4-1-2014

Hybrid biogeography-based evolutionary algorithms

Haiping Ma
Shaoxing University, mahp@usx.edu.cn

Dan Simon
Cleveland State University, d.j.simon@csuohio.edu

Minrui Fei
Shanghai University, mrfei@staff.shu.edu.cn

Xinzhan Shu
Zhejiang A & F University, qiacoshu@163.com

Zixiang Chen
Shaoxing University

Hybrid biogeography-based evolutionary algorithms

Haiping Ma [a,c,*], Dan Simon [b], Minrui Fei [c], Xinzhan Shu [d], Zixiang Chen [a]

[a] Department of Physics and Electrical Engineering, Shaoxing University, Shaoxing, Zhejiang, China
[b] Department of Electrical and Computer Engineering, Cleveland State University, Cleveland, OH, USA
[c] Shanghai Key Laboratory of Power Station Automation Technology, School of Mechatronic Engineering and Automation, Shanghai University, Shanghai, China
[d] Department of Computer, School of Information Engineering, Zhejiang A & F University, Hangzhou, Zhejiang, China

1. Introduction

Evolutionary algorithms (EAs) are optimization techniques that have become highly popular in recent decades (Simon, 2013; Yao et al., 1999; Wolpert and Macready, 1997; Neri and Cotta, 2012). One of the main reasons for their success is that they provide a general purpose mechanism to solve a wide range of problems. Many EAs have been proposed, including genetic algorithms (GA) (Ahn, 2006), evolution strategies (ES) (Beyer, 1994; Beyer and Sendhoff, 2008), ant colony optimization (ACO) (Dorigo et al., 2002; Dorigo and Gambardella, 1997), particle swarm optimization (PSO) (Bratton and Kennedy, 2007), differential evolution (DE) (Das and Suganthan, 2011), estimation of distribution algorithms (EDA) (Pedro and Lozano, 2002), immune system optimization (Hofmeyr and Forrest, 2000), artificial bee colony (ABC) optimization (Karaboga and Basturk, 2007), and many others (Simon, 2013).

Hybrid EAs are attractive alternatives to standard EAs. The combination of several algorithms in hybrid EAs allows them to exploit the strength of each algorithm. It has been shown that by properly selecting the constituent algorithms and hybridization strategies, hybrid EAs can outperform their constituent algorithms due to their synergy (Niknam and Farsani, 2010). This characteristic is a strong motivation for the study of hybrid EAs. Many hybrid EAs have been proposed to improve performance and to find global optima (Makeyev et al., 2010; Mongus et al., 2012). Although some of these improvements are significant, the development of new hybrid EA strategies is worthy of further investigation.

Current research directions in hybrid EAs involve several major areas. The first area is the application of hybrid EAs to special types of optimization problems, such as constrained optimization (Wang et al., 2009) and multi-objective optimization (Niknam, 2009). The second area is the application of hybrid EAs to real-world optimization problems (Liang et al., 2009; Lin et al., 2009). The third area is the determination of which EAs to combine in a hybrid algorithm (Lozano and Garcia-Martinez, 2010; Blum et al., 2011). The fourth area is the determination of how to hybridize a given set of EAs into a single algorithm (Gen and Lin, in press; Prodhon, 2011); that is, how to determine the hybridization strategy. The goal of this paper is to address the fourth research area. In general, we propose the application of EA information-sharing ideas to the hybridization of constituent EAs. That is, just as a single EA uses specific mechanisms to share information among candidate solutions, we use the same mechanisms to share information among constituent EAs in a hybrid EA. The information-sharing mechanism that we propose in this paper is based on biogeography-based optimization (BBO).

* Corresponding author at: Department of Physics and Electrical Engineering, Shaoxing University, Shaoxing, Zhejiang, China. Tel.: +86 575 8834 5673.
E-mail addresses: Mahp@usx.edu.cn (H. Ma), d.j.simon@csuohio.edu (D. Simon), mrfei@staff.shu.edu.cn (M. Fei), qiacoshu@163.com (X. Shu).

BBO is an EA that was introduced in 2008 (Simon, 2008, 2011). It is modeled after the immigration and emigration of species between habitats. One distinctive feature of BBO is that in each generation, BBO uses the fitness of each candidate solution to determine the candidate solution's immigration and emigration rate. The emigration rate increases with fitness, and the immigration rate decreases with fitness. BBO has demonstrated good performance on benchmark functions (Ma, 2010; Boussaïd et al., 2012). It has also been applied to many real-world optimization problems, including economic load dispatch (Bhattacharya and Chattopadhyay, 2010), wireless network power allocation (Boussaïd et al., 2011, 2013a), flexible job shop scheduling (Rahmati and Zandieh, 2012), power system optimization (Jamuna and Swarup, 2012), antenna design (Singh et al., 2010), and others (Chatterjee et al., 2012; Wang and Xu, 2011).

The main contribution of this paper is to propose new EA hybridization strategies that are based on migration behaviors in biogeography. We propose biogeography-based hybridization at both the iteration level and the algorithm level. Although BBO has already been hybridized with other algorithms, this paper represents the first time that EAs have been hybridized with each other using biogeography-based migration. Our motivation for biogeography-based migration in hybrid EAs is twofold: first, we note the good performance obtained in past research with BBO; and second, we note the good performance obtained in past research with hybrid EAs. Given these two factors, we hypothesize that hybridization using biogeography-based operations will provide some advantages over other hybrid EAs. We demonstrate our hybridization approaches with several recently-developed EAs, and we analyze the optimization results with statistical tests.

In iteration-level hybridization we combine various EAs with BBO. In algorithm-level hybridization we combine various EAs using ideas from biogeography. Note that in algorithm-level hybridization, we do not necessarily combine a particular EA with BBO. Instead we use the BBO migration strategy to combine multiple EAs. In this approach, various EAs are taken as the baseline algorithms, and then we make use of the migration mechanism of BBO to adaptively improve the solutions. That is, the constituent EAs generate offspring individuals each generation, and then we use the BBO migration operator to exchange information between these individuals.

The recently developed EAs that we hybridize include covariance matrix adaptation evolution strategy (CMA-ES) (Hansen, 2006; Hansen et al., 2003), stud genetic algorithm (SGA) (Khatib and Fleming, 1998), self-adaptive differential evolution (SaDE) (Zhao et al., 2011), 2011 standard particle swarm optimization (PSO2011) (Omran and Clerc, 2011), PSO with linearly varying inertia weight (LPSO) (Shi and Eberhart, 1998; Chatterjee and Siarry, 2006), and PSO with constriction factor (CPSO) (Clerc and Kennedy, 2002; Eberhart and Shi, 2000). We choose these algorithms because they are some of the most recent and best-performing EA variants. The six algorithms that we choose form a representative set rather than a complete set. We could hybridize many other algorithms besides these six. However, the goal here is not to be exhaustive, but rather to present a general biogeography-based hybridization strategy and demonstrate it on a representative set of constituent algorithms and benchmarks.

The rest of this paper is organized as follows: Section 2 gives a brief overview of EAs, including the constituent algorithms used in the rest of the paper. Section 3 presents our new hybridization methods. Section 4 tests our new algorithms on the continuous optimization benchmark functions from the 2013 Congress on Evolutionary Computation (CEC) and on some real-world traveling salesman problems, and performs some robustness tests. Section 5 gives conclusions and directions for future research.

2. Evolutionary algorithms

This section presents the basic outlines of the constituent EAs used in this paper, including CMA-ES, SGA, SaDE, PSO, and BBO.

2.1. Covariance matrix adaptation evolution strategy (CMA-ES)

ES is an evolutionary algorithm based on the ideas of adaptation during recombination, mutation, and selection. There are many variants of ES, and CMA-ES is a recent ES variant that has demonstrated good performance (Hansen, 2006; Hansen et al., 2003). It is a non-elitist algorithm that first samples a number of new candidate solutions from a multivariate normal distribution and then updates the sampling distribution using the better candidate solutions. The update consists of two major mechanisms: step size control and covariance matrix adaptation. In step size control, the length of the path of the most recent iteration step is adjusted. In covariance matrix adaptation, the likelihood of successful steps is increased. The time scales of the two updates are independent. The step size can change fast to allow for fast convergence to a good solution. The covariance matrix changes on a slower time scale to maintain stability.

2.2. Stud genetic algorithm (SGA)

GAs are the most popular EAs, and were introduced as a computational analogy of adaptive biological systems. They are modeled on natural selection. There are many GA variants, one of which is the stud GA (SGA) (Khatib and Fleming, 1998). The basic idea of SGA is to use the best solution in the population as one of the parents in all recombination operations. That is, instead of stochastic selection of both parents, only one parent is selected stochastically, and the other parent is always chosen as the fittest individual (the stud). The benefits of this GA variation are improved optimization performance and computational efficiency.

2.3. Self-adaptive differential evolution (SaDE)

DE is a simple evolutionary algorithm that creates new candidate solutions by combining the parent solution and several other candidate solutions. A candidate solution replaces the parent solution if it has better fitness. This is a greedy selection scheme that often outperforms traditional evolutionary algorithms. SaDE is one of the best DE variants (Zhao et al., 2011). It uses a self-adaptive mechanism on control parameters F and CR. Each candidate solution in the population is extended with control parameters F and CR that are adjusted during evolution. Better values of these control parameters lead to better candidate solutions, which in turn are more likely to survive the selection process to produce the next solution and propagate the good parameter values. SaDE is highly independent of the optimization problem's characteristics and complexity, and it involves self-adaptation and learning by experience. SaDE demonstrates consistently good performance on a variety of problems, including both unimodal and multimodal problems.

2.4. Particle swarm optimization (PSO)

PSO is a swarm optimization algorithm that is inspired by the collective behavior of a flock of birds or a school of fish. PSO consists of a swarm of particles moving through the search space of possible problem solutions. Every particle has a position vector encoding a candidate solution to the problem and a velocity vector to update position. PSO relies on the learning strategy of the particles to guide its search direction. Traditionally, each particle uses its historical best value and the global best value of the entire

swarm to guide its search. PSO2011 is a standard PSO implementation that includes PSO improvements that have been made over a period of many years (Omran and Clerc, 2011). PSO2011 has a structure that is more complex than standard PSO, but it demonstrates good optimization performance. Many other methods have also been proposed to improve the performance of PSO, and we use two of them in this paper: PSO with linearly varying inertia weight (LPSO) (Shi and Eberhart, 1998; Chatterjee and Siarry, 2006), and PSO with constriction factor (CPSO) (Clerc and Kennedy, 2002; Eberhart and Shi, 2000).

2.5. Biogeography-based optimization (BBO)

BBO is an EA that is inspired by biogeography (Simon, 2008). In BBO, a biogeography habitat denotes a candidate optimization problem solution, and it is comprised of a set of features, which are also called decision variables, or independent variables. A set of biogeography habitats denotes a population of candidate solutions, and habitat suitability index (HSI) in biogeography denotes the fitness of a candidate solution. Like other EAs, each candidate solution in BBO probabilistically shares decision variables with other candidate solutions to improve candidate solution fitness. This sharing process is analogous to migration in biogeography. That is, each candidate solution immigrates decision variables from other candidate solutions based on its immigration rate, and emigrates decision variables to other candidate solutions based on its emigration rate. BBO consists of two main steps: migration and mutation.

