Appl Intell
DOI 10.1007/s10489-016-0848-1

A novel disruption in biogeography-based optimization with application to optimal power flow problem

Jagdish Chand Bansal[1] **· Pushpa Farswan**[1]

Abstract Biogeography-based optimization (BBO) is an emerging meta-heuristic algorithm. Due to ease of implementation and very few user-dependent parameters, BBO gained popularity among researchers. The performance of BBO is highly dependent on its two operators, migration and mutation. The performance of BBO can be significantly improved by either modifying these operators or by introducing a new operator into it. This paper proposes a new operator, namely the disruption operator to improve the capability of exploration and exploitation in BBO. The proposed DisruptBBO (DBBO) has been tested on well-known benchmark problems and compared with various versions of BBO and other state-of-the-art metaheuristics. The experimental results and statistical analyses confirm the superior performance of the proposed DBBO in solving various nonlinear complex optimization problems. The proposed algorithm has also been applied to the optimal power flow optimization problem from the electrical engineering background.

Keywords Disruption operator · Migration operator · Optimal power flow · Biogeography-based optimization

✉ Jagdish Chand Bansal
jcbansal@gmail.com

Pushpa Farswan
pushpafarswan6@gmail.com

[1] South Asian University, New Delhi, India

1 Introduction

Population-based meta-heuristic algorithms have gained popularity in recent years. Requirements of techniques for more accurate and efficient solutions to real life optimization problems motivate researchers to develop new meta-heuristic algorithms. Biogeography-based optimization (BBO) algorithm is a recent addition to the area of meta-heuristic algorithms. It is inspired by the migration phenomenon of species among various habitats. It was introduced by Dan Simon in 2008 [25].

BBO gained attention of researchers because of few parameters to tune and the relatively higher efficiency it achieves. There have been many improvements to the original BBO algorithm by implementing improved migration and mutation. In [15], Ma et al. proposed Blended BBO (B-BBO). In B-BBO, a migration operator combines the features of both immigrating and emigrating islands and is employed to handle constrained optimization problems. In [26], Simon et al. proposed a Linearized BBO (LBBO) for improving the solution of non-separable problems. LBBO combined with a periodic re-initialization and local search operator developed an algorithm for global optimization in continuous search space. In [14], Lohakare et al. proposed a memetic BBO named as aBBOmDE. In aBBOmDE, the performance of BBO is accelerated with the modified DE (differential evolution), so that the convergence speed is enhanced and exploitation is achieved by using an original migration operator. In [7], Feng et al. proposed an Improved BBO (IBBO) by integrating a new migration operator and introducing some other implementations. Here the improved migration operator simultaneously adopts more information from emigrating habitats as well as from two other randomly selected habitats. That maintains population diversity and preserves exploitation ability. In [29], Xiong et al. propose an enhanced BBO variant, called Polyphyletic BBO (POLBBO) by using a polyphyletic migration operator that formally utilizes as many as four individuals informations to construct a new solution vector that is expected to increase the population diversity. In [28], Wang et al. introduced the krill herd migration operator in the BBO algorithm to exploit the search region thoroughly. In [3], Bansal proposed a modified version of the BBO with modified blended crossover and polynomial mutation (BBO-MBLX-PM). The performance of the BBO-MBLX-PM algorithm has shown to be more successful compared to the original BBO algorithm. In [10], Guo et al. introduce a backtracking BBO (BBBO) algorithm. The BBBO algorithm uses external population information for generating new population and improving the exploration ability of the algorithm. The most recent survey on BBO can found in [9], Guo et al. and [8], Garg et al.

Although, various proposed variants have shown improvements in the performance of the original BBO. This observation leads to explore BBO for further improvements. This has been tried in this paper by incorporating a disruption operator in BBO. This paper implements a disruption operator in BBO for better exploration and exploitation search capabilities of the algorithm.

Disruption operator is defined in Section 3. Disruption operator has already been successfully applied in the Gravitational search algorithm (GSA) [21] and Bare-bones Particle Swarm Optimization [13]. In [21], Sarafrazi et al. proposed Improved gravitational search algorithm (IGSA). In IGSA, the new operator (disruption operator) is able to improve the ability of GSA to explore further and exploit the search region. Here searching starts with exploration and as the time passes it switches to exploitation. Recently the same operator is applied in bare-bones particle swarm optimization by Liu et al. [13] and proposed the disruption bare-bones particle swarm optimization (DBPSO) algorithm to overcome the lack of diversity and premature convergence. The proposed DBPSO is able to enhance the diversity along with speed up the convergence rate of Bare-bones particle swarm optimization (BPSO). In DBPSO, the i^{th} position of particle is changed, if the ratio of distance between the i^{th} particle and the particle whose fitness is just better than i, to the distance between i^{th} particle and best particle is less than the specific threshold.

This paper also establishes the proposed DisruptBBO (DBBO) by applying it to an electrical engineering optimization problem, optimal power flow (OPF).

Rest of the paper is organized as follows: The details of BBO are described in Section 2. The implementation of the disruption operator in BBO is explained in Section 3. Through various experiments and statistical analysis, the proposed DBBO is tested over test problems in Section 4. In Section 5, the proposed DBBO is applied to an optimal power flow problem. Section 6 concludes the paper.

2 Biogeography-based optimization

The well known method of studying geographical distribution of biological organisms is biogeography, whose earliest works can be traced back to the days by Alfred Wallace and Charles Darwin [25]. The mathematical model of biogeography was brought in the picture by Robert Mac Arthur and Edward Wilson. They considered the migration of species from one island to another island, the arrival of new species and the extinction of some existing species [17]. Recently a new evolutionary population-based optimization technique has been proposed which is based on the basic nature of biogeography. It has been named biogeography-based optimization (BBO) [25]. However whereas the study of biogeography considers evolution, migration and extinction, BBO is inspired by only migration of species among islands. In a biogeography model, the fitness of a geographical area is assessed by a habitat suitability index (called HSI). Habitats which are more suitable for species to reside are said to have a high HSI. Similarly, habitats which are less suitable for species to reside are said to have low a HSI. In this way high HSI habitats house a relatively larger number of species. The characterization of habitability is called the suitability index variable. Rainfall, vegetation, temperature, etc., are called suitability index variables ($SIVs$). The migration of species among different habitats is mainly controlled by two parameters, immigration rate (λ) and emigration rate (μ). λ and μ are functions of the number of species in a habitat. $P_s(t)$ is the probability that there are s species in the habitat at any time t.

$$P_s(t+\Delta t) = P_s(t)(1-\lambda_s\Delta t-\mu_s\Delta t) + P_{s-1}\lambda_{s-1}\Delta t + P_{s+1}\mu_{s+1}\Delta t \tag{1}$$

Where λ_s is the immigration rate when there are s species in the habitat. μ_s is the emigration rate when there are s species in the habitat.

At time t+Δt one of the following conditions must hold for s species in the habitat.

1. If there are s species in the habitat at time t then there will be no immigration and no emigration of species within time t and t+Δt.

2. If there are $(s-1)$ species in the habitat at time t then one species will immigrate between time t and $t+\Delta t$.
3. If there are $(s+1)$ species in the habitat at time t then one species will emigrate between time t and $t+\Delta t$.

For ignoring the probability of more than one immigration or emigration, Δt is assumed to be very small. Taking the limit as $\Delta t \longrightarrow 0$

$$\dot{P}_s = \begin{cases} -(\lambda_s+\mu_s)P_s + \mu_{s+1}P_{s+1}, & s=0 \\ -(\lambda_s+\mu_s)P_s + \lambda_{s-1}P_{s-1} + \mu_{s+1}P_{s+1}, & 1 \le s \le s_{max}-1 \\ -(\lambda_s+\mu_s)P_s + \lambda_{s-1}P_{s-1}, & s = s_{max} \end{cases} \tag{2}$$

We can obtain a matrix relation exhibiting the dynamic equations of the probabilities of the number of species in the habitat as

$$\begin{bmatrix} \dot{P}_0 \\ \dot{P}_1 \\ \vdots \\ \vdots \\ \dot{P}_{S_{max}} \end{bmatrix} = \begin{bmatrix} -(\lambda_0+\mu_0) & \mu_1 & 0 & \cdots & 0 \\ \lambda_0 & -(\lambda_1+\mu_1) & \mu_2 & \cdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \lambda_{n-2} & -(\lambda_{n-1}+\mu_{n-1}) & \mu_n \\ 0 & \cdots & 0 & \lambda_{n-1} & -(\lambda_n+\mu_n) \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ \vdots \\ \vdots \\ P_{S_{max}} \end{bmatrix} \tag{3}$$

The primary concept of biogeography has been used to design a population-based optimization procedure that can be potentially applied to optimize many engineering optimization problems. BBO is based on two simple biogeography concepts (migration and mutation). In the designed BBO algorithm each habitat H has a potential $m \times 1$ vector solution where m is the number of $SIVs$ in each habitat. HSI of each habitat corresponds to fitness function of population-based algorithms. A habitat with the highest HSI reveals that it is the best candidate for the optimum solution among all habitats. It is assumed that the ecosystem constitutes of N_p habitats i.e. the population size is N_p. In the basic BBO algorithm, the immigration and emigration rates vary linearly with the number of species (Fig. 1) and they are calculated using the following formula:

$$\lambda_i = I\left(1 - \frac{k_i}{n}\right) \tag{4}$$

$$\mu_i = E\left(\frac{k_i}{n}\right) \tag{5}$$

Here λ_i stands for the immigration rate of i^{th} individual (island),
μ_i stands for the emigration rate of i^{th} individual (island),
I stands for the maximum possible immigration rate,
E stands for the maximum possible emigration rate,
n stands for the maximum possible number of species that island can support, and
k_i stands for the fitness rank of i^{th} island after sorting the fitness of i^{th} island. Thus, for the worst solution k_i is considered as 1 and for the best solution k_i is considered as n.

