, <http://academic.csuohio.edu/simond/bbo/>.
- [24] M. M. Raghuwanshi and O. G. Kakde, "Survey on multiobjective evolutionary and real coded genetic algorithms," in *Proceedings of the 8th Asia Pacific Symposium on Intelligent and Evolutionary Systems*, pp. 150–161, Cairns, Australia, December 2004.
- [25] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, 2006.
- [26] A. Amirjanov, "The development of a changing range genetic algorithm," *Computer Methods in Applied Mechanics and Engineering*, vol. 195, no. 19, pp. 2495–2508, 2006.

- [27] H. Liu, Z. Cai, and Y. Wang, "Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization," *Applied Soft Computing Journal*, vol. 10, no. 2, pp. 629–640, 2010.
- [28] F. Huang, L. Wang, and Q. He, "An effective co-evolutionary differential evolution for constrained optimization," *Applied Mathematics and Computation*, vol. 186, no. 1, pp. 340–356, 2007.
- [29] B. Tessema and G. G. Yen, "A self adaptive penalty function based algorithm for constrained optimization," in *Proceeding of the IEEE Congress on Evolutionary Computation (CEC '06)*, pp. 246–253, Vancouver, Canada, July 2006.
- [30] E. Mezura-Montes and C. C. Coello, "A simple multimembered evolution strategy to solve constrained optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 1, pp. 1–17, 2005.
- [31] Q. He and L. Wang, "An effective co-evolutionary particle swarm optimization for constrained engineering design problems," *Engineering Applications of Artificial Intelligence*, vol. 20, no. 1, pp. 89–99, 2007.
- [32] M. E. Mezura and C. C. Coello, "Useful infeasible solutions in engineering optimization with evolutionary algorithms," in *Proceedings of the 4th Mexican International Conference on Artificial Intelligence*, pp. 625–662, Monterrey, Mexico, November 2005.
- [33] K. E. Parsopoulos and M. N. Vrahatis, "Unified Particle Swarm Optimization for solving constrained engineering optimization problems," in *Proceedings of the International Conference on Natural Computation (ICNC '05)*, pp. 582–591, Changsha, China, August 2005.
- [34] T. Ray and K. M. Liew, "Society and civilization: an optimization algorithm based on the simulation of social behavior," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 4, pp. 386–396, 2003.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