Migration is a probabilistic operator that is intended to improve a candidate solution y_k. For each decision variable of a given candidate solution y_k, the candidate solution's immigration rate λ_k is used to probabilistically decide whether or not to immigrate. If immigration is selected, then the emigrating candidate solution y_j is probabilistically chosen based on the emigration rate μ_j. The generalization of the standard BBO migration operator is written as follows (Ma and Simon, 2011a):

$$y_k(a) \leftarrow \delta y_k(a) + (1-\delta) y_j(a) \tag{1}$$

where a is a decision variable index; and δ is a real number between 0 and 1 which could be random or deterministic, or it could be proportional to the relative fitness of the solutions y_k and y_j. Eq. (1) means that the new decision variable of y_k comes from a combination of its own decision variable and a decision variable of the emigrating solution y_j. When $\delta = 0$, the decision variable of y_k is completely replaced by the decision variable of y_j. Note that if a decision variable immigrates to $y_k(a)$, then that decision variable is replaced in y_k. However, if a decision variable emigrates from $y_j(a)$, then that decision variable still remains in y_j. This is analogous to migration in nature: some representatives of species emigrate to new islands, but other representatives of that species remain on their original island.

In BBO, each candidate solution y_k has its own immigration rate λ_k and emigration rate μ_k. A good candidate solution has a relatively high emigration rate and low immigration rate, while the converse is true for a poor candidate solution. Here, immigration rate λ_k and emigration rate μ_k are based on particular migration curves, such as the linear migration curves presented in Fig. 1, where the maximum immigration rate and maximum emigration rate are both equal to 1. Nonlinear immigration rates λ_k and emigration rates μ_k have been discussed by Ma and Simon (2011b) in detail, and are also discussed in Section 4.4. The probability of immigrating to y_k and the probability of emigrating from y_k are calculated respectively as

$$\begin{aligned} &\text{Prob(immigration to } y_k) = \lambda_k \\ &\text{Prob(emigration from } y_k) = \frac{\mu_k}{\sum_{j=1}^{N} \mu_j} \end{aligned} \tag{2}$$

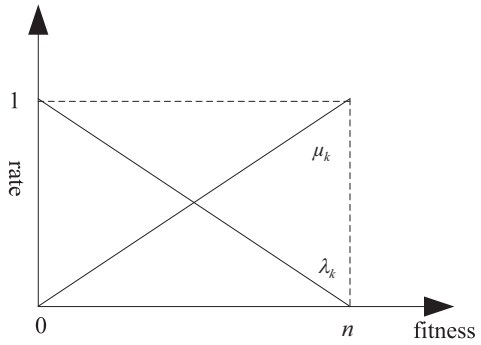

Fig. 1. Linear migration curves for BBO. λ_k is immigration rate and μ_k is emigration rate, n is the best fitness, and we assume that the maximum immigration rate and maximum emigration rate are both equal to 1.

where N is the population size, and the probability of emigration from y_k is based on roulette-wheel selection.

Mutation is a probabilistic operator that randomly modifies a decision variable of a candidate solution. The purpose of mutation is to increase diversity among the population, just as in other EAs. A description of one generation of BBO is given in Algorithm 1.

Algorithm 1. One generation of the BBO algorithm. y is the entire population of candidate solutions, y_k is the kth candidate solution, $y_k(a)$ is the ath decision variable of y_k, and δ is a control parameter between 0 and 1.

```
For each candidate solution y_k do
    For each candidate solution decision variable index a do
        Use λ_k to probabilistically decide whether to immigrate to y_k (see Eq. (2))
        If immigrating then
            Use {μ} to probabilistically select the emigrating candidate solution y_j (see Eq. (2))
            y_k(a) ← δ y_k(a) + (1 − δ) y_j(a)
        End if
    End for
    Probabilistically decide whether to mutate y_k
End for
```

3. Hybrid evolutionary algorithms

In this section, we propose two kinds of hybrid evolutionary algorithms based on biogeography. One is iteration-level hybridization (Section 3.1) and the other is algorithm-level hybridization (Section 3.2).

3.1. Iteration-level hybridization

Iteration-level hybridization is a straightforward method in which various EAs are executed in sequence. Iteration-level hybridization divides the search procedure into two stages: (1) in the first stage, one EA with high convergence speed is used to shrink the search region to more promising areas; (2) in the second stage, another EA with good exploration ability is used to explore the previously-limited area more extensively to find better solutions. Iteration-level hybridization will perform at least as well as one algorithm alone, and more often will perform better due to the synergy of exploration and exploitation. Previous studies have

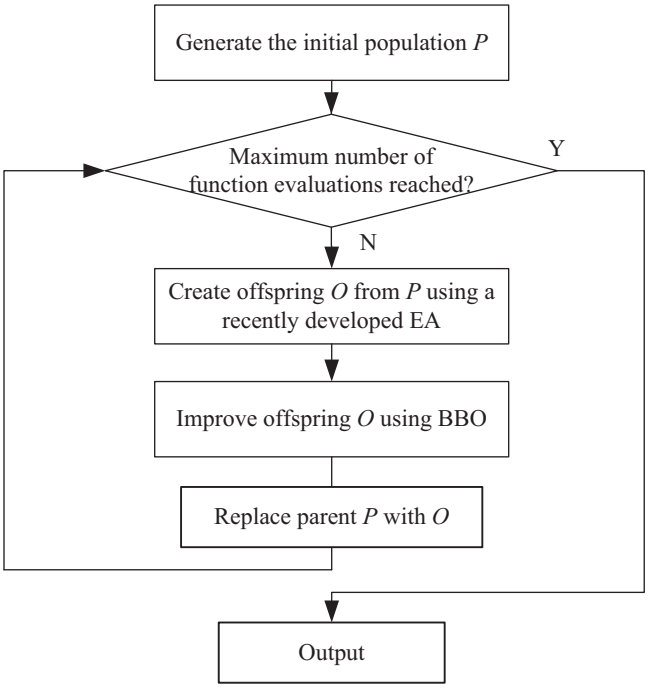

Fig. 2. Flowchart of iteration-level hybridization combining a recently developed EA with BBO, where P is the parent population and O is the offspring population.

found that this kind of hybrid EA improves optimization performance (Jaimes and Coello Coello, 2005). Another attractive feature of iteration-level hybridization is that its structure is simple and easily programmed.

In this paper, we implement iteration-level hybridization by combining recently developed EAs with BBO, in which one of recently developed EAs is used in the first stage to obtain good candidate solutions, and then BBO is used in the second stage to improve the candidate solutions obtained by the first EA. The goal of this hybridization approach is to balance the exploration and exploitation ability of various EAs with BBO. A general flowchart of iteration-level hybridization is shown in Fig. 2. The main procedure of iteration-level hybridization is shown in Algorithm 2. One generation of an iteration-level hybridization of SaDE and BBO is shown in Algorithm 3, which is a special case of Algorithm 2. Note that Algorithm 3 is provided for purposes of illustration. Any other EA could be hybridized at the iteration level with BBO, in which case Algorithm 3 would be modified accordingly.

Algorithm 2. Iteration-level hybridization.

```
Randomly initialize the parent population P
Evaluate the fitness of all candidate solutions in P
While the halting criterion is not satisfied do
    Execute a recently developed EA (for example, SaDE) to create offspring population O
    Evaluate the fitness of each solution in offspring population O
    Calculate the immigration rate λ and emigration rate μ of each solution
    Perform one generation of BBO as shown in Algorithm 1 to improve the solutions in offspring population O
    Replace the parent population P with the offspring population O
End while
```

Algorithm 3. One generation of an iteration-level hybridization of SaDE and BBO, where N is the population size. y and z comprise the entire population of candidate solutions, y_k is the kth candidate solution, and $y_k(a)$ is the ath decision variable of y_k. CR and F are the probability of crossover and the scaling factor of SaDE respectively, and δ is a BBO control parameter between 0 and 1.

```
z ← y
For each candidate solution z_k (k=1 to N ) do
    For each candidate solution decision variable index a do
        Pick three random solutions y_r1, y_r2 and y_r3 mutually distinct from each other and from z_k
        Pick a random index n between 1 and N
        Use CR (probabilistic) or n (deterministic) to decide on recombination
        If recombination then
            z_k(a) ← y_r1(a) + F(y_r2(a) − y_r3(a))
        End if
        Update the control parameters F and CR using the SaDE adaptive mechanism
    End for
    Evaluate the fitness of each candidate solution z_k in the population
    For each z_k define emigration rate μ_k proportional to the fitness of z_k, where μ_k ∈ [0,1]
    For each candidate solution z_k define immigration rate λ_k = 1 − μ_k
    For each candidate solution decision variable index a do
        Use λ_k to probabilistically decide whether to immigrate to z_k
        If immigrating then
            Use {μ} to probabilistically select the emigrating solution y_j
            z_k(a) ← δ z_k(a) + (1 − δ) y_j(a)
        End if
    End for
End for
y ← z
```

3.2. Algorithm-level hybridization

Algorithm-level hybridization is a method that involves several subpopulations running independently and periodically exchanging information with each other (Jaimes and Coello Coello, 2005). This hybridization method is shown in Fig. 3. The information exchange provides the mechanism to enhance a given subpopulation with the improvements achieved in other subpopulations. Therefore, algorithm-level hybridization will perform as good as each constituent algorithm, and more often it will perform better due to the exchange of information among the algorithms.

An important aspect of algorithm-level hybridization is the migration strategy, which is configured by various parameters: (i) *migration frequency*: how often is information shared between algorithms? (ii) *migration rate*: how much information migrates between algorithms? (iii) *information selection*: what information is selected to migrate between algorithms? and (iv) *migration topology*: which subpopulations exchange information with each other? A study of migration parameters has been presented by Jaimes and Coello Coello (2005), but that strategy needs to be adjusted based on a priori knowledge of the problem. It is hard to obtain useful information about the best migration strategy in the most real-world problems.

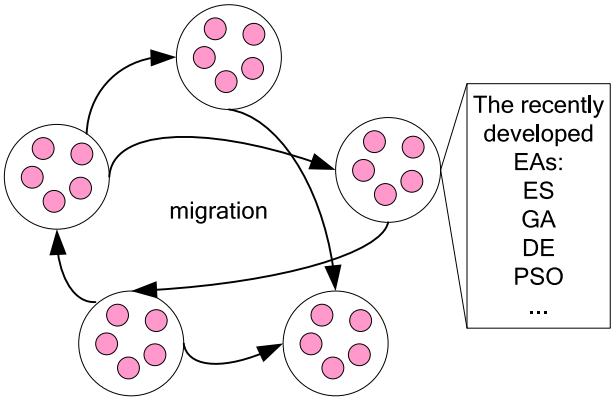

Fig. 3. Algorithm-level hybridization.

In our approach to algorithm-level hybridization, the fitness of each solution is used by BBO to determine the migration strategy between each algorithm, including migration frequency, migration rate, information selection, and migration topology. This migration strategy is naturally adaptive because of the information-exchanging mechanism of BBO. The advantage of this method is that BBO determines the migration parameter settings of algorithm-level hybridization, and interaction with a human decision maker does not need to occur during optimization. This hybridization approach, which combines recently developed EAs using biogeography-based strategies, has the common features of algorithm-level hybridization, but it also has the distinctive migration characteristics of BBO.

A flowchart of algorithm-level hybridization combining recently developed EAs using biogeography is shown in Fig. 4, where three subpopulations are used, although fewer or more could also be used, depending on the application. Different subpopulations execute independently and generate their own offspring subpopulations. All offspring subpopulations are combined with biogeography-based migration. In this way, algorithm-level hybridization will always keep the solutions that are improved by migration, which will lead to better optimization performance.