It suffices to assume a linear relationship between the number of species and migration rate from the view point of many applications. The relation between migration rate (λ and μ) and the number of species is illustrated in Fig. 1. If there are zero species on the island, then the immigration rate is maximum, denoted by I. If there are maximum number of species (S_{max}) on the island, then the emigration rate is maximum, denoted by E. At the state of an equilibrium, the number of species is denoted by S_0 and in the equilibrium state, the immigration rate and emigration rate are equal. The islands are referred to as high HSI islands if the number of species is above S_0 and the islands are referred as low HSI islands if the number of species is less than S_0. Algorithm 1 describes the pseudo code of BBO.

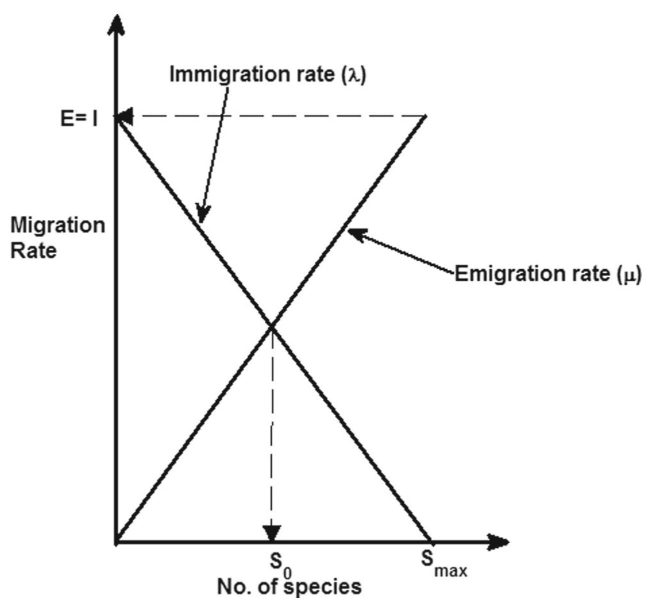


Fig. 1 Relation between number of species and migration rate [25]

Algorithm 1 Biogeography-based optimization algorithm

Initialize the population
Population size $\leftarrow N_p$;
Sort the population according to the increasing order of fitness
Calculate λ and μ
Generation index $\leftarrow GenIndex$;
for $GenIndex = 1$ to $MaxGen$ **do**
  Apply migration
  for $i = 1$ to N_p **do**
    Select habitat H_i according to λ_i
    if $rand(0, 1) < \lambda_i$ **then**
      for $e = 1$ to N_p **do**
        Select habitat H_e according to μ_e
        Replace the selected SIV of H_i by randomly selected SIV of H_e
      end for
    end if
  end for
  Apply mutation
  for $i = 1$ to N_p **do**
    Compute mutation probability $m(S)$
    if $rand(0, 1) < m(S)$ **then**
      Replace $H_i(SIV)$ with randomly generated SIV
    end if
  end for
  Sort the population according to the increasing order of fitness
  Keep the elite solution
  Stop, if termination criteria satisfied
end for

Migration and mutation are two crucial operators in BBO. “Migration” and the “Mutation” procedures evolve new candidate solutions. This procedure of governing the habitats to the “Migration” procedure, followed by the “Mutation” procedure, is continued to the next generation until the termination criteria are satisfied. These criteria can be the maximum number of generations or obtaining the desired solution. The basic concept of migration procedure is to probabilistically share the information between habitats by utilizing the immigration rate (λ_s) and emigration rate (μ_s). The migration operator is same as the crossover operator of the evolutionary algorithm and is responsible for sharing the feature among candidate solutions for modifying fitness. In the migration procedure, the immigrating habitat is selected according to the probability of immigration rate and the emigrating habitat is selected according to the probability of emigration rate of habitats. Then it is probabilistically decided as to which of the SIV of the immigrating habitat needs to be modified. Once the SIV is selected, the algorithm replaces that SIV by the emigrating habitat’s SIV. The other important phenomenon is the mutation. The mutation occurs by sudden changes in islands due to the occurrence of random events. It is responsible for maintaining the diversity of island in the BBO process. Analysis of Fig. 1 reveals that very high HSI solutions and very low HSI solutions have very low probabilities while medium HSI solutions have the relatively high probability to exist as a solution. So mutation gives the same chance to improve low HSI solutions as to high HSI solutions. The mutation rate $mut(i)$ is expressed as:

$$mut(i) = m_{max}\left(1 - \frac{P_i}{P_{max}}\right) \tag{6}$$

where m_{max} is the user defined parameter and $P_{max} = max\{P_i\}$; i=1, 2,...., N_p.

3 Disrupted biogeography-based optimization

3.1 Motivation

This section proposes a modified version of biogeography-based optimization using the disruption operator.

Operator “Disruption”, originated from astrophysics. It is the phenomenon: “When a swarm of gravitationally bound particles having a total mass, m, approaches too close to a massive object of mass M, then the swarm tends to be torn apart. The same can happen to a solid body held together by gravitational forces when it approaches a much more massive object”. It was given by Harwit [11]. To incorporate the disruption phenomenon in BBO, it is assumed that the highest HSI island is the star (best) island, and other islands can potentially disrupt and scatter under the gravity of the star island.

In [21], a new operator named the disruption operator is incorporated in the gravitational search algorithm. Thus the obtained version of GSA outperformed the original GSA and other related algorithms. The disruption operator was introduced in GSA [21]. These facts motivated the authors to develop a suitable disruption operator for BBO as well.

3.2 Disruption operator

In the context of the paper the disruption operator D is defined as below:

$$D = \begin{cases} R_{i,nbd}U(-\frac{1}{2}, \frac{1}{2}), & if\, R_{i,best} < \frac{1}{2}Dia_{GenIndex} \\ R_{i,best}U(-\frac{1}{2}, \frac{1}{2}), & otherwise. \end{cases} \tag{7}$$

In (7), $R_{i,nbd}$ represents the Euclidean distance between island i and the island which is most close to island

i, $U(-\frac{1}{2},\frac{1}{2})$ represents the uniformly distributed random numbers in the interval $[-\frac{1}{2},\frac{1}{2}]$. $Dia_{GenIndex}$ is the maximum Euclidean distance between any two islands at the generation index ($GenIndex$).

3.3 BBO with disruption operator

To maintain the diversity during the search as well as for achieving a good convergence rate, the disruption operator defined in the Section 3.2 is applied in BBO subject to a condition. Since the objective of introducing the disruption operator in BBO is to modify the search strategy with better exploration and exploitation capabilities, therefore, it is obvious that the condition of applying the disruption operator must be dependent on the distances among individuals at the given time in the search space. Therefore at any time, i^{th} solution in BBO is disrupted (modified) using the disruption operator, if the ratio of distances between island i from its nearest neighboring island (other than star island) and the distance between island i from star island is less than predefined threshold C. i.e. In other words, in an iteration, the disruption operator is applied to the i^{th} individual subject to the condition

$$\frac{R_{i,nbd}}{R_{i,best}} < C \tag{8}$$

Where $R_{i,best}$ is the Euclidean distance between i^{th} island and star island. In other words, $R_{i,nbd}$ is the Euclidean distance between i^{th} island and it's nearest island which is necessarily not the star island. The parameter C in (8) is defined as below:

$$C = \theta\left(1 - \frac{GenIndex}{MaxGen}\right) \tag{9}$$

Here, θ is a parameter which is either a constant or a function of generations. $GenIndex$ is the generation index and $MaxGen$ is the maximum number of generations. From condition (8), it is clear that disruption of any individual highly depends on the value of C. Large values of C will provide more opportunities for disruption to appear and hence more opportunities for the exploration. While small values of C will restrict the disruption operator to be implemented very frequently which will ultimately lead to the exploitation of the search space. Therefore, generation wise the decreasing value of C will provide exploration in early generations while refining the solution at later generations. It can be seen from (9), that C is a decreasing function of θ. In [21], a linearly decreasing C is considered as defined in (9) with $\theta = 100$. In [13], to examine the effect of θ on the performance of the proposed algorithm, a number of experiments have been carried out with different values of $\theta = 0.1$, 1, 10, 50 and 100. The experiments of [13] show that $\theta = 10$ performs better than other the choices.

In this paper instead of considering a constant value of θ, a variable θ is adopted (details of θ is given in Section 4.1).

Moreover in the above cited articles, in the definition of the disruption operator D, $R_{i,best}$ is compared with 1, But $R_{i,best}$ depends upon the search space range of the problem and therefore instead of 1 or any other constant, it should be a function of the search space. Therefore in this paper instead of 1, half of the maximum distance between any two islands at any generation ($\frac{1}{2}Dia_{GenIndex}$) is considered for comparison with $R_{i,best}$.

The disruption operator of [21] to any individual was defined as below:

$$X_i(new) = X_i(old) * D \tag{10}$$

This approach of disruption operator implementation has an inherent drawback. Here D is defined as below in (11):

$$D = \begin{cases} R_{i,j}U(-\frac{1}{2},\frac{1}{2}), & if\, R_{i,best} \geq 1 \\ 1 + R_{i,best}U(-\frac{10^{-16}}{2},\frac{10^{-16}}{2}), & otherwise. \end{cases} \tag{11}$$

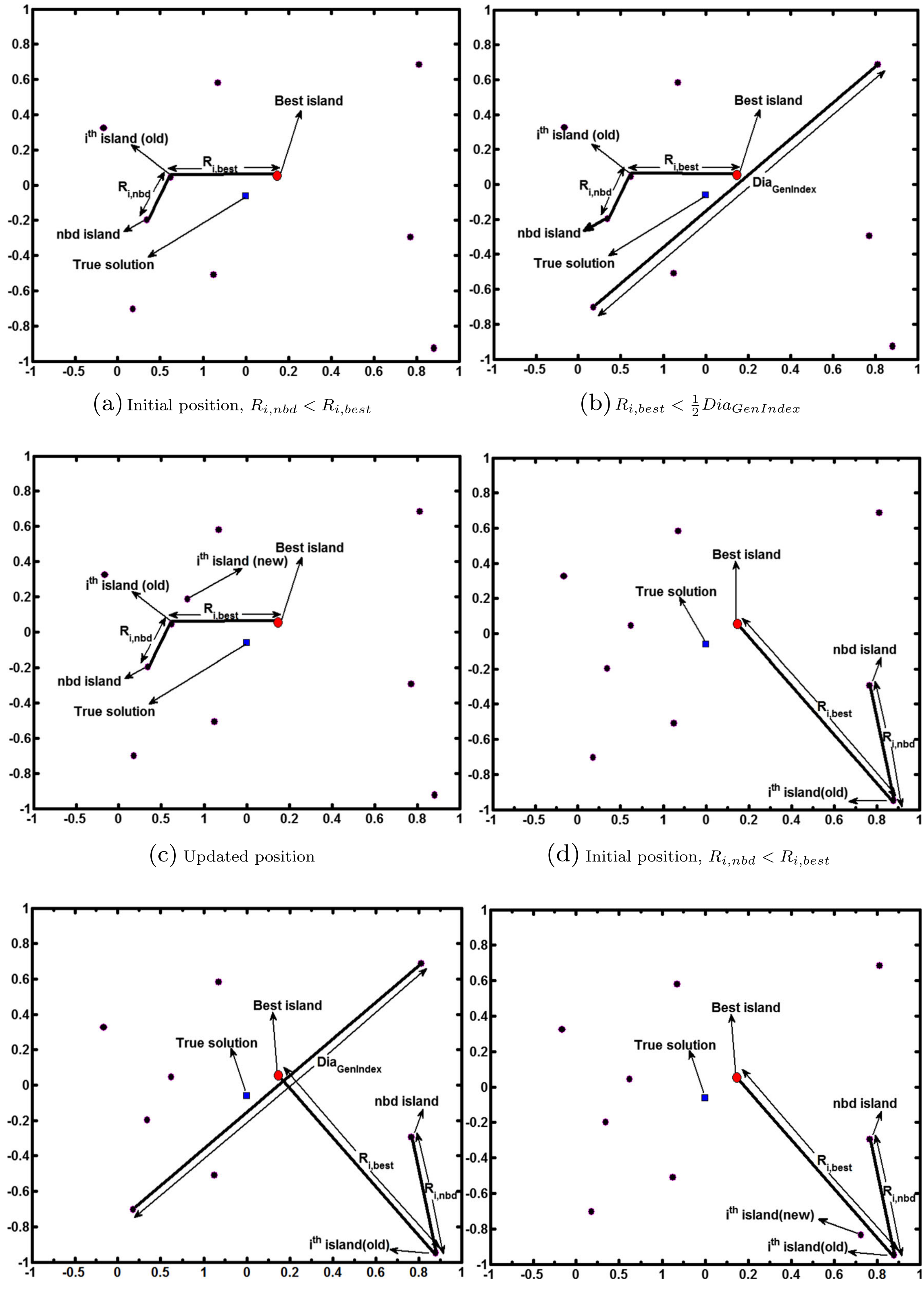
Where $R_{i,j}$ is the Euclidean distance between the i^{th} individual and j^{th} individual. Here j is the nearest neighborhood of the i^{th}individual. It is clear that in the case of very close individuals, the value of D may become less than 1, too frequently. If D is less than 1, then the solution will converge towards the origin. Now, if the optimal solution of the considered problem is at the origin, the algorithm becomes biased and will converge quickly to the origin, but the algorithm will not converge to the near optimal solution if the optimal solution is far from the origin. Similarly in [13], the implementation of disruption have been carried out by multiplying the disruption operator to an individual. Therefore, in this paper for any individual with condition (8) and the value of C given in (9), disruption as defined in (7) is implemented as below:

$$X_i(new) = X_i(old) + D \tag{12}$$

Now, since D provides a disruption to an individual, therefore the value of D must not always be a large or very small quantity.

Since disruption is applied with the condition $\frac{R_{i,nbd}}{R_{i,best}} < C$ and $C < 1$ (see Section 4.1). Thus, $\frac{R_{i,nbd}}{R_{i,best}} < 1$, therefore $R_{i,best} > R_{i,nbd}$.

In order to visualize the effect of the disruption operator on the population at any given time, consider 10 individuals in a 2-dimensional search space (Fig. 2). In Fig. 2, i^{th} individual, best individual, nbd individual, $Dia_{GenIndex}$, $R_{i,nbd}$, $R_{i,best}$ and true solution are marked in the rectangular search space $[-1, 1]$. For explanation of working of the disruption operator, let us consider $U(-\frac{1}{2},\frac{1}{2}) = 0.43$ in (13) and for (14), $U(-\frac{1}{2},\frac{1}{2}) = -0.45$. Now, as can be seen in Fig. 2a



(a) Initial position, $R_{i,nbd} < R_{i,best}$

(b) $R_{i,best} < \frac{1}{2} Dia_{GenIndex}$

(c) Updated position

(d) Initial position, $R_{i,nbd} < R_{i,best}$

(e) $R_{i,best} \geq \frac{1}{2} Dia_{GenIndex}$

(f) Updated position

Fig. 2 Position update process through disruption operator in 2-dimensional search space and d, $R_{i,nbd} < R_{i,best}$, therefore i^{th} individual is selected for disruption. In Fig. 2b, condition $R_{i,best} < \frac{1}{2} Dia_{GenIndex}$ is true and therefore disruption D is applied on the i^{th} individual as given in (13) and the new position of the i^{th} individual is updated as given in (15). Finally after disruption the new position of the i^{th} individual is seen in Fig. 2c. Similarly in Fig. 2e, condition $R_{i,best} \geq \frac{1}{2} Dia_{GenIndex}$ is true. So the disruption D is applied on the i^{th} individual as given in (14) and the new position of the i^{th} individual is updated as given in (15). Finally after disruption the new position of the i^{th} individual is given in Fig. 2f.

$$D = R_{i,nbd} * 0.43,\ Where\ 0.43\ is\ U\left(-\frac{1}{2}, \frac{1}{2}\right) \tag{13}$$

$$D = R_{i,best} * -0.45,\ Where - 0.45\ is\ U\left(-\frac{1}{2}, \frac{1}{2}\right) \tag{14}$$

Therefore,

$$X_i(new) = X_i(old) + D$$

Since X is a two dimensional vector.

$$\begin{aligned}(X_{i,1}, X_{i,2})(new) &= (X_{i,1}, X_{i,2})(old) + D \\ Where\ D &= R_{i,k} * r\end{aligned}$$