The main procedure of this approach to algorithm-level hybridization is shown in Algorithm 4. One generation of an algorithm-level hybridization of SaDE and BBO is shown in Algorithm 5, which is a special case of Algorithm 4. Note that Algorithm 5 is provided for purposes of illustration. Any other EA could be hybridized at the algorithm level with BBO, in which case Algorithm 5 would be modified accordingly.

Algorithm 4. Algorithm-level hybridization.

1: Randomly initialize the overall population P and divide it into subpopulations P_i ($i=1\ldots n$, where n denotes the number of subpopulations)
2: Evaluate the fitness of all candidate solutions in P
3: **While** the halting criterion is not satisfied **do**
4: **For** each subpopulation P_i **do**
5: Perform an independent EA to create offspring subpopulation O_i
6: **End for**
7: Evaluate the fitness of offspring population O, which is composed of all subpopulations O_i
8: Calculate the immigration rate λ and emigration rate μ for each offspring in the overall population O

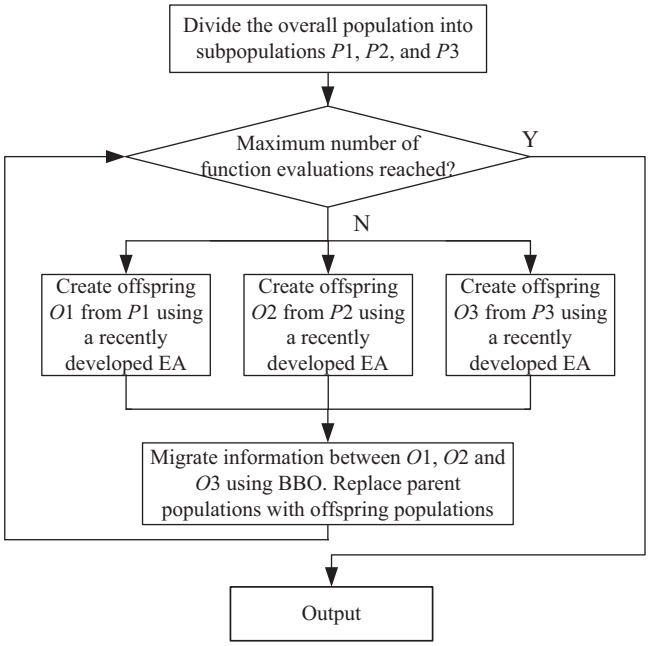

Fig. 4. Flowchart of algorithm-level hybridization combining recently developed EAs with biogeography-based migration, where there are three subpopulations. $P1$, $P2$, $P3$ are parent subpopulations, and $O1$, $O2$, $O3$ are offspring subpopulations.

9: **For** each offspring subpopulation O_i **do**
10: Immigrate solution information from the overall offspring population O using one generation of BBO as shown in Algorithm 1
11: **End for**
12: **End while**

Algorithm 5. One generation of an algorithm-level hybridization of SaDE and BBO, where P_i is the subpopulation, n is the number of subpopulations, and K is the subpopulation size. y and z are the entire population of candidate solutions, y_k is the kth candidate solution, and $y_k(a)$ is the ath decision variable of y_k. CR and F are the probability of crossover and the scaling factor of SaDE respectively, and δ is a BBO control parameter between 0 and 1.

1: $z \leftarrow y$
2: Divide z into subpopulations P_i ($i=1$ to n)
3: **For** each subpopulation P_i **do**
4: **For** each candidate solution z_{ik} ($k=1$ to K) **do**
5: **For** each candidate solution decision variable index a **do**
6: Pick three random solutions y_{r1}, y_{r2} and y_{r3} mutually distinct from each other and from z_{ik}
7: Pick a random index n between 1 and the population size
8: Use CR (probabilistic) or n (deterministic) to decide on recombination
9: **If** recombination **then**
10: $z_{ik}(a) \leftarrow y_{r1}(a) + F(y_{r2}(a) - y_{r3}(a))$
11: **End if**
12: Update the control parameters F and CR using the SaDE adaptive mechanism
13: **End for**
14: **End for**
15: **End for**

16: Evaluate the fitness of each candidate solution z_{ik} in all subpopulations
17: **For** each z_{ik} define emigration rate μ_{ik} proportional to fitness of z_{ik}, where $\mu_{ik} \in [0,1]$
18: For each candidate solution z_{ik} define immigration rate $\lambda_{ik} = 1 - \mu_{ik}$
19: **For** each subpopulation P_i **do**
20: **For** each candidate solution z_{ik} **do**
21: **For** each candidate solution decision variable index a **do**
22: Use λ_{ik} to probabilistically decide whether to immigrate to z_{ik}
23: **If** immigrating **then**
24: Use $\{\mu\}$ to probabilistically select the emigrating solution y_j from the combined population
25: $z_{ik}(a) \leftarrow \delta z_{ik}(a) + (1-\delta) y_j(a)$
26: **End if**
27: **End for**
28: **End for**
29: **End for**
30: $y \leftarrow z$

4. Experimental results

In this section we investigate the performance of iteration-level and algorithm-level hybrid EAs. Section 4.1 discusses the simulation setup, Section 4.2 presents performance comparisons on the 2013 CEC benchmark functions, Section 4.3 presents the results of statistical tests, Section 4.4 discusses the robustness to tuning parameters of the best hybrid algorithm in this study, and Section 4.5 applies the proposed hybrid algorithms to real-world traveling salesman problems.

4.1. Simulation setup

The performances of iteration-level and algorithm-level hybridization combining recently developed EAs with BBO are evaluated on the 28 global continuous optimization benchmark functions presented in Table 1, which are from the 2013 Congress on Evolutionary Computation (CEC) ⟨http://www3.ntu.edu.sg/home/EPNSugan/⟩. To prevent exploitation of symmetry of the search space and of the typical zero value associated with the global optimum, the optimum point in the search space is shifted to a value different from zero and the function values of the global optima are shifted to non-zero values (Liang et al., 2013).

The recently developed EAs in the proposed hybrid methods include CMA-ES, SGA, SaDE, PSO2011, LPSO, and CPSO. We select CMA-ES because it is the most successful variant of ES (Hansen, 2006; Hansen et al., 2003). We select SGA because it is an improvement of the basic GA that uses the best individual at each generation for crossover (Khatib and Fleming, 1998). We select SaDE because it is one of the most powerful DE algorithms and has demonstrated excellent performance on many problems (Zhao et al., 2011). We select PSO2011 because it is popular in the literature, and contains improvements gained as a result of many years of PSO studies (Omran and Clerc, 2011). We select PSO with linearly varying inertia weight (LPSO) (Shi and Eberhart, 1998; Chatterjee and Siarry, 2006) and PSO with constriction factor (CPSO) (Clerc and Kennedy, 2002; Eberhart and Shi, 2000) because they are classic PSO variants that are known to perform well for a large number of problems.

We do not test any hybridization using basic DE variants. One reason is that we instead hybridized SaDE, as described above, and SaDE generally performs better than basic DE. Another reason is that BBO has already been hybridized with basic DE in several publications, to which we refer the reader for additional details (Boussaïd et al., 2011, 2013a, 2013b).

In summary, we propose the following hybrid EAs: SGA/BBO-I; SGA/BBO-A; CMA-ES/BBO-I; CMA-ES/BBO-A; SaDE/BBO-I; SaDE/BBO-A; PSO2011/BBO-I; PSO2011/BBO-A; LPSO/BBO-I; LPSO/BBO-A; CPSO/BBO-I; and CPSO/BBO-A, where I and A denote iteration-level hybridization and algorithm-level hybridization respectively.

For example, SGA/BBO-I denotes iteration-level hybridization of SGA and BBO and is given in Fig. 2, Algorithms 2, and 3, where SaDE in the algorithm is replaced with SGA. Similarly, SGA/BBO-A denotes algorithm-level hybridization of SGA and BBO and is given in Fig. 4, Algorithms 4, and 5, where SaDE in the algorithm is replaced with SGA. Similar statements can be made for each of the hybrid EAs studied in this paper.

The next step is to set the parameters of each constituent algorithm. For BBO we use a maximum immigration rate and a maximum emigration rate of 1, linear migration curves as suggested in Fig. 1, a mutation probability of 0.001, and the parameter $\delta = 0$ in (1) which denotes standard migration. For SGA we use real coding, roulette-wheel selection, single-point crossover with a crossover probability of 1, and a mutation probability of 0.001. For CMA-ES we use the parameter given by Hansen (2006). The SaDE parameter settings are adapted according to the learning progress (Zhao et al., 2011): the scaling factor F is randomly sampled from the normal distribution $N(0.5, 0.3)$, and the crossover rate CR follows the normal distribution $N(0.5, 0.1)$. For PSO2011, we use an inertia weight $w = 1/2\log(2)$, a cognitive constant $c_1 = 0.5 + \log(2)$, and a social constant c_2 for neighborhood interaction that is the same as c_1. For LPSO we use a fixed initial inertia weight $w_{init} = 0.2$, an inertia weight slope $m = -2.5 \times 10^{-4}$, and a nonlinear modulation index $n = 1$ (Shi and Eberhart, 1998). For CPSO we use a constriction coefficient $K = 2/2 - \varphi - \sqrt{\varphi^2 - 4\varphi}$, where $\varphi = 4.1$ (Clerc and Kennedy, 2002). The other parameters of LPSO and CPSO are the same as those of PSO2011.

For algorithm-level hybridization, we use three subpopulations, and each subpopulation implements the same EA. The population size of each subpopulation in algorithm-level hybridization is 30, so the total population size is 90. For fair comparisons, the population size of iteration-level hybridization is also set to 90. We evaluate each function in 50 dimensions with the function evaluation limit of 500,000. All algorithms are terminated after the maximum number of function evaluations is reached, or if the objective function error value is below 10^{-8}.

4.2. Performance comparisons

We simulated each algorithm discussed in the previous section 25 times on each benchmark, and the results are shown in Tables 2, 3 and 4. The tables show that SaDE/BBO-A performs best on 13 functions (F1, F3, F4, F5, F7, F10, F11, F14, F18, F19, F23, F24 and F28), SaDE/BBO-I performs best on 12 functions (F1, F2, F5, F6, F8, F9, F13, F15, F17, F20, F21, and F27), CMA-ES/BBO-A performs best on 5 functions (F1, F11, F14, F16 and F25), PSO2011/BBO-A performs best on 4 functions (F1, F5, F12 and F14), LPSO/BBO-A performs best on 3 functions (F1, F5 and F22), CPSO/BBO-A performs best on 3 functions (F1, F5 and F26), PSO2011/BBO-I, LPSO/BBO-I, and CPSO/BBO-I performs best on 2 functions (F1 and F5), and CMA-ES/BBO-I performs best on function F11.

The results indicate that the 18 algorithms can generally be listed in order from best-performing to worst-performing as follows:

(1) SaDE/BBO-A
(2) SaDE/BBO-I
(3) CMA-ES/BBO-A

Table 1
2013 CEC benchmark functions, where the search range of all functions is $-100 \le x_i \le 100$. More details about these functions can be found in Liang et al. (2013).