Where k is nbd or best individual and

$$\begin{aligned} r &= U\left(-\frac{1}{2}, \frac{1}{2}\right) \\ Also\ R_{i,k} &= (R_{i,k1}, R_{i,k2}) \\ Now\ (X_{i,1}, X_{i,2})(new) &= (X_{i,1}, X_{i,2})(old) \\ &\quad +(R_{i,k1}, R_{i,k2}) * r \\ &= (X_{i,1}(old) + R_{i,k1} * r, X_{i,2}(old) \\ &\quad + R_{i,k2} * r) \end{aligned} \tag{15}$$

Clearly, if $R_{i,best} < \frac{1}{2} Dia_{GenIndex}$ then the value of D will be a random number in the hypersphere $[-\frac{R_{i,nbd}}{2}, \frac{R_{i,nbd}}{2}]$. Obviously, initially the $Dia_{GenIndex}$ will be large and condition $R_{i,best} < \frac{1}{2} Dia_{GenIndex}$ shows that the distance between i^{th} individual and the best individual is smaller, therefore, the disruption D is considered near the i^{th} individual in the range $[-\frac{R_{i,nbd}}{2}, \frac{R_{i,nbd}}{2}]$. Now at later generations, $Dia_{GenIndex}$ will relatively be a smaller quantity and therefore the condition $R_{i,best} \geq \frac{1}{2} Dia_{GenIndex}$ will represent that the distance between i^{th} individual and the best individual is also small. In this case, we required relatively high disruption. Therefore the disruption is considered near the best individual in the range $[-\frac{R_{i,best}}{2}, \frac{R_{i,best}}{2}]$. In this way, the proposed disruption operator explores initially and as generation increases it helps to exploit the search space.

Thus, the disruption operator as defined in (7), (8), (9) and (12) is applied to biogeography-based optimization in expectation of development of a BBO with better exploration and exploitation strategy. The BBO with the disruption operator is named as DisruptBBO (DBBO). Working of DisruptBBO is given in Algorithm 2.

Algorithm 2 DisruptBBO

Initialize the population
Population size ← N_p;
Sort the population according to the increasing order of fitness
Calculate λ and μ
Generation index ← $GenIndex$;
Maximum Generations ← $MaxGen$;
for $GenIndex = 1$ to $MaxGen$ **do**
  According to the value of λ and μ, select habitat for migration
  Apply migration as in Algorithm 1
  Apply mutation as in Algorithm 1
  Sort the population according to the increasing order of fitness
  Apply disruption operator
  for $i = 1$ to N_p **do**
    if $\frac{R_{i,nbd}}{R_{i,best}} < \theta(1 - \frac{GenIndex}{MaxGen})$ **then**
      if $R_{i,best} < \frac{1}{2} Dia_{GenIndex}$ **then**
        $D = R_{i,nbd} U(-\frac{1}{2}, \frac{1}{2})$
      else
        $D = R_{i,best} U(-\frac{1}{2}, \frac{1}{2})$
      end if
      $X_i(new) = X_i(old) + D$
    else
      $X_i(new) = X_i(old)$
    end if
  end for
  Sort the population according to the increasing order of fitness
  Keep the elite solution
  Stop, if termination criteria satisfied
end for

4 Experimental results and discussion

In order to see the effect of inclusion of the disruption operator in BBO, 20 different continuous, unbiased minimization functions were selected with associated offset values as given in Table 1. This set of problems consists of problems

Table 1 Test problems (TP: Test Problem, D: Dimensions, C: Characteristic, U: Unimodal, M: Multimodal, S: Separable, NS: Non-Separable, AE: Acceptable error)

TP	Objective function	Search Range	Optimum Value	D	C	AE
Alpine	$f_1(x) = \sum_{i=1}^{D} \mid (x_i sin(x_i) \mid +0.1x_i$	[-10, 10]	f(**0**) = 0	30	M, S	1.0E-05
Axis parallel hyper ellipsoid	$f_2(x) = \sum_{i=1}^{D} ix_i^2$	[-5.12, 5.12]	f(**0**) = 0	30	U, S	1.0E-05
De jong's f_4	$f_3(x) = \sum_{i=1}^{D} ix_i^4$	[-5.12, 5.12]	f(**0**) = 0	30	U, S	1.0E-05
Ellipsoidal	$f_4(x) = \sum_{i=1}^{D} (x_i - i)^2$	[-30, 30]	f(1, 2, 3,..., D) = 0	30	U, S	1.0E-05
Griewank	$f_5(x) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos(\frac{x_i}{\sqrt{i}}) + 1$	[-600, 600]	f(**0**) = 0	30	M, NS	1.0E-05
Rosenbrock	$f_6(x) = \sum_{i=1}^{D-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$	[-2.048, 2.048]	f(**1**) = 0	30	U, NS	1.0E-02
Salomon prob 3	$f_7(x) = 1 - \cos\left(2\pi\sqrt{\sum_{i=1}^{D} x_i^2}\right) + 0.1\sqrt{\sum_{i=1}^{D} x_i^2}$	[-100, 100]	f(**0**) = 0	30	M, S	1.0E-01
Schwefel	$f_8(x) = -\sum_{i=1}^{D} x_i sin(\mid x_i \mid^{1/2})$	[-512, 512]	$f(\pm[\pi(0.5+k)]^2)$ = -418.9829 * D	30	M, S	1.0E-05
Schwefel221	$f_9(x) = \max_i \mid x_i \mid, 1 \leq i \leq D$	[-100, 100]	f(**0**) = 0	30	U, S	1.0E-05
Schwefel222	$f_{10}(x) = \sum_{i=1}^{D} \mid x_i \mid + \prod_{i=1}^{D} \mid x_i \mid$	[-10, 10]	f(**0**) = 0	30	U, NS	1.0E-05
Sphere	$f_{11}(x) = \sum_{i=1}^{D} x_i^2$	[-5.12, 5.12]	f(**0**) = 0	30	U, S	1.0E-05
Pathological	$f_{12}(x) = \sum_{i=1}^{D-1} 0.5 + \frac{sin^2\sqrt{(100x_i^2+x_{i+1}^2)}-0.5}{1+0.001(x_i^2-2x_ix_{i+1}+x_{i+1}^2)^2}$	[-100, 100]	f(**0**) = 0	30	M, NS	1.0E-05
Michalewicz	$f_{13}(x) = -\sum_{i=1}^{D} sin(x_i)\left[\frac{sini(x_i)^2}{\pi}\right]^{20}$	[0,π]	f_{min} = -9.66015	10	M, S	1.0E-05
Zakharov's	$f_{14}(x) = \sum_{i=1}^{D} x_i^2 + \left(\sum_{i=1}^{D} \frac{i}{2}x_i\right)^2 + \left(\sum_{i=1}^{D} \frac{i}{2}x_i\right)^4$	[-5.12, 5.12]	f(**0**) = 0	30	M, NS	1.0E-02
Neumaier3	$f_{15}(x) = \sum_{i=1}^{D} (x_i - 1)^2 - \sum_{i=2}^{D} x_i x_{i-1}$	[-100, 100]	$f_{min} = -\frac{(D*(D+4)*(D-1))}{6}$	10	U, NS	1.0E-01
Brown 3	$f_{16}(x) = \sum_{i=1}^{D-1} \left[(x_i^2)^{(x_{i+1}^2+1)} + (x_{i+1}^2)^{(x_i^2+1)}\right]$	[-1, 4]	f(**0**) = 0	30	U, NS	1.0E-05
Beale	$f_{17}(x) = [1.5 - x_1(1-x_2)]^2 + [2.25 - x_1(1-x_2{}^2)]^2 + [2.625 - x_1(1-x_2{}^3)]^2$	[-4.5, 4.5]	f(3, 0.5) = 0	2	U, NS	1.0E-05
Easom	$f_{18}(x) = -cosx_1cosx_2e^{((-(x_1-\pi)^2-(x_2-\pi)^2))}$	[-100, 100]	$f(\pi, \pi)$ = -1	2	U, S	1.0E-13
Ackley	$f_{19}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)) + 20 + e$	[-30, 30]	f(**0**) = 0	30	M, NS	1.0E-05
Rastrigin	$f_{20}(x) = \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i)] + 10D$	[-5.12, 5.12]	f(**0**) = 0	30	M, S	1.0E-05

which are of varying degree of complexities, some are multimodal and some, separable and non-separable.

4.1 Experimental setting

The following experimental setting was adopted to observe the performance of the proposed DisruptBBO metaheuristic.

- Population size N_p = 50
- Mutation probability = 0.01
- Elitism size = 2
- Maximum immigration rate = 1
- Maximum emigration rate = 1
- Maximum number of generations/iterations = 1000
- Total number of runs/simulations = 100

  The value of θ has not been adopted from [21] and [13] but determined based an experimental analysis, so that an optimal value of θ can be obtained. The performance of DBBO for the constant value of θ such as 0.1, 0.2,..., 1 is tested on test problems f_3, f_4, f_5, f_6 and f_{19}. As can be seen in Fig. 3, the performance of DBBO with different values of θ between 0.1 and 1 is indistinguishable. Therefore, the linearly increasing value of θ from 0.1 to 1 (Inc.(0.1-1)) and linearly decreasing value of θ from 1 to 0.1 (Dec.(1-0.1)) as defined in (16) and (17), respectively are considered for experiments.

$$\theta = 0.1 + \frac{(1-0.1)}{MaxGen} * GenIndex \tag{16}$$

$$\theta = 1 - \frac{(1-0.1)}{MaxGen} * GenIndex \tag{17}$$

Similar experiments were also performed for linearly decreasing values of θ from 100 to 0.1 (Dec.(100-0.1)), linearly increasing values of θ from 0.1 to 100 (Inc.(0.1-100)), constant values of θ, $\theta = 0.1$, $\theta = 1$, $\theta = 10$, $\theta = 50$, and $\theta = 100$. Figure 4 shows the mean error over 100 runs. It is clear that the performance of DBBO is very sensitive to θ, and the optimal choice of θ is its linearly decreasing value from 1 to 0.1.

Therefore, for all further experiments, θ is considered to be linearly decreasing from 1 to 0.1 in DBBO. Now with linearly decreasing value of θ given in (17), the value of C changes as given in (18):

$$C = 1 - \frac{1.9 * GenIndex}{MaxGen} + \frac{0.9 * GenIndex^2}{MaxGen^2} \tag{18}$$

The effect of the modified choice of θ over C can be observed in Fig. 5. Figure 5a depicts the variation in C based on the choice of θ as given in ([13, 21]) while Fig. 5b shows the same in the case of the new choice of θ. Obviously, now C is a nonlinear decreasing function of generations as compared to linearly decreasing function of generations ([13, 21]).

Parameter settings for the algorithms: Basic BBO (BBO), Linearized BBO (LBBO) [26] and Blended BBO (B-BBO) [15] are same as given above.

4.2 Results and discussion

Numerical results using experimental settings of Section 4.1 are given in Table 2. In Table 2, minimum error ($MinE$), standard deviation (SD), mean error (ME), the mean number of generations (MG) and success rate (SR) are reported. The numerical results are primarily compared with those of basic BBO, LBBO, and B-BBO. The performances of these algorithms has been compared based on SR, MG, ME and SD criteria. First, the algorithms are compared based on SR. If the SR is not significantly different then the

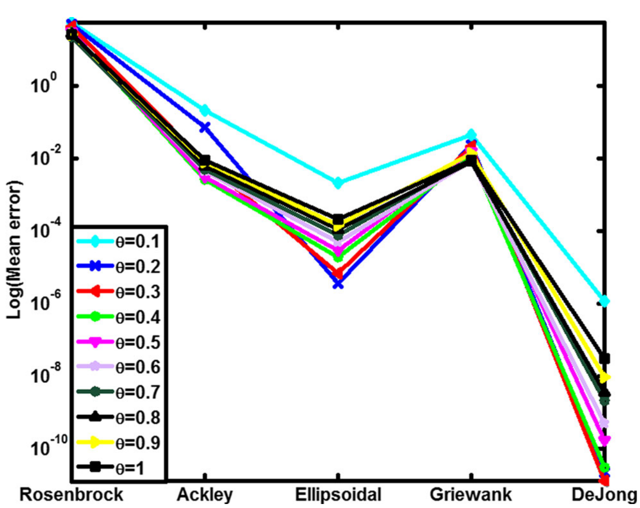


Fig. 3 Performance of DBBO with various values of θ between 0.1 to 1

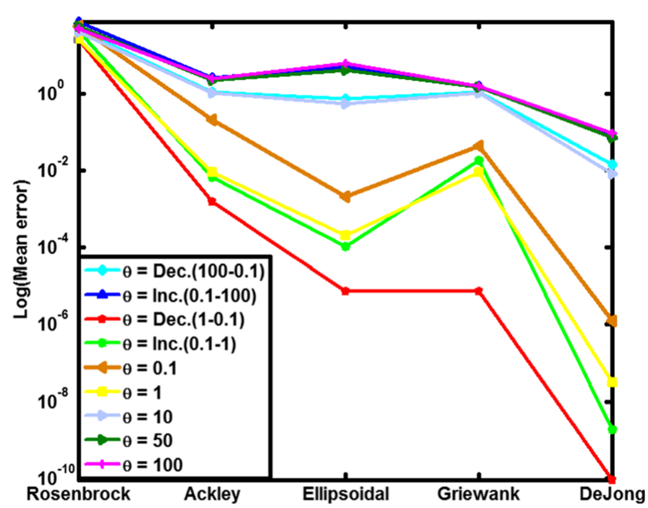
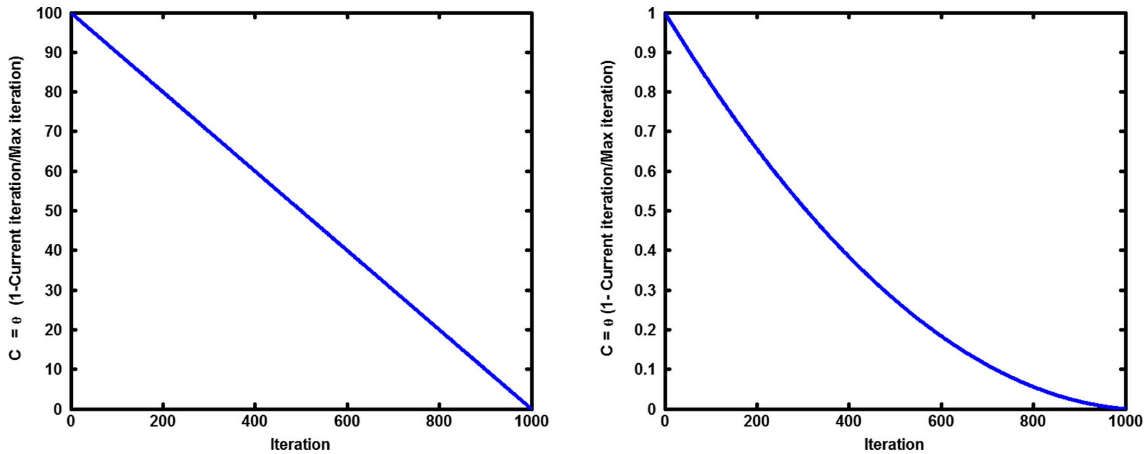


Fig. 4 Performance of DBBO with various choices of θ



(a) Variation of C according to θ given in [21], [13]

(b) Variation of C according to θ given in Eq. (17)

Fig. 5 Variation of constant C as a function of θ with respect to iterations

comparison is based on the MG. Similarly, if the MG for the algorithms are not significantly different, algorithms are compared based on the ME and finally based on the SD.

From Table 2, it can be observed that the performance of DBBO is better on 13 functions (f_1, f_2, f_4, f_5, f_6, f_8, f_{10}, f_{11}, f_{12}, f_{15}, f_{16}, f_{19}, f_{20}) from amongst all considered algorithms. These functions are high dimensional and include unimodal, multimodal, separable, non-separable having a solution at the origin as well as at points other than origin. It can be concluded that DBBO has the capability to balance exploration and exploitation efficiently for the high dimensional problems. However LBBO outperforms DBBO over 6 functions (f_3, f_7, f_9, f_{13}, f_{17}, f_{18}). These functions include unimodal, multimodal, separable and non-separable. f_3, f_7, f_9 and f_{13} have the solutions at the origin. According to this analysis, it is clear that most of the time LBBO gives better results for those problems which have optimal solution at the origin. B-BBO outperforms DBBO over one function f_{14}, which is multimodal and non-separable. From these comparisons it is clear that most of the time DBBO outperforms in the sense of reliability, efficiency, and accuracy as compared to BBO, B-BBO and is competitive with LBBO.

Some more intensive statistical analyses based on Mann-Whitney U rank sum test [5], Boxplot and Performance Index [4] and diversity analysis [23] are also being presented with the numerical results of BBO, LBBO, B-BBO and DBBO.

4.2.1 Statistical analysis

The empirical distribution of data is efficiently represented graphically by the boxplot analysis tool [5]. The boxplots for SR, MG, and ME for all algorithms BBO, LBBO, B-BBO, and DBBO have been depicted in Fig. 6. Figure 6a, b and c show the boxplot regarding success rate, mean generation index and mean error, respectively. Figure 6b and c, clearly show that DBBO performs better as compared to the others from reliability, efficiency, and accuracy points of view.

From the boxplots of Fig. 6, it is clear that the outcome of SR, MG and ME are not normally distributed. Therefore, a non-parametric test Mann-Whitney U rank sum [5] is applied to see the difference among the performance of different algorithms. In the present study this test was performed on mean error at 5 % level of significance (a = 0.05) between DBBO - BBO, DBBO - LBBO and DBBO - B-BBO. Table 3 shows the results of Mann-Whitney U rank sum test for mean error of 100 simulations. In Table 3, '+' sign appears if DBBO performs better than considered algorithms, '-' sign appears, if DBBO performs worse than considered algorithms and '=' sign appears when on the compared algorithms have no significant difference. Table 3 includes 48 '+' signs out of 60 comparisons. Therefore, we conclude that DBBO algorithm is significantly better than BBO, LBBO and B-BBO over considered test problems.