		Function name	Minimum
Unimodal functions	F1	Sphere function	−1400
	F2	Rotated high conditioned elliptic function	−1300
	F3	Rotated bent cigar function	−1200
	F4	Rotated discus function	−1100
	F5	Different powers function	−1000
Basic multimodal functions	F6	Rotated Rosenbrock function	−900
	F7	Rotated Schaffer F7 function	−800
	F8	Rotated Ackley function	−700
	F9	Rotated Weierstrass function	−600
	F10	Rotated Griewank function	−500
	F11	Rastrigin function	−400
	F12	Rotated Rastrigin function	−300
	F13	Discontinuous rotated Rastrigin function	−200
	F14	Schwefel function	−100
	F15	Rotated Schwefel function	100
	F16	Rotated Katsuura function	200
	F17	Lunacek Bi_Rastrigin function	300
	F18	Rotated Lunacek Bi_Rastrigin function	400
	F19	Expanded Griewank plus Rosenbrock function	500
	F20	Expanded Schaffer F6 function	600
Composition functions	F21	Composition function 1 (n=5, rotated)	700
	F22	Composition function 2 (n=3, unrotated)	800
	F23	Composition function 3 (n=3, rotated)	900
	F24	Composition function 4 (n=3, rotated)	1000
	F25	Composition function 5 (n=3, rotated)	1100
	F26	Composition function 6 (n=5, rotated)	1200
	F27	Composition function 7 (n=5, rotated)	1300
	F28	Composition function 8 (n=5, rotated)	1400

Table 2
Comparisons of the best error values of the 2013 CEC benchmark functions with D=50 for SGA and CMA-ES. Here [$a \pm b$] indicates the mean value and the corresponding standard deviation of 25 independent simulations. The best result in each row (Tables 2, 3 and 4 combined) is shown in bold font. CPU times (min) are shown in the last row of the table.

	SGA	SGA/BBO-I	SGA/BBO-A	CMA-ES	CMA-ES/BBO-I	CMA-ES/BBO-A
F1	4.57E−06 ± 1.53E−07	4.66E−12 ± 2.36E−13	7.19E−15 ± 5.15E−16	6.11E−14 ± 9.10E−15	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**
F2	1.90E+06 ± 4.62E+05	1.93E+03 ± 8.09E+02	3.28E+01 ± 2.46E+00	1.90E+04 ± 3.83E+03	4.02E+02 ± 3.32E+01	5.65E+01 ± 5.44E+00
F3	7.19E+03 ± 3.28E+02	1.34E+01 ± 6.12E+00	6.71E+00 ± 3.90E+00	3.59E+01 ± 4.62E+00	5.28E+00 ± 1.89E+00	8.03E+00 ± 1.93E+00
F4	9.01E+03 ± 7.23E+02	4.21E−01 ± 3.54E−02	8.08E−01 ± 1.78E−02	1.21E+00 ± 7.78E−01	4.45E−02 ± 4.20E−03	1.26E−03 ± 3.86E−04
F5	9.16E−02 ± 9.65E−03	1.23E−04 ± 7.89E−05	3.25E−06 ± 7.19E−07	5.18E−05 ± 6.01E−06	3.21E−10 ± 8.31E−11	7.78E−12 ± 1.19E−13
F6	1.11E+02 ± 2.32E+01	2.36E−01 ± 8.38E−02	1.19E−03 ± 2.35E−04	6.63E+01 ± 9.77E+00	8.91E−02 ± 6.63E−03	1.55E−02 ± 4.65E−03
F7	7.85E+07 ± 4.16E+06	3.81E+04 ± 3.29E+03	2.37E+03 ± 1.19E+02	1.33E+08 ± 2.25E+07	6.33E+05 ± 1.25E+04	7.90E+05 ± 9.03E+04
F8	2.33E+06 ± 1.90E+05	8.90E+03 ± 5.66E+03	1.26E+03 ± 2.37E+02	7.06E+05 ± 6.38E+04	1.86E+01 ± 4.78E+00	1.52E+02 ± 6.67E+01
F9	5.47E+10 ± 2.19E+09	1.54E+06 ± 2.17E+05	4.44E+04 ± 1.19E+03	8.92E+10 ± 1.17E+09	9.28E+04 ± 1.93E+03	8.80E+07 ± 8.23E+06
F10	6.89E+11 ± 3.13E+10	5.78E+07 ± 8.94E+06	3.25E+06 ± 6.77E+05	4.97E+10 ± 4.56E+09	1.38E+04 ± 2.25E+03	8.32E+05 ± 1.19E+04
F11	7.76E+10 ± 1.85E+09	3.26E+08 ± 1.05E+07	7.89E+07 ± 3.28E+06	9.00E+07 ± 7.89E+06	**4.24E+00 ± 2.15E+00**	**4.24E+00 ± 3.77E+00**
F12	2.66E+08 ± 3.28E+07	1.98E+04 ± 2.44E+03	9.17E+01 ± 1.16E+00	1.35E+06 ± 1.92E+05	8.80E+02 ± 5.23E+01	9.29E+01 ± 2.44E+00
F13	9.05E+05 ± 4.27E+06	9.86E+00 ± 7.83E−01	5.54E+02 ± 1.32E+00	7.66E+02 ± 9.08E+01	1.15E−01 ± 9.91E−02	5.56E−02 ± 2.79E−03
F14	8.21E+11 ± 8.03E+10	7.27E+08 ± 2.70E+07	7.89E+03 ± 9.33E+02	7.26E+10 ± 4.43E+09	8.36E+04 ± 6.26E+03	**2.17E+01 ± 4.21E+00**
F15	9.32E+03 ± 4.26E+02	2.60E+03 ± 5.16E+02	9.76E+03 ± 1.27E+02	3.28E+02 ± 1.95E+01	8.08E+02 ± 2.84E+01	1.99E+01 ± 4.47E+00
F16	6.71E+09 ± 3.19E+08	5.89E+06 ± 1.93E+05	3.71E+03 ± 4.26E+02	5.44E+08 ± 4.83E+07	1.36E+02 ± 1.16E+01	**2.46E−01 ± 8.93E−02**
F17	5.66E+04 ± 1.83E+03	3.25E+02 ± 6.67E+01	2.26E+02 ± 9.83E+01	1.36E+03 ± 1.59E+02	7.02E+02 ± 5.53E+01	9.18E+02 ± 1.22E+01
F18	2.41E+11 ± 6.65E+10	1.87E+08 ± 5.26E+07	5.32E+06 ± 1.64E+05	2.19E+10 ± 4.54E+09	4.39E+08 ± 9.90E+07	3.42E+05 ± 5.82E+04
F19	1.99E+10 ± 6.27E+09	1.19E+05 ± 1.38E+04	9.01E+07 ± 3.14E+06	9.09E+08 ± 7.16E+07	1.90E+04 ± 3.65E+03	1.26E+06 ± 9.43E+05
F20	7.85E+04 ± 1.26E+03	5.46E+02 ± 2.14E+01	6.65E+00 ± 8.91E−01	3.24E+03 ± 8.85E+02	2.21E+01 ± 8.29E+00	8.85E−01 ± 8.89E−02
F21	8.05E+07 ± 4.16E+06	3.24E+03 ± 3.25E+02	3.28E+01 ± 9.00E+00	1.15E+05 ± 9.93E+04	5.46E+02 ± 2.38E+01	9.29E+03 ± 1.25E+02
F22	9.43E+05 ± 1.75E+04	1.17E+03 ± 9.05E+02	1.27E+03 ± 1.01E+02	2.26E+02 ± 6.50E+01	3.21E+01 ± 6.67E+00	1.77E+00 ± 9.90E−01
F23	9.79E+10 ± 6.17E+09	4.93E+08 ± 4.21E+07	8.89E+08 ± 5.76E+07	4.58E+08 ± 3.18E+07	7.52E+06 ± 2.46E+05	3.10E+05 ± 1.26E+04
F24	8.23E+08 ± 4.31E+07	1.92E+06 ± 6.63E+05	1.25E+07 ± 1.92E+06	9.04E+09 ± 8.99E+08	1.16E+05 ± 9.03E+04	8.93E+07 ± 7.38E+06
F25	1.16E+08 ± 8.53E+07	8.05E+05 ± 8.01E+04	6.78E+04 ± 8.95E+03	8.87E+07 ± 1.50E+06	5.29E+05 ± 2.25E+04	**9.01E+01 ± 1.22E+00**
F26	1.03E+06 ± 2.95E+05	9.17E+05 ± 1.54E+04	5.99E+03 ± 9.73E+02	1.26E+04 ± 1.11E+03	6.13E+04 ± 7.16E+03	5.57E+04 ± 4.37E+03
F27	7.80E+10 ± 7.99E+09	8.22E+06 ± 3.28E+05	1.25E+08 ± 8.17E+07	5.53E+07 ± 2.83E+06	9.04E+05 ± 3.32E+04	1.13E+05 ± 6.62E+04
F28	6.34E+10 ± 1.26E+09	7.38E+08 ± 1.19E+07	3.18E+08 ± 2.33E+07	2.23E+10 ± 1.27E+09	4.33E+06 ± 7.79E+05	7.74E+05 ± 4.87E+04
CPU time	126.79	167.24	131.70	235.44	279.65	240.36

(4) PSO2011/BBO-A
(5) LPSO/BBO-A and CPSO/BBO-A
(6) PSO2011/BBO-I, LPSO/BBO-I, and CPSO/BBO-I
(7) CMA-ES/BBO-I
(8) SGA, SGA/BBO-I, SGA/BBO-A, CMA-ES, SaDE, PSO2011, LPSO, and CPSO

Table 3
Comparisons of the best error values of the 2013 CEC benchmark functions with $D=50$ for SaDE and PSO2011. Here $[a \pm b]$ indicates the mean value and the corresponding standard deviation of 25 independent simulations. The best result in each row (Tables 2, 3 and 4 combined) is shown in bold font. CPU times (min) are shown in the last row of the table.

	SaDE	SaDE/BBO-I	SaDE/BBO-A	PSO2011	PSO2011/BBO-I	PSO2011/BBO-A
F1	3.32E−10 ± 4.47E−11	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	3.35E−19 ± 3.44E−20	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**
F2	6.53E+03 ± 3.29E+02	**1.12E+01 ± 1.98E+00**	6.73E+02 ± 2.37E+01	7.36E+03 ± 1.39E+02	3.41E+02 ± 5.34E+01	6.51E+02 ± 1.77E+01
F3	4.19E+02 ± 2.16E+00	3.01E−01 ± 2.67E−01	**9.12E−02 ± 4.28E−02**	1.43E+01 ± 2.38E+00	8.90E+00 ± 7.18E+00	8.98E+00 ± 3.29E+00
F4	3.11E+00 ± 2.45E−01	2.35E−06 ± 9.05E−07	**7.73E−10 ± 1.02E−11**	6.55E+00 ± 1.66E−01	4.01E−04 ± 3.98E−05	7.50E−08 ± 1.38E−09
F5	7.66E−10 ± 2.79E−11	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	4.26E−08 ± 5.44E−09	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**
F6	1.30E+02 ± 1.35E+01	**1.23E−05 ± 4.53E−06**	6.77E−04 ± 2.11E−05	5.90E+02 ± 3.28E+02	3.28E−03 ± 1.90E−04	3.43E−03 ± 2.24E−04
F7	6.63E+07 ± 7.72E+06	8.67E+02 ± 3.28E+01	**5.14E+00 ± 6.42E−01**	8.34E+07 ± 4.16E+06	9.12E+02 ± 4.46E+01	6.87E+02 ± 1.16E+01
F8	5.99E+05 ± 8.29E+04	**2.27E−01 ± 4.43E−02**	8.01E+00 ± 9.03E−01	4.51E+06 ± 5.89E+05	6.87E+01 ± 2.38E+00	2.93E+01 ± 5.33E+00
F9	1.82E+09 ± 7.01E+08	**5.65E+00 ± 1.39E−01**	6.65E+00 ± 2.17E−01	7.82E+10 ± 7.33E+09	8.90E+07 ± 1.15E+06	9.19E+06 ± 1.90E+05
F10	6.05E+10 ± 1.25E+08	2.90E+01 ± 6.65E+00	**8.93E+00 ± 2.93E−01**	9.16E+10 ± 2.16E+09	1.10E+06 ± 5.54E+05	6.77E+06 ± 7.35E+05
F11	1.17E+07 ± 6.67E+06	2.16E+01 ± 5.33E+00	**4.24E+00 ± 1.18E+00**	8.01E+09 ± 4.58E+08	2.11E+03 ± 3.29E+02	8.01E+01 ± 2.66E+00
F12	3.99E+05 ± 3.19E+04	9.92E+00 ± 1.18E−01	8.00E+00 ± 4.66E−02	6.63E+06 ± 3.29E+05	4.63E+00 ± 2.73E−01	**3.22E−03 ± 3.25E−04**
F13	2.52E−01 ± 5.12E−02	**7.54E−05 ± 2.71E−06**	1.92E−01 ± 8.93E−02	1.19E+01 ± 1.76E+00	6.79E−01 ± 1.22E−02	6.76E−03 ± 1.03E−04
F14	1.88E+09 ± 6.24E+08	3.90E+01 ± 8.03E+00	**2.17E+01 ± 3.34E+00**	8.21E+09 ± 4.44E+08	1.18E+03 ± 7.36E+02	**2.17E+01 ± 1.09E+00**
F15	2.67E+04 ± 8.87E+03	**1.16E+01 ± 6.67E+00**	1.11E+02 ± 2.54E+01	7.76E+05 ± 5.34E+04	4.35E+02 ± 5.89E+01	5.44E+03 ± 5.39E+02
F16	4.26E+07 ± 1.25E+06	7.82E+01 ± 8.80E+00	3.43E+01 ± 7.65E+00	2.17E+08 ± 1.65E+07	8.82E+02 ± 1.90E+01	7.12E+02 ± 2.88E+01
F17	7.09E+03 ± 4.33E+02	**1.35E−01 ± 2.23E−02**	5.67E+01 ± 4.26E−01	3.43E+02 ± 2.19E+01	7.76E+01 ± 2.34E+00	9.83E+00 ± 4.32E−01
F18	3.18E+08 ± 1.28E+07	7.70E+05 ± 1.19E+04	**8.98E+01 ± 5.38E+00**	8.92E+09 ± 8.75E+08	9.93E+05 ± 1.76E+04	6.01E+02 ± 1.23E+01
F19	6.67E+05 ± 8.10E+04	8.54E+04 ± 5.99E+03	**1.65E+01 ± 9.87E+00**	9.08E+05 ± 2.15E+05	1.18E+02 ± 2.90E+01	5.99E+04 ± 9.06E+03
F20	1.55E+01 ± 1.16E−05	**2.43E−02 ± 1.05E−03**	1.90E+01 ± 1.22E+00	7.71E+03 ± 4.43E+04	6.02E+01 ± 4.33E+00	1.34E+02 ± 7.76E+01
F21	2.35E+02 ± 2.24E+01	**7.81E+00 ± 4.43E−01**	8.73E+02 ± 4.34E+01	2.59E+04 ± 3.29E+03	8.60E+01 ± 5.27E+00	7.18E+03 ± 1.22E+02
F22	1.73E+01 ± 7.55E+00	9.01E+00 ± 5.29E−01	9.56E+02 ± 8.08E+01	2.34E+02 ± 1.38E+02	6.67E−02 ± 1.38E−03	9.21E−03 ± 5.47E−04
F23	9.09E+09 ± 4.32E+08	8.87E+06 ± 3.88E+05	**3.62E+02 ± 5.76E+01**	8.93E+09 ± 2.16E+08	3.32E+07 ± 4.26E+06	7.74E+06 ± 6.63E+05
F24	2.63E+04 ± 3.89E+03	9.12E+02 ± 1.37E+01	**2.11E+02 ± 1.64E+00**	9.32E+06 ± 4.43E+05	9.25E+04 ± 1.44E+04	1.10E+05 ± 4.11E+04
F25	5.56E+04 ± 511E+03	3.45E+03 ± 2.05E+02	6.84E+02 ± 8.73E+01	6.89E+05 ± 5.23E+04	7.87E+04 ± 3.19E+03	7.63E+04 ± 2.18E+03
F26	1.44E+02 ± 2.35E+01	7.76E+02 ± 4.16E+01	8.63E+02 ± 1.29E+01	1.90E+03 ± 1.77E+02	1.80E+02 ± 5.27E+01	8.89E+03 ± 7.35E+02
F27	3.91E+06 ± 9.00E+05	**8.13E+02 ± 3.99E+01**	9.12E+02 ± 7.06E+01	5.76E+07 ± 4.10E+06	7.75E+05 ± 3.00E+05	1.14E+04 ± 6.68E+03
F28	7.05E+09 ± 8.71E+08	7.76E+05 ± 4.45E+04	**9.12E+02 ± 9.18E+01**	3.11E+09 ± 5.43E+08	8.67E+06 ± 4.24E+05	6.02E+05 ± 7.19E+04
CPU time	194.18	252.71	207.42	155.04	213.74	162.83

Table 4
Comparisons of the best error values of the 2013 CEC benchmark functions with $D=50$ for LPSO and CPSO. Here $[a \pm b]$ indicates the mean value and the corresponding standard deviation of 25 independent simulations. The best result in each row (Tables 2, 3 and 4 combined) is shown in bold font. CPU times (min) are shown in the last row of the table.

	LPSO	LPSO/BBO-I	LPSO/BBO-A	CPSO	CPSO/BBO-I	CPSO/BBO-A
F1	2.15E−15 ± 3.78E−16	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	5.53E−16 ± 1.274E−17	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**
F2	4.12E+03 ± 2.44E+02	5.22E+02 ± 1.39E+01	7.81E+03 ± 4.23E+02	8.04E+04 ± 4.32E+02	1.21E+02 ± 2.30E+01	3.17E+02 ± 2.28E+01
F3	2.04E+01 ± 5.42E+00	1.04E+01 ± 3.16E+00	1.00E+01 ± 3.65E+00	5.52E+01 ± 7.32E+00	1.77E+01 ± 1.05E+00	6.32E+00 ± 1.15E+00
F4	4.36E−02 ± 1.81E−03	5.64E−06 ± 2.11E−07	7.15E−08 ± 3.66E−09	5.36E−04 ± 2.47E−05	8.71E−07 ± 9.38E−08	2.45E−08 ± 4.16E−09
F5	4.24E−12 ± 5.23E−13	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	5.64E−15 ± 1.18E−16	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**
F6	1.21E−01 ± 7.20E−02	6.32E−03 ± 1.14E−04	7.78E−04 ± 1.29E−05	7.63E−02 ± 4.15E−03	7.65E−02 ± 3.29E−03	8.76E−04 ± 5.35E−05
F7	9.30E+05 ± 5.77E+04	4.65E+05 ± 3.28E+04	7.63E+04 ± 2.31E+03	7.65E+05 ± 1.96E+04	2.74E+04 ± 6.31E+03	7.99E+03 ± 4.65E+02
F8	5.36E+07 ± 2.44E+06	7.58E+04 ± 1.19E+03	3.62E+02 ± 4.19E+01	6.35E+06 ± 2.28E+05	7.42E+03 ± 2.15E+02	3.64E+02 ± 5.20E+01
F9	7.26E+09 ± 5.13E+08	5.36E+04 ± 2.38E+03	4.12E+05 ± 8.36E+04	2.28E+08 ± 7.65E+07	6.36E+05 ± 4.11E+04	4.34E+03 ± 3.72E+02
F10	7.31E+08 ± 5.66E+07	7.25E+05 ± 1.18E+04	3.46E+04 ± 5.09E+03	4.43E+08 ± 7.70E+07	5.65E+04 ± 4.74E+03	1.31E+04 ± 4.43E+03
F11	7.75E+06 ± 2.31E+05	4.36E+06 ± 8.14E+05	9.63E+04 ± 1.17E+03	4.32E+07 ± 2.36E+06	5.98E+03 ± 1.19E+02	3.47E+02 ± 6.54E+01
F12	7.59E+06 ± 2.16E+05	7.63E+00 ± 1.18E−01	4.35E−01 ± 7.64E−02	4.21E+06 ± 2.47E+05	1.16E+00 ± 3.28E−01	4.33E−01 ± 2.71E−02
F13	2.33E+01 ± 1.02E+00	7.45E−02 ± 8.65E−03	7.74E−03 ± 2.01E−04	7.75E−01 ± 2.26E−00	1.12E−01 ± 1.39E−02	8.69E−01 ± 4.43E−02
F14	7.66E+06 ± 2.34E+05	4.69E+03 ± 7.14E+02	1.53E+02 ± 4.11E+01	2.66E+05 ± 1.35E+04	6.35E+03 ± 2.48E+02	7.98E+01 ± 5.56E+00
F15	1.23E+05 ± 6.33E+04	2.71E+03 ± 2.78E+02	6.35E+02 ± 1.29E+01	8.65E+06 ± 3.32E+05	1.66E+03 ± 2.39E+02	8.74E+03 ± 4.43E+02
F16	2.17E+09 ± 1.65E+08	8.82E+04 ± 1.90E+03	4.71E+03 ± 2.36E+02	5.04E+07 ± 3.28E+06	1.76E+03 ± 4.52E+02	8.65E+04 ± 3.74E+03
F17	9.65E+02 ± 3.28E+01	4.17E+00 ± 2.30E−01	5.44E+00 ± 2.23E−01	7.65E+03 ± 4.11E+02	3.24E+01 ± 6.78E+00	6.03E+00 ± 7.12E−01
F18	5.59E+07 ± 7.74E+06	2.15E+03 ± 8.66E+02	5.39E+04 ± 4.28E+03	1.76E+06 ± 3.14E+05	6.53E+02 ± 2.47E+01	8.65E+03 ± 4.26E+02
F19	1.05E+05 ± 6.61E+04	5.33E+04 ± 2.47E+03	3.19E+03 ± 1.18E+02	5.65E+04 ± 4.41E+03	3.25E+04 ± 7.76E+03	1.12E+04 ± 8.86E+03
F20	5.32E+03 ± 4.17E+04	7.75E−01 ± 2.36E−02	3.22E−01 ± 7.54E−02	2.39E+03 ± 6.40E+04	8.32E−01 ± 1.66E−00	8.42E−02 ± 2.32E−01
F21	1.86E+04 ± 5.24E+03	2.26E+01 ± 7.14E+00	2.96E+03 ± 5.40E+02	6.32E+03 ± 1.76E+02	4.22E+01 ± 6.17E+00	9.02E+03 ± 3.76E+02
F22	7.26E+01 ± 9.01E+02	7.49E−02 ± 2.33E−03	**2.77E−03 ± 5.19E−04**	4.99E+01 ± 2.40E+02	7.32E−03 ± 4.23E−03	3.21E−02 ± 6.55E−03
F23	5.33E+07 ± 7.18E+06	4.26E+04 ± 1.31E+03	6.54E+04 ± 4.23E+03	2.66E+09 ± 5.14E+08	7.65E+03 ± 2.81E+02	3.22E+03 ± 7.13E+02
F24	1.01E+07 ± 3.47E+06	2.85E+04 ± 3.64E+03	5.41E+03 ± 2.26E+02	2.61E+07 ± 1.96E+06	4.32E+03 ± 8.64E+02	9.68E+04 ± 3.47E+03
F25	3.24E+06 ± 5.17E+05	6.84E+04 ± 3.22E+03	1.89E+04 ± 4.36E+03	8.82E+07 ± 7.43E+06	9.85E+04 ± 2.35E+03	7.14E+04 ± 4.49E+03
F26	6.47E+03 ± 1.56E+02	5.33E+03 ± 2.17E+02	8.65E+02 ± 1.25E+01	8.77E+04 ± 1.48E+03	6.32E+02 ± 7.14E+01	**4.36E+01 ± 1.23E+01**
F27	3.69E+08 ± 2.38E+07	7.48E+04 ± 5.19E+03	3.48E+05 ± 3.64E+04	1.26E+08 ± 2.35E+07	6.14E+03 ± 7.16E+02	4.43E+03 ± 5.31E+02
F28	1.16E+07 ± 3.22E+06	3.26E+04 ± 7.74E+03	1.07E+04 ± 7.80E+03	6.48E+08 ± 7.71E+07	3.22E+05 ± 8.62E+04	2.37E+03 ± 1.45E+02
CPU time	172.02	232.91	193.70	185.13	246.52	202.01

Tables 2–4 show that the constituent algorithms of the hybrid EAs, including SGA, CMA-ES, SaDE, PSO2011, LPSO, and CPSO, cannot find any optimal solutions, and the performance of the hybrid algorithms are better than their constituent algorithms. These results indicate that hybrid biogeography-based algorithms can improve performance for the continuous benchmark functions in this paper.