In order to analyze the diversity variation during the search in the considered algorithms we use the diversity measure as defined in [20]. The diversity measure of D-dimensional problem is defined as

$$diversity(D_A) = \frac{1}{L \times N_p} \sum_{i=1}^{N_p} \left(\sqrt{\sum_{j=1}^{D} (x_i^j - \bar{x}^j)^2} \right) \quad (19)$$

Where L is the length of longest diagonal in the search space, D is the dimensionality of the problem, x_i^j is the j^{th} dimension of the i^{th} individual and $\bar{x}^j$ is the j^{th} value of the average point $(\bar{x})$ at any generation. The average point

Table 2 Comparison of DBBO with some BBO variants

Test problem	Algorithm	MinE	SD	ME	MG	SR
f_1	BBO	2.33E−02	7.43E−03	3.42E−02	1000.00	0
	LBBO	1.36E−03	1.23E−03	3.63E−03	1000.00	0
	B-BBO	7.68E−03	1.85E−02	2.31E−02	1000.00	0
	DBBO	1.37E−04	1.01E−03	1.45E−03	1000.00	0
f_2	BBO	4.11E−02	6.21E−02	1.50E−01	1000.00	0
	LBBO	6.93E−05	1.38E−04	3.00E−04	1000.00	0
	B-BBO	3.72E−03	4.85E−03	1.12E−02	1000.00	0
	DBBO	9.36E−06	9.90E−05	8.71E−05	997.73	7
f_3	BBO	1.11E−05	2.34E−04	3.04E−04	1000.00	0
	LBBO	3.48E−06	9.66E−07	9.17E−06	317.77	100
	B-BBO	2.67E−06	7.53E−06	1.11E−05	901.51	78
	DBBO	6.41E−06	7.32E−07	9.13E−06	590.63	100
f_4	BBO	1.29E−01	1.36E−01	3.67E−01	1000.00	0
	LBBO	3.77E−02	2.16E−01	2.36E−01	1000.00	0
	B-BBO	2.41E+00	3.90E+00	8.42E+00	1000.00	0
	DBBO	8.49E−06	3.56E−06	1.07E−05	974.76	79
f_5	BBO	9.57E−01	2.03E-02	1.03E+00	1000.00	0
	LBBO	6.65E−03	2.24E-02	3.95E−02	1000.00	0
	B-BBO	1.60E−01	1.32E-01	4.24E−01	1000.00	0
	DBBO	7.66E−06	9.24E-03	7.63E−03	990.51	18
f_6	BBO	1.34E+01	2.96E+01	5.84E+01	1000.00	0
	LBBO	2.51E+01	3.70E-01	2.72E+01	1000.00	0
	BBBO	2.71E+01	2.17E-01	2.80E+01	1000.00	0
	DBBO	1.60E−02	2.70E+01	2.43E+01	1000.00	0
f_7	BBO	8.00E−01	2.22E-01	1.31E+00	1000.00	0
	LBBO	3.00E−01	7.20E-02	4.69E−01	1000.00	0
	B-BBO	4.00E−01	8.36E-02	6.22E−01	1000.00	0
	DBBO	3.00E−01	9.24E-02	4.97E−01	1000.00	0
f_8	BBO	3.78E+00	3.38E+00	9.46E+00	1000.00	0
	LBBO	9.81E+00	9.61E+00	2.77E+01	1000.00	0
	B-BBO	1.04E+02	1.19E+02	3.22E+02	1000.00	0
	DBBO	3.84E−04	1.43E-03	9.85E−04	1000.00	0
f_9	BBO	2.26E+00	1.11E+00	5.17E+00	1000.00	0
	LBBO	1.85E−01	5.76E-02	3.00E−01	1000.00	0
	B-BBO	7.63E−01	2.26E-01	1.35E+00	1000.00	0
	DBBO	4.50E−01	4.76E−01	1.33E+00	1000.00	0
f_{10}	BBO	4.21E−01	1.17E−01	6.82E−01	1000.00	0
	LBBO	1.09E−02	5.11E−03	2.07E−02	1000.00	0
	B-BBO	8.11E−02	3.75E−02	1.63E−01	1000.00	0
	DBBO	3.41E−04	1.73E−03	2.29E−03	1000.00	0
f_{11}	BBO	3.40E−03	3.95E−03	1.09E−02	1000.00	0
	LBBO	9.42E−06	1.06E−05	2.31E−05	993.74	8
	B-BBO	3.25E−04	4.52E−04	1.01E−03	1000.00	0
	DBBO	7.80E−06	6.65E−06	1.24E−05	887.54	75
f_{12}	BBO	2.53E+00	5.05E−01	3.50E+00	1000.00	0

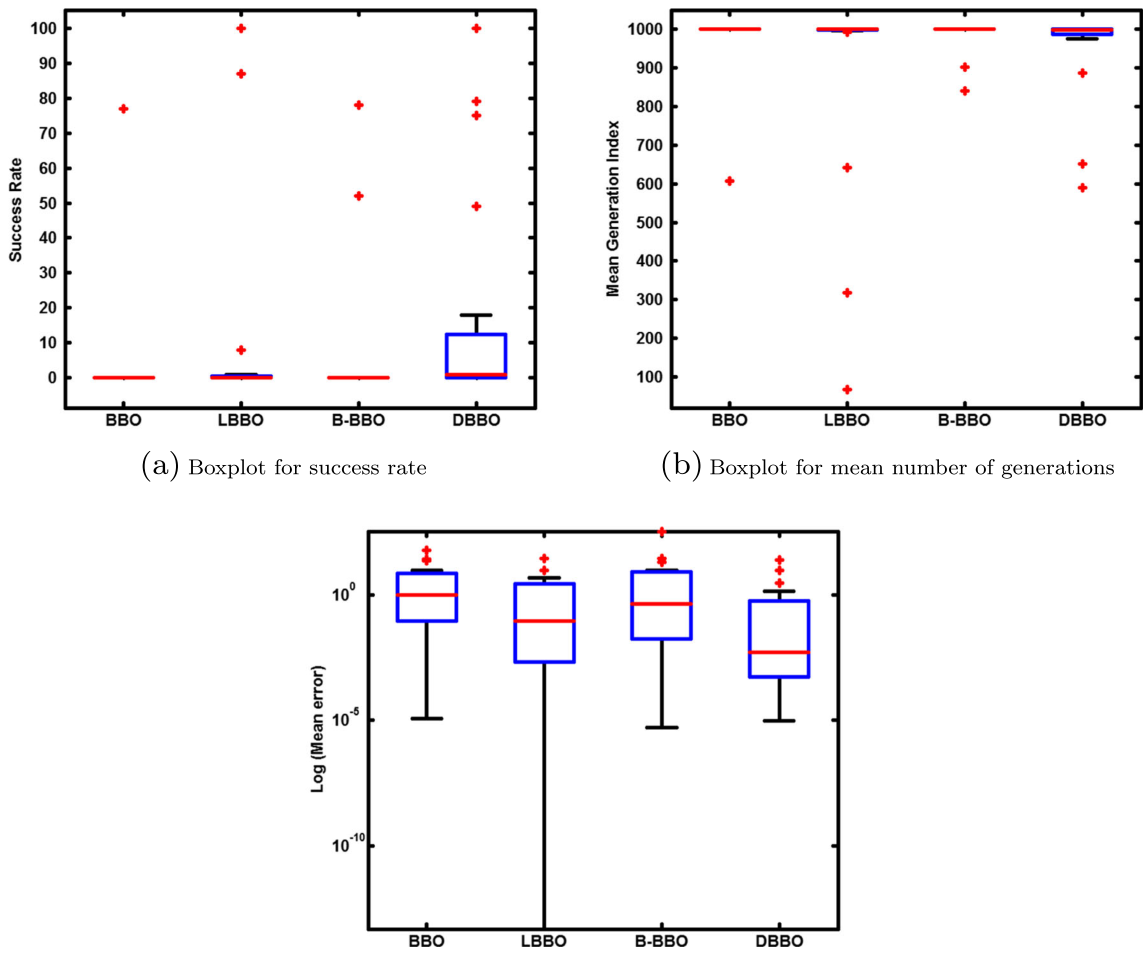
Table 2 (continued)