We also note from the above ordered list of best-performing algorithms that SaDE hybrids perform best, and algorithm-level hybrids perform better than iteration-level hybrids. This leads to a couple of interesting conclusions. First, it indicates that SaDE performs better than the other algorithms, at least for the tuning parameters and benchmarks considered in this paper. Second, it indicates the superiority of algorithm-level hybridization over iteration-level hybridization. This may be due to the interacting subpopulations that comprise algorithm-level hybridization, which is a structure that other research has also found to be highly efficient for global optimization (Das et al., 2011; Lassig and Sudholt, 2010).

Next we briefly consider the types of functions for which the various algorithms are best-suited. Tables 2, 3, and 4 show that the best-performing algorithm on each of the unimodal functions (F1–F5) is always one of the SaDE/BBO hybrids. The SaDE/BBO hybrids also perform well on the other function categories, but their superior performance on all five unimodal functions indicates that the performance levels of the hybrid algorithms become more even as the optimization problem becomes more difficult. We also note from the tables that algorithm-level hybrids perform best on six of the eight composition functions (F21–F28). This implies that for extremely difficult and complex optimization problems, algorithm-level hybridization would probably be preferred over iteration-level hybridization. This is consistent with the observation in the previous paragraph about the superiority of algorithm-level hybridization due to its interacting subpopulations.

The average running times of all algorithms are shown in the last row of Tables 2, 3 and 4. Here MATLAB® is used as the programming language, and the computer is a 2.40 GHz Intel Pentium® 4 CPU with 4 GB of memory. We find that the average running times of the constituent algorithms are less than their corresponding hybrid algorithms. For example, the average running times of SGA are less than SGA/BBO-I and SGA/BBO-A. We also find that the algorithms can be ranked from fastest to slowest as follows:

(1) SGA and its hybrid algorithms
(2) PSO2011 and its hybrid algorithms
(3) LPSO and its hybrid algorithms
(4) CPSO and its hybrid algorithms
(5) SaDE and its hybrid algorithms
(6) CMA-ES and its hybrid algorithms.

We also find that the average running times of the algorithm-level hybrids are less than those of the iteration-level hybrids for the same constituent algorithm. For example, the average running time of SGA/BBO-A is less than SGA/BBO-I. The reason is that algorithm-level hybridization uses multiple parallel subpopulations to reduce computation time with the same total population size as iteration-level hybridization. Certain EA operations, such as roulette-wheel selection, have computational effort on the order of N^2, where N is the population size. So multiple subpopulations are more computationally efficient than a single large population. Multiple subpopulations are also amenable to parallel processing, which can further reduce computational effort.

Finally, we note from Table 2 that SGA and its hybrid algorithms did not perform the best in any of the 28 benchmarks. This could be due to the simplicity of SGA, which we see from the fact that SGA and its hybrid algorithms have the fastest run time. Better performance might be obtained by varying the tuning parameters of SGA, but SGA includes only a couple of tuning parameters (crossover probability and mutation probability), and also, better performance might also be obtained in the other EAs with additional tuning. We conclude that SGA is a simple algorithm that provides good performance, but is generally not competitive with more complex EAs such as CMA-ES, DE, and PSO.

4.3. Statistical tests

In order to further compare the performance of the hybrid algorithms, we perform a Holm multiple comparison test, considering SADE/BBO-A as the control method, which is regarded as the best algorithm based on the results of Section 4.2. The Holm multiple comparison test is a nonparametric statistical test that obtains a probability (p-value) that determines the degree of difference between a control algorithm and a set of alternative algorithms, assuming that the algorithms have statistically significant differences as a whole (Demsar, 2006). To quantify whether a set of algorithms shows a statistically significant difference as a whole, we first apply Friedman's test (with a significance level $\alpha=0.05$) to the mean error rankings (Friedman, 1940). If the test rejects the null hypothesis that all of the algorithms perform similarly, we consider the best algorithm as the control method and compare it with the remaining algorithms according to their rankings (Demsar, 2006; Dunn, 1961; Hochberg and Tamhane, 1987). Additional details about the Holm multiple comparison test can be found in the literature (Derrac et al., 2011).

We compare our proposed hybrid algorithms with seven algorithms that were accepted for the 2013 CEC competition ⟨http://www3.ntu.edu.sg/home/EPNSugan/⟩. We collect benchmark performance data for these EAs from the below references. Note that in ⟨http://www3.ntu.edu.sg/home/EPNSugan/⟩ there are 22 accepted papers, but we only use the seven that show relatively good performance and that provide software to reproduce results.

(1) iCMAES-ILS, which is a hybrid algorithm combining IPOP-CMA-ES (CMA-ES with increasing population size) and iterated local search (ILS) (Liao and Stuetzle, 2013).
(2) NBIPOP-aCMA-ES, which is an active CMA-ES with increasing bi-population size and decreasing initial step-size (Loshchilov, 2013).
(3) DRMA-LSch-CMA, which is a dynamically updated region-based memetic algorithm with local search chaining and CMA-ES (Lacroix et al., 2013).
(4) SHADE, which is success history based adaptive differential evolution (Tanabe and Fukunaga, 2013).
(5) MVMO-SH, which is a mean–variance mapping optimization algorithm incorporating local search and multi-parent crossover strategies (Rueda and Erlich, 2013).
(6) SPSRDEMMS, which is structured population size reduction differential evolution with multiple mutation strategies (Zamuda et al., 2013).
(7) b6e6rl-CDE, which combines 12 different differential evolution strategies and parameter settings (Tvrdik and Polakova, 2013).

Table 5 shows the results of the Holm multiple comparison test between SADE/BBO-A as the control algorithm, and all other algorithms, including the other hybrid algorithms proposed in this paper and the 2013 CEC algorithms.

As we see in Table 5, for the 2013 CEC benchmark functions, NBIPOP-aCMA-ES is the best algorithm with an average rank of 6.37, iCMAES-ILS is the second best with an average rank of 6.64, and our newly proposed SaDE/BBO-A algorithm is the third best with an average rank of 6.68. Although SaDE/BBO-A is not the best, it is in the top three algorithms. If we add some state-of-the-art operators to the algorithm, we might be able to improve performance. Furthermore, Table 5 shows statistically significant differences between SADE/BBO-A and all other algorithms except NBIPOP-aCMA-ES and iCMAES-ILS, as indicated by p-values smaller than 0.05. The larger p-values for NBIPOP-aCMA-ES and iCMAES-ILS, which are 0.34785 and 0.55134

Table 5
Holm multiple comparison test results of the hybrid EAs and the 2013 CEC benchmark algorithms, which shows the average rank and the p-values. SaDE/BBO-A is the control algorithm, and its average rank is 6.68 based on the Friedman test (not shown in the table).

Algorithm	Rank	p-value	Algorithm	Rank	p-Value	Algorithm	Rank	p-Value
SGA/BBO-I	16.89	0.00008	PSO2011/BBO-A	10.89	0.00755	NBIPOP-aCMA-ES	6.37	0.34785
SGA/BBO-A	14.61	0.00072	LPSO/BBO-I	10.96	0.00704	DRMA-LSch-CMA	8.46	0.02883
CMA-ES/BBO-I	12.77	0.00399	LPSO/BBO-A	9.64	0.02077	SHADE	8.35	0.02270
CMA-ES/BBO-A	9.96	0.01001	CPSO/BBO-I	10.21	0.00780	MVMO-SH	8.69	0.02446
SaDE/BBO-I	7.21	0.08426	CPSO/BBO-A	9.17	0.01115	SPSRDEMMS	9.85	0.01018
PSO2011/BBO-I	11.21	0.00479	iCMAES-ILS	6.64	0.55134	b6e6rl-CDE	11.38	0.00424

Table 6
Comparisons of the best error values of the 2013 CEC benchmark functions for SaDE/BBO-A with sinusoidal migration, generalized migration with $\delta = 0.5$, and a mutation rate of 0.1. Here $[a \pm b]$ indicates the mean value and the corresponding standard deviation, and the best result in each row is shown in bold font.

	SaDE/BBO-A	SaDE/BBO-A with sinusoidal migration	SaDE/BBO-A with $\delta = 0.5$	SaDE/BBO-A with mutation rate = 0.1
F1	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**
F2	6.73E+02 ± 2.37E+01	**1.05E+02 ± 4.21E+01**	2.55E+02 ± 3.29E+01	3.18E+02 ± 7.16E+01
F3	9.12E−02 ± 4.28E−02	**1.09E−02 ± 4.87E−03**	5.24E−02 ± 7.52E−03	3.26E−02 ± 6.88E−03
F4	7.73E−10 ± 1.02E−11	2.36E−09 ± 1.15E−10	**6.28E−10 ± 3.04E−11**	2.81E−02 ± 1.86E−03
F5	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**	**0.00E+00 ± 0.00E+00**
F6	6.77E−04 ± 2.11E−05	**1.54E−04 ± 7.98E−05**	7.56E−04 ± 7.21E−05	3.22E−03 ± 7.14E−04
F7	**5.14E+00 ± 6.42E−01**	2.72E+01 ± 6.35E+00	9.04E+00 ± 3.28E−01	7.18E+00 ± 2.44E−01
F8	8.01E+00 ± 9.03E−01	7.31E+00 ± 3.72E−01	7.85E+00 ± 4.16E−01	**6.32E+00 ± 2.37E−01**
F9	6.65E+00 ± 2.17E−01	9.54E−01 ± 2.16E−02	**2.47E−01 ± 2.36E−02**	1.21E+00 ± 5.98E−01
F10	8.93E+00 ± 2.93E−01	1.33E+01 ± 5.41E+02	1.24E+01 ± 3.22E+02	**2.19E+00 ± 4.26E−01**
F11	**4.24E+00 ± 1.18E+00**	8.14E+00 ± 3.22E+00	6.18E+00 ± 2.31E+00	7.24E+00 ± 6.33E+00
F12	8.00E+00 ± 4.66E−02	**4.33E+00 ± 7.18E−01**	5.25E+00 ± 7.19E−01	5.54E+00 ± 1.79E+01
F13	1.92E−01 ± 8.93E−02	4.26E+00 ± 3.14E−01	3.79E+00 ± 5.12E−01	**1.03E−01 ± 8.58E−01**
F14	**2.17E+01 ± 3.34E+00**	4.51E+01 ± 3.47E+00	6.32E+01 ± 7.26E+00	7.19E+01 ± 3.35E+00
F15	1.11E+02 ± 2.54E+01	**1.99E+01 ± 4.25E+00**	3.87E+01 ± 5.90E+00	5.17E+02 ± 4.65E+01
F16	3.43E+01 ± 7.65E+00	**2.16E+00 ± 4.50E−01**	4.25E+00 ± 3.87E−01	8.76E+01 ± 4.25E+00
F17	5.67E+01 ± 4.26E−01	2.77E+00 ± 4.25E−01	**1.42E+00 ± 7.55E−01**	3.78E+02 ± 4.16E+01
F18	8.98E+01 ± 5.38E+00	7.54E+02 ± 2.16E+01	9.58E+02 ± 2.47E+01	**6.35E+01 ± 3.29E+00**
F19	**1.65E+01 ± 9.87E+00**	3.21E+01 ± 4.39E+00	7.54E+01 ± 2.18E+00	8.46E+01 ± 5.01E+00
F20	**1.90E+01 ± 1.22E+00**	4.16E+01 ± 3.42E+00	2.37E+01 ± 7.11E+00	2.35E+01 ± 4.86E+00
F21	8.73E+02 ± 4.34E+01	2.60E+02 ± 1.54E+01	**2.27E+01 ± 9.19E+00**	3.74E+01 ± 1.18E+00
F22	9.56E+02 ± 8.08E+01	7.41E+01 ± 3.56E+00	4.16E+01 ± 5.59E+00	**2.70E+01 ± 5.13E+00**
F23	**3.62E+02 ± 5.76E+01**	9.20E+02 ± 7.58E+01	7.62E+02 ± 6.00E+01	7.26E+02 ± 5.43E+01
F24	2.11E+02 ± 1.64E+00	3.05E+02 ± 4.17E+01	1.43E+02 ± 6.38E+01	**1.03E+02 ± 7.75E+01**
F25	6.84E+02 ± 8.73E+01	4.17E+01 ± 5.98E+00	**3.36E+01 ± 1.15E+00**	4.33E+02 ± 3.24E+01
F26	8.63E+02 ± 1.29E+01	**1.06E+01 ± 3.47E+00**	2.39E+01 ± 4.44E+00	5.70E+02 ± 1.25E+01
F27	9.12E+02 ± 7.06E+01	5.47E+01 ± 3.26E+00	**4.26E+01 ± 7.98E+00**	8.14E+02 ± 3.29E+01
F28	9.12E+02 ± 9.18E+01	7.71E+02 ± 2.88E+01	**5.38E+02 ± 4.46E+01**	7.11E+03 ± 6.37E+02

respectively, indicate that although NBIPOP-aCMA-ES and iCMAES-ILS obtain better performance than SaDE/BBO-A, the difference is not statistically significant.

4.4. Robustness tests

EAs often show sensitivity to variations in tuning parameters. In this section, we perform some robustness studies on the best hybrid algorithm, SaDE/BBO-A. We consider only the tuning parameters of BBO since SaDE is adapted according to the learning progress. Here we investigate the sinusoidal migration curve as suggested by Ma and Simon (2011b), the generalized migration operator with $\delta = 0.5$ in (1), and a mutation rate of 0.1. The benchmark functions and all other parameters are the same as those in the previous experiments, and the results are shown in Table 6.

According to Table 6, SaDE/BBO-A with sinusoidal migration performs best on 7 functions (F2, F3, F6, F12, F15, F16, and F26), SaDE/BBO-I with generalized migration with $\delta = 0.5$ performs best on 7 functions (F4, F9, F17, F21, F25, F27, and F28), SaDE/BBO-A with the mutation rate of 0.1 performs best on 6 functions (F8, F10, F13, F18, F22, and F24), and standard SaDE/BBO-A performs best on 6 functions (F7, F11, F14, F19, F20, and F23). For functions F1 and F5, all algorithms find the optimal solution. The results show that SaDE/BBO-A with sinusoidal migration, and SaDE/BBO-A with generalized migration with $\delta = 0.5$ have the same performance. They are slightly better than standard SaDE/BBO-A, and SaDE/BBO-A with the mutation rate of 0.1. These results indicate that SaDE/BBO-A tuning parameters can influence performance, but in general the effect is not significant.

4.5. Real-world applications to traveling salesman problems

In this section we apply the proposed hybrid algorithms to the traveling salesman problem (TSP), which is an important and representative real-world combinatorial problem because it is simple to state but difficult to solve, and because many combinatorial problems can be reduced to a TSP. The closed TSP can be simply described as follows. The salesman is required to find the shortest tour connecting all cities, but he must visit each city only once, and he must return to the original city. There is much literature that discusses TSP algorithms using various hybrid EAs (Mo and Xu, 2010; Nguyen et al., 2007; Simon et al., 2011). Here we use the inver-over operator (Tao and Michalewicz, 1998), which has proven to be an effective crossover operator for the TSP. Details discussing the combination of EAs with the inver-over operator can be found in the literature (Simon et al., 2011). We use seven TSP benchmarks, including Bays-29, Berlin-52, St-70, Ch-

Table 7
Comparison of the cost (traveling distance) for all hybrid algorithms on TSP benchmarks. The numbers show the best performance, averaged over 25 simulations. The best performance in each row is shown in bold font.

	SGA/ BBO-I	SGA/ BBO-A	CMA-ES/ BBO-I	CMA-ES/ BBO-A	SaDE/ BBO-I	SaDE/ BBO-A	PSO2011/ BBO-I	PSO2011/ BBO-A	LPSO/ BBO-I	LPSO/ BBO-A	CPSO/ BBO-I	CPSO/ BBO-I
Bays-29	2147	2119	2098	2074	**2023**	2029	2055	2043	2064	2058	2050	2046
Berlin-52	8944	8503	8825	8512	8234	**7924**	9470	9332	9547	9324	8712	8632
St-70	1982	1746	1325	1198	**965**	976	1785	1324	2020	2043	1975	1864
Ch-130	7561	6919	6356	6117	6175	**6114**	6335	6781	6744	6754	6915	6633
Brg-180	2395	2087	2244	1976	1973	**1962**	2371	2045	2335	2006	2376	2043
Rat-575	9725	9114	9436	8121	7562	**7475**	8452	8119	8636	8746	8227	8075
D-1291	77141	65790	80143	75117	64521	**57337**	73126	67149	74575	70891	72009	65432

130, Brg-180, Rat-575, and D-1291, all of which are available in Reinelt (2008). Note that the number in each benchmark label indicates the number of cities in the problem; for example, the D-1291 problem includes 1291 cities. The parameters used in the hybrid algorithms here are the same as those in Section 4.1.

Table 7 shows comparisons of the cost (traveling distance) after 10,000 generations, averaged over 25 simulations. The results show that SaDE/BBO-A obtains the best cost for five of the problems, and the second best cost for the other two problems. This indicates that SaDE/BBO-A is significantly better than the other hybrid algorithms for the TSP benchmarks we study. These results are also consistent with the continuous benchmark function results in Section 4, which also found that SaDE/BBO-A was the best hybrid algorithm.

5. Conclusion

We proposed a new EA hybridization strategy based on information exchange mechanisms from biogeography. We then used the new approach to hybridize several popular EAs, and we tested the hybrid EAs on the continuous optimization benchmark functions from the 2013 Congress on Evolutionary Computation (CEC). The proposed hybridizations included algorithm-level hybridization and iteration-level hybridization, both of which have a simple structure. The new hybrid algorithms make use of the optimization ability of the recently developed EAs, augmented with the information exchange mechanism of biogeography for improved performance. The test results showed that the proposed hybrids significantly outperformed their constituent algorithms with the selected tuning parameters and generation limits, and algorithm-level hybridization was slightly better than iteration-level hybridization for the continuous benchmark functions that we studied. Statistical tests showed that SaDE/BBO-A was at least the third best algorithm when compared to the 2013 CEC algorithms, and was statistically even with the best two algorithms.

We applied our proposed hybrid EAs to the traveling salesman problem, and the results showed that our proposed SaDE/BBO-A is the best hybrid algorithm for the real-world problems that we tested. We also studied the tuning parameters of SaDE/BBO-A to confirm that its performance is relatively robust to tuning parameters.

For future work, we suggest several important directions. First, future work could include testing on additional benchmark functions, testing with higher dimensions, testing on noisy functions, and comparing the proposed hybrid algorithms with additional EAs. The second direction for future work is to develop and study the performance of other biogeography-based hybridization algorithms on the continuous optimization benchmark functions in this paper. The third direction for future work is to apply the proposed hybrid EAs to more real-world problems. The fourth direction for future work is to incorporate additional state-of-art operators to our proposed hybrid algorithms to obtain better performance.

Finally, we note that just as ideas from BBO have been used in this paper to develop new hybridization approaches, ideas from other EAs could also be used to develop new hybridization approaches. For example, information exchange mechanisms based on DE, PSO, SGA, or any other EA, could be used to combine EAs running in parallel. The use of BBO as a hybridization strategy should motivate the investigation of other EAs as hybridization strategies, and then these hybridization strategies could be compared with one another in future work.

Acknowledgment

This material was supported in part by the National Science Foundation under Grant no. 0826124, the National Natural Science Foundation of China under Grant nos. 61305078, 61179041 and the Shaoxing City Public Technology Applied Research Project under Grant no. 2013B70004. The suggestions of the anonymous reviewers were helpful in improving this paper after its original submission to the journal.

References

Ahn, C., 2006. Advances in Evolutionary Algorithms: Theory, Design and Practice. Springer Publishing, New York

Beyer, H.G., 1994. Toward a theory of evolution strategies: the (μ, λ)-theory. Evol. Comput. 2 (4), 381–407.

Beyer, H.G., Sendhoff, B., 2008. Covariance matrix adaptation revisited—the CMSA evolution strategy. Lecture Notes Comput. Sci. 5199, 123–132.

Bhattacharya, A., Chattopadhyay, P., 2010. Hybrid differential evolution with biogeography-based optimization for solution of economic load dispatch. IEEE Trans. Power Syst. 25 (4), 1955–1964.

Blum, C., Puchinger, J., Raidl, G., Roli, A., 2011. Hybrid metaheuristics in combinatorial optimization: a survey. Appl. Soft Comput. 11 (6), 4135–4151.

Boussaïd, I., Chatterjee, A., Siarry, P., Ahmed-Nacer, M., 2011. Two-stage update biogeography based optimization using differential evolution algorithm (DBBO). Comput. Oper. Res. 38 (8), 1188–1198.

Boussaïd, I., Chatterjee, A., Siarry, P., Ahmed-Nacer, M., 2012. Biogeography based optimization for constrained optimization problems. Comput. Oper. Res. 39 (12), 3293–3304.

Boussaïd, I., Chatterjee, A., Siarry, P., Ahmed-Nacer, M., 2013a. A comparative study of modified BBO variants and other metaheuristics for optimal power allocation in wireless sensor networks. In: Chatterjee, A., Nobahari, H., Siarry, P. (Eds.), Advances in Heuristic Signal Processing and Applications, pp. 79–110

Boussaïd, I., Chatterjee, A., Siarry, P., Ahmed-Nacer, M., 2013b. Hybrid BBO-DE algorithms for fuzzy entropy-based thresholding. In: Chatterjee, A., Siarry, P. (Eds.), Computational Intelligence in Image Processing, pp. 37–69In: Chatterjee, A., Siarry, P. (Eds.), Computational Intelligence in Image Processing, pp. 37–69

Boussaïd, I., Chatterjee, A., Siarry, P., Ahmed-Nacer, M., 2011. Hybridizing biogeography-based optimization with differential evolution for optimal power allocation in wireless sensor networks. IEEE Trans. Veh. Technol. 60 (5), 2347–2353.