Test problem	Algorithm	MinE	SD	ME	MG	SR
	LBBO	2.64E+00	6.11E−01	4.36E+00	1000.00	0
	B-BBO	3.09E+00	6.86E−01	4.76E+00	1000.00	0
	DBBO	1.25E−03	3.19E−01	1.57E−01	1000.00	0
f_{13}	BBO	9.66E+00	1.43E−12	9.66E+00	1000.00	0
	LBBO	9.66E+00	6.36E−13	9.66E+00	1000.00	0
	B-BBO	9.66E+00	4.30E−12	9.66E+00	1000.00	0
	DBBO	9.66E+00	2.27E−12	9.66E+00	1000.00	0
f_{14}	BBO	9.09E+00	1.15E+01	2.57E+01	1000.00	0
	LBBO	2.00E−01	7.09E−01	1.11E+00	1000.00	0
	B-BBO	1.95E−01	1.79E−01	4.67E−01	1000.00	0
	DBBO	6.44E−01	1.83E+00	3.00E+00	1000.00	0
f_{15}	BBO	1.01E−01	2.06E+01	2.34E+01	1000.00	0
	LBBO	9.99E−02	2.97E+00	4.68E+00	997.76	1
	B-BBO	4.37E+00	8.15E+00	2.04E+01	1000.00	0
	DBBO	3.22E−02	1.06E+00	6.11E−01	651.32	49
f_{16}	BBO	1.56E−03	1.88E−03	5.04E−03	1000.00	0
	LBBO	1.53E−04	2.09E−04	4.88E−04	1000.00	0
	B-BBO	2.74E−02	2.92E−02	8.08E−02	1000.00	0
	DBBO	9.52E−06	4.07E−05	4.77E−05	995.84	6
f_{17}	BBO	2.57E−07	2.12E−05	1.19E−05	607.84	77
	LBBO	8.09E−07	7.62E−02	7.63E−03	641.44	87
	B-BBO	1.04E−07	7.86E−02	7.89E−03	840.02	52
	DBBO	1.22E−06	3.98E−03	2.54E−03	985.04	2
f_{18}	BBO	9.00E−06	4.90E−01	6.10E−01	1000.00	0
	LBBO	1.11E−15	2.97E−14	4.73E−14	66.62	100
	B-BBO	2.01E−08	7.36E−06	5.02E−06	1000.00	0
	DBBO	3.40E−14	1.71E−01	3.10E−02	992.94	2
f_{19}	BBO	4.81E−01	1.91E−01	8.63E−01	1000.00	0
	LBBO	1.03E−02	5.14E−03	2.03E−02	1000.00	0
	B-BBO	1.08E−01	3.24E−01	3.68E−01	1000.00	0
	DBBO	3.16E−04	5.89E−04	1.73E−03	1000.00	0
f_{20}	BBO	6.35E−01	5.27E−01	1.49E+00	1000.00	0
	LBBO	6.28E−03	3.01−01	1.44E−01	1000.00	0
	B-BBO	2.39E+00	2.49E+00	7.54E+00	1000.00	0
	DBBO	8.66E−06	3.01E−01	1.24E−01	996.18	7

is calculated as the mean of the position of all individuals as given in (20).

$$\bar{X} = (\bar{x}^1, \bar{x}^2, ..., \bar{x}^D) = \left(\frac{1}{N_p} \sum_{i=1}^{N_p} x_i^1, \frac{1}{N_P} \sum_{i=1}^{N_p} x_i^2, ..., \frac{1}{N_p} \sum_{i=1}^{N_p} x_i^D \right) \tag{20}$$

The diversity measure is dependent on the population size, problem dimension, and search ranges. Low population diversity shows that population has clusters in small regions. Conversely, high population diversity shows that the population is scattered in a wide region. Low population diversity measure is responsible for local convergence. However, high diversity measure is responsible for poor convergence. In this test the same parameter setting is considered as given in Section 4.1 except the total number of simulations which is set to be 1. In order to determine the diversity measurement for all considered algorithms in single run for selected function



(a) Boxplot for success rate

(b) Boxplot for mean number of generations

(c) Boxplot for mean error

Fig. 6 Comparison based on boxplots

(f_4, f_5, f_{11}, f_{20}) are shown in Figs. 7, 8, 9, and 10. The selected functions include unidimensional, multidimensional, separable and non-separable. Here function f_4 has the optimal solution far from the origin and functions f_5, f_{11} and f_{20} have optimal solutions at the origin. In order to clearly visualize the impact of diversity measure for all selected functions, we plotted the graphs in four ranges of iterations. In the first range, diversity measure from initial iteration to the iterations where the diversity of proposed DBBO algorithm is comparatively larger than other considered algorithms. In the second range, diversity measure of the proposed DBBO algorithm is comparatively lower than other considered algorithms. In the third range, diversity measure in last 200 iterations from (800-1000) is considered. In the fourth range, diversity measure in all iterations from (0-1000) is considered. Figures 7a, 8a, 9a and 10a represent diversity measure in the first range, Figs. 7b, 8b, 9b and 10b represent diversity measure in the second range, Figs. 7c, 8c, 9c and 10c represent diversity measure in the third range and finally Figs. 7d, 8d, 9d and 10d represent diversity measure in the fourth range of iterations. From the analysis of the diversity measure for four selected functions, it can be said that the diversity of the proposed DBBO algorithm is comparatively higher in the initial iterations and lower in later iterations than other considered algorithms. In order to see the compared effect of diversity in the proposed DBBO algorithm, we tested two algorithms DBBO and BBO independently 30 times on three functions (f_5, f_{11}, f_{20}). Figure 11 shows the diversity measure between the BBO and DBBO algorithms with an average over 30 runs on three functions. Figure 11a, d and g, show the diversity measure of three functions in 0 to 1000 iteration. In initial iterations, it is difficult to distinguish the diversity measure between DBBO and BBO algorithms. Thus, the diversity measure of three functions (f_5, f_{11}, f_{20}) in initial stage (0-50 iteration) is shown in Fig. 11b, e and h, respectively. Here we observe that in the initial stage there is a slight difference in the diversity measure between DBBO and BBO algorithms and the diversity measure of the DBBO algorithm

Table 3 Comparison of DBBO and other BBO variants based on Mann-Whitney U rank sum test

Test problem	DBBO Vs BBO	DBBO Vs LBBO	DBBO Vs B-BBO
f_1	+	+	+
f_2	+	+	+
f_3	+	=	+
f_4	+	+	+
f_5	+	+	+
f_6	+	+	+
f_7	+	−	−
f_8	+	+	+
f_9	+	−	=
f_{10}	+	+	+
f_{11}	+	+	+
f_{12}	+	+	+
f_{13}	=	−	+
f_{14}	+	−	−
f_{15}	+	+	+
f_{16}	+	+	+
f_{17}	−	+	+
f_{18}	+	−	−
f_{19}	+	+	+
f_{20}	+	+	+
Total number of + sign	18	14	16

decreases with iterations as compare to BBO algorithm. Therefore, from the analysis of diversity measure, it is concluded that fast convergence is performed by the DBBO algorithm at later iterations as compared to BBO algorithm.

Further, to see the performance of the considered algorithm by giving the weighted importance to success rate, mean generation index and mean error, performance indices (PIs) were evaluated [23]. The value of the PI(performance index) for BBO, LBBO, B-BBO and DBBO are evaluated using

$$PI = \frac{1}{N}\sum_{i=1}^{N}(k_1\alpha_1^i + k_2\alpha_2^i + k_3\alpha_3^i) \tag{21}$$

$$Where\ \alpha_1^i = \frac{Sr^i}{Tr^i};\ \alpha_2^i = \begin{cases} \frac{MinG^i}{MG^i}, & if\, Sr^i > 0 \\ 0, & if\, Sr^i = 0 \end{cases}\ and\ \alpha_3^i = \frac{Mo^i}{Ao^i}$$

$i = 1, 2, ..., N$

- Sr^i = Successful simulations/runs of the i^{th} problem.
- Tr^i = Total simulation of the i^{th} problem.
- $MinG^i$ = Minimum of mean generation index used for obtaining the required solution of the i^{th} problem.
- MG^i = Mean generation index used for obtaining the required solution of the i^{th} problem.
- Mo^i = Minimum of mean error obtained for the i^{th} problem.
- Ao^i = Mean error obtained by an algorithm for the i^{th} problem.
- N = Total number of optimization problems evaluated.