Bratton, D., Kennedy, J., 2007. Defining a standard for particle swarm optimization. In: Proceedings of the IEEE Swarm Intelligence Symposium. pp. 120–127.

Chatterjee, A., Siarry, P., 2006. Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization. Comput. Oper. Res. 33, 859–871.

Chatterjee, A., Siarry, P., Nakib, A., Blanc, R., 2012. An improved biogeography based optimization approach for segmentation of human head CT-scan images employing fuzzy entropy. Eng. Appl. Artif. Intell. 25, 1698–1709.
Clerc, M., Kennedy, J., 2002. The particle swarm—explosion, stability and convergence in a multidimensional complex space. IEEE Trans. Evol. Comput. 6 (1), 58–73.
Das, S., Suganthan, P.N., 2011. Differential evolution—a survey of the state-of-the-art. IEEE Trans. Evol. Comput. 15 (1), 4–31.
Das, S., Maity, S., Qu, B.-Y., Suganthan, P., 2011. Real-parameter evolutionary multimodal optimization—a survey of the state-of-the-art. Swarm Evol. Comput. 1 (2), 71–88.
Demsar, J., 2006. Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. 7 (1), 1–30.
Derrac, J., Garcia, S., Molina, D., Herrera, F., 2011. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol. Comput. 1 (1), 3–18.
Dorigo, M., Gambardella, L.M., 1997. Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans. Evol. Comput. 1 (1), 53–66.
Dorigo, M., Maniezzo, V., Colorni, A., 2002. Ant system: optimization by a colony of cooperating agents. IEEE Trans. Syst., Man, and Cybern.—Part B 26 (1), 29–41.
Dunn, O.J., 1961. Multiple comparisons among means. J. Am. Stat. Assoc. 56, 52–56.
Eberhart, R.C., Shi, Y., 2000. Comparing inertia weights and constriction factors in particle swarm optimization. In: Proceedings of the IEEE Congress on Evolutionary Computation. pp. 84–88.
Friedman, M., 1940. A comparison of alternative tests of significance for the problem of m rankings. Ann. Math. Stat. 11 (1), 86–92.
Gen, M., Lin, L. Multiobjective evolutionary algorithm for manufacturing scheduling problems: state-of-the-art survey. J. Intell. Manuf., http://dx.doi.org/10.1007/s10845-013-0804-4, in press.
Hansen, N., 2006. The CMA evolution strategy: a comparing review. In: Lozano, J.A., Larrañaga, P., Inza, I., Bengoetxea, E. (Eds.), Towards a New Evolutionary Computation: Advances in Estimation of Distribution Algorithms. Springer, New York, pp. 75–102
Hansen, N., Müller, S.D., Koumoutsakos, P., 2003. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evol. Comput. 11 (1), 1–18.
Hochberg, Y., Tamhane, A., 1987. Multiple Comparison Procedures. John Wiley & Sons, Hoboken, NJ.
Hofmeyr, S., Forrest, S., 2000. Architecture for an artificial immune system. Evol. Comput. 8 (4), 443–473.
Jaimes, A.L., Coello Coello, C.A., 2005. MRMOGA: parallel evolutionary multiobjective optimization using multiple resolutions. In: Proceedings of the IEEE Congress on Evolutionary Computation. pp. 2294–2301.
Jamuna, K., Swarup, K., 2012. Multi-objective biogeography based optimization for optimal PMU placement. Appl. Soft Comput. 12 (5), 1503–1510.
Karaboga, D., Basturk, B., 2007. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J. Glob. Optimiz. 39 (3), 459–471.
Khatib, W., Fleming, P., 1998. The stud GA: a mini revolution?. In: Eiben, A., Back, T., Schoenauer, M., Schwefel, H. (Eds.), Parallel Problem Solving from Nature. Springer, New York, pp. 683–691
Lacroix, B., Molina, D., Herrera, F., 2013. Dynamically updated region based memetic algorithm for the 2013 CEC special session and competition on real parameter single objective optimization. In: Proceedings of the IEEE Congress on Evolutionary Computation. pp. 1945–1951.
Lassig, J., Sudholt, D., 2010. The benefit of migration in parallel evolutionary algorithms. In: Proceedings of the Genetic and Evolutionary Computation Conference. pp. 1105–1112.
Liang, C., Huang, Y., Yang, Y., 2009. A quay crane dynamic scheduling problem by hybrid evolutionary algorithm for berth allocation planning. Comput. Ind. Eng. 56 (3), 1021–1028.
Liang, J.J., Qu, B.Y., Suganthan, P.N., Hernández-Díaz, Alfredo G., 2013. Problem definitions and evaluation criteria for the CEC 2013 special session and competition on real-parameter optimization. Technical Report.
Liao, T., Stuetzle, T., 2013. Benchmark results for a simple hybrid algorithm on the CEC 2013 benchmark set for real parameter optimization. In: Proceedings of the IEEE Congress on Evolutionary Computation. pp. 1938–1944.
Lin, L., Gen, M., Wang, X., 2009. Integrated multistage logistics network design by using hybrid evolutionary algorithm. Comput. Ind. Eng. 56 (3), 854–873.
Loshchilov, I., 2013. CMA-ES with restarts for solving CEC 2013 benchmark problems. In: Proceedings of the IEEE Congress on Evolutionary Computation. pp. 369–376.
Lozano, M., Garcia-Martinez, C., 2010. Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification: overview and progress report. Comput. Oper. Res. 37 (3), 481–497.
Ma, H., 2010. An analysis of the equilibrium of migration models for biogeography-based optimization. Inf. Sci. 180 (18), 3444–3464.
Ma, H., Simon, D., 2011a. Blended biogeography-based optimization for constrained optimization. Eng. Appl. Artif. Intell. 24 (3), 517–525.
Ma, H., Simon, D., 2011b. Analysis of migration models of biogeography-based optimization using Markov theory. Eng. Appl. Artif. Intell. 24 (6), 1052–1060.
Makeyev, O., Sazonov, E., Moklyachuk, M., Logez-Meye, P., 2010. Hybrid evolutionary algorithm for microscrew thread parameter estimation. Eng. Appl. Artif. Intell. 23, 446–452.
Mo, H., Xu, L., 2010. Biogeography migration algorithm for traveling salesman problem. In: Tan, Y., Shi, Y., Tan, K.-C. (Eds.), Advances in Swarm Intelligence, pp. 405–414
Mongus, D., Repnik, B., Mernik, M., Zalik, B., 2012. A hybrid evolutionry algorithm for tuning a cloth-simulation model. Appl. Soft Comput. 12, 266–273.
Neri, F., Cotta, C., 2012. Memetic algorithms and memetic computing optimization: a literature review. Swarm Evol. Comput. 2, 1–14.
Nguyen, H.D., Yoshihara, I., Yamamori, K., Yasunaga, M., 2007. Implementation of an effective hybrid GA for large-scale traveling salesman problems. IEEE Trans. Syst., Man, Cybern. 37 (1), 92–99.
Niknam, T., 2009. An efficient hybrid evolutionary algorithm based on PSO and HBMO algorithms for multi-objective distribution feeder reconfiguration. Energy Convers. Manag. 50 (8), 2074–2082.
Niknam, T., Farsani, E.A., 2010. A hybrid self-adaptive particle swarm optimization and modified shuffled frog leaping algorithm for distribution feeder reconfiguration. Eng. Appl. Artif. Intell. 23, 1340–1349.
Omran, M., Clerc, M., 2011. Particle Swarm Central. ⟨http://www.particleswarm.info/⟩.
Pedro, L., Lozano, J.A., 2002. Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. Kluwer Academic Publishers, Norwell, MA.
Prodhon, C., 2011. A hybrid evolutionary algorithm for the periodic location-routing problem. Eur. J. Oper. Res. 210 (2), 204–212.
Rahmati, S., Zandieh, M., 2012. A new biogeography-based optimization (BBO) algorithm for the flexible job shop scheduling problem. Int. J. Adv. Manuf. Technol. 58 (9), 1115–1129.
Reinelt, G., 2008. TSPLIB. ⟨http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95⟩.
Rueda, J., Erlich, I., 2013. Mean–variance mapping optimization for solving the IEEE-CEC 2013 competition problems. In: Proceedings of the IEEE Congress on Evolutionary Computation. pp. 1664–1671.
Shi, Y., Eberhart, R.C., 1998. Parameter selection in particle swarm optimization. Evolutionary Programming VII. Lecture Notes Comput. Sci. 1447, 591–600.
Simon, D., 2008. Biogeography-based Optimization. IEEE Trans. Evol. Comput. 12 (6), 702–713.
Simon, D., 2011. A dynamic system model of biogeography-based optimization. Appl. Soft Comput. 11, 5652–5661.
Simon, D., 2013. Evolutionary Optimization Algorithms. John Wiley & Sons, Hoboken, NJ.
Simon, D., Rarick, R., Ergezer, M., Du, D., 2011. Analytical and numerical comparisons of biogeography-based optimization and genetic algorithms. Inf. Sci. 181, 1224–1248.
Singh, U., Singla, H., Kamal, T., 2010. Design of Yagi-Uda antenna using biogeography based optimization. IEEE Trans. Antennas Propag. 58 (10), 3375–3379.
Suganthan, P.N. ⟨http://www3.ntu.edu.sg/home/EPNSugan/⟩.
Tanabe, R., Fukunaga, A., 2013. Evaluating the performance of SHADE on CEC 2013 benchmark problems. In: Proceedings of the IEEE Congress on Evolutionary Computation. pp. 1952–1959.
Tao, G., Michalewicz, Z., 1998. Inver-over operator for the TSP. In: Eiben, A., Back, T., Schoenauer, H., Schwefel, H. (Eds.), Parallel Problem Solving from Nature—PPSN, pp. 803–812
Tvrdik, J., Polakova, R., 2013. Competitive differential evolution applied to CEC 2013 problems. In: Proceedings of the IEEE Congress on Evolutionary Computation. pp. 1651–1657.
Wang, L., Xu, Y., 2011. An effective hybrid biogeography-based optimization algorithm for parameter estimation of chaotic systems. Expert Syst. Appl. 38 (12), 15103–15109.
Wang, Y., Cai, Z., Zhou, Y., Fan, Z., 2009. Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique. Struct. Multidiscip. Optimiz. 37 (4), 395–413.
Wolpert, D.H., Macready, W.G., 1997. No free lunch theorems for optimization. IEEE Trans. Evol. Comput. 1 (1), 67–82.
Yao, X., Liu, Y., Lin, G., 1999. Evolutionary programming made faster. IEEE Trans. Evol. Comput. 3 (2), 82–102.
Zamuda, A., Brest, J., Mezura-Montes, E., 2013. Structured population size reduction differential evolution with multiple mutation strategies on CEC 2013 real parameter optimization. In: Proceedings of the IEEE Congress on Evolutionary Computation. pp. 1925–1931.
Zhao, S.Z., Suganthan, P.N., Das, S., 2011. Self-adaptive differential evolution with multi-trajectory search for large-scale optimization. Soft Comput. 15 (11), 2175–2185.