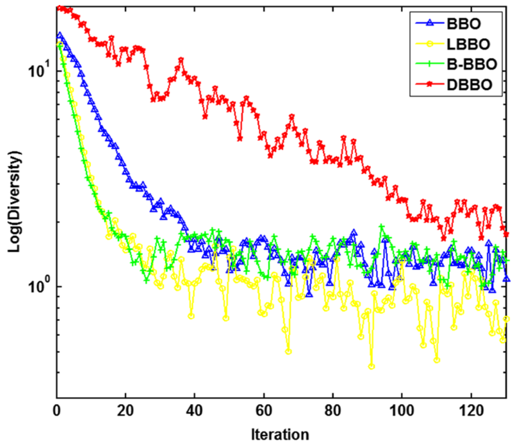
The corresponding weight assigned to the success rate, mean generation index and mean error are represented by k_1, k_2, and k_3, respectively where $k_1 + k_2 + k_3 = 1$ and $0 \le k_1, k_2, k_3 \le 1$. For calculating PIs, equal weights are assigned to two variables while the weight of remaining variable vary from 0 to 1 as given in [6]. The resultant cases are:

1. $k_1 = W, k_2 = k_3 = \frac{1-W}{2}, 0 \le W \le 1;$
2. $k_2 = W, k_1 = k_3 = \frac{1-W}{2}, 0 \le W \le 1;$
3. $k_3 = W, k_1 = k_2 = \frac{1-W}{2}, 0 \le W \le 1;$

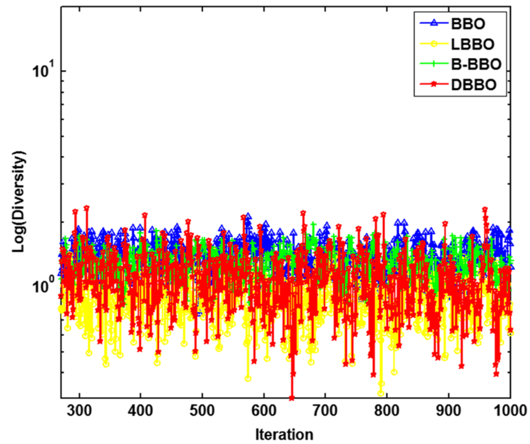
The graph corresponding to each case (1), (2) and (3) for the considered algorithms BBO, LBBO, B-BBO, and DBBO are depicted in Fig. 12. In these figures weights k_1, k_2, and k_3 are represented along the X-axis while the PI is represented along the Y-axis.

In case (1), the mean generation index and mean error have assigned equal weights, in case (2), success rate and mean error have assigned equal weights and in case (3) the success rate and mean generation index have assigned equal weights. In all cases, the PI of DBBO algorithm is higher compared to these of the considered algorithms. Thus, DBBO performs better than BBO, LBBO and B-BBO over the considered criteria. From above statistical analyses, it is concluded that DBBO is a better performer when we consider some popular variants of BBO.

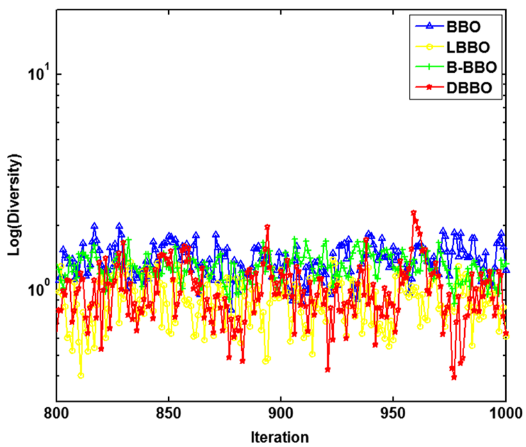
In order to justify the validity of proposed DBBO, it is also compared with some most recent BBO variants and other state-of-the-art algorithms. BBO with modified blended crossover and polynomial mutation (BBO-MBLX-PM) [3] and backtracking BBO (BBBO) [10] are selected as the most the recent BBO variants for comparison. In Table 4, results of BBO-MBLX-PM are presented in column 3. These have been taken from the original paper [3]. For a fair comparison, DBBO parameters are the set same as those of BBO-MBLX-PM. Table 4 presents the comparison between the BBO-MBLX-PM algorithm and the DBBO algorithm on 20 test problems. In Table 4, average fitness (MeanF) over 30 runs is reported for each test problem. The best performances in Table 4 are marked in bold font. Results show that performance of the DBBO algorithm is better than the BBO-MBLX-PM algorithm. Similarly, in Table 5, the DBBO algorithm is compared with the BBBO algorithm. Here the BBBO results are reported from [10], and DBBO



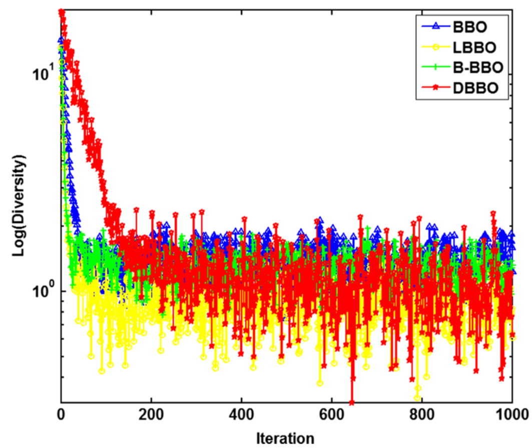
(a) Diversity measure of sphere function from 1 – 130 iterations



(b) Diversity measure of sphere function from 270 – 1000 iterations



(c) Diversity measure of sphere function from 800 – 1000 iterations

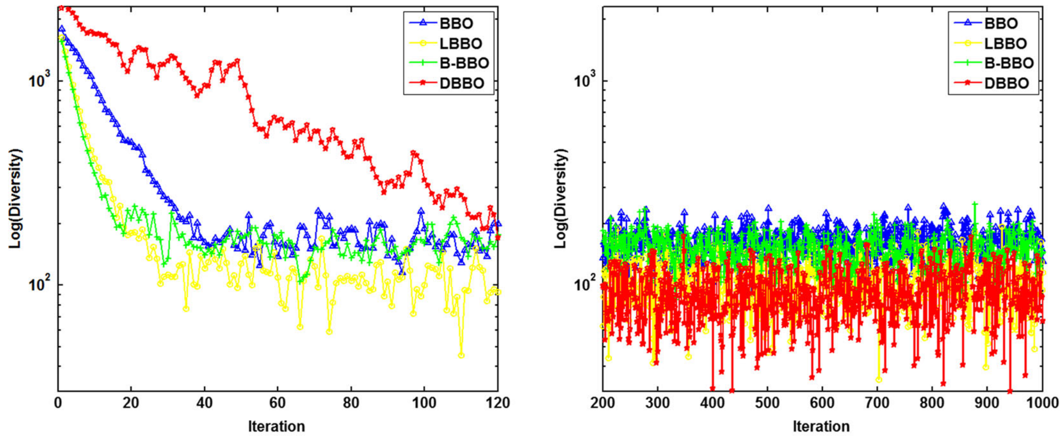


(d) Diversity measure of sphere function from 1 – 1000 iterations

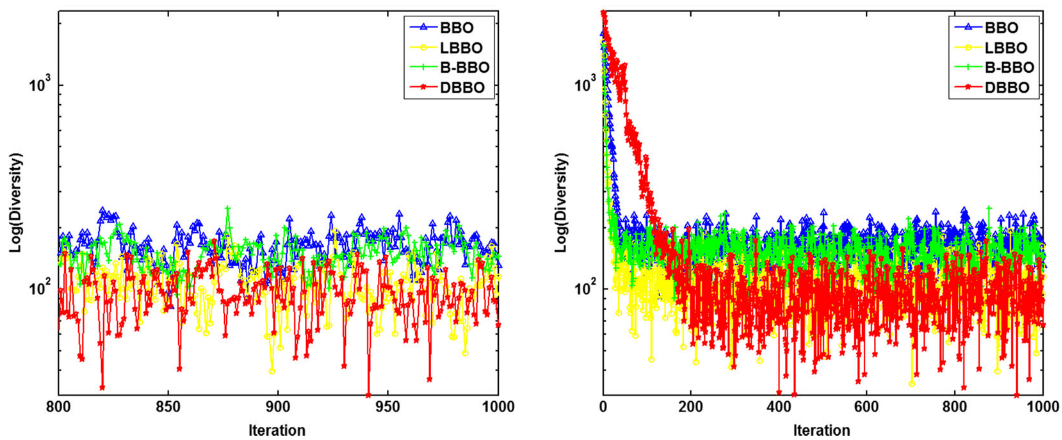
Fig. 7 Diversity comparison for sphere function in a single run

parameters are also set same as in BBBO. In Table 5, minimum error (BestE), average error (MeanE) are reported for each test problem. The best performances in Table 5 are marked in bold font. The performance of the DBBO algorithm is better than the BBBO algorithm based on average error comparison, but the BBBO algorithm is better based on minimum error. Finally, it is concluded that DBBO is a competitive algorithm to the most recent variants of BBO.

The comparison has extended to state-of-the-art algorithms, second best particle (SP-PSO) [24], particle swarm optimization with second global best particle (SG-PSO) [24], lévy distributed differential evolution (LdDE) [12] and hybrid artificial bee colony optimizer (HABC) [16] algorithms. Again, for all these comparisons the parameters of DBBO and compared algorithms' were set to be same. Results of the compared algorithms have been reproduced from their original articles. In Tables 6, 7, 8 and 9, best performances are marked by bold font. Table 6 presents the comparison among SP-PSO, SG-PSO and DBBO algorithms. In Table 6, the average function value (MeanF) is reported for each test problem. It is clearly shown that the performance of DBBO is better than SG-PSO and SP-PSO. In Tables 7 and 8, LdDE and DBBO algorithms are considered for comparison. Comparison results in Tables 7 and 8 are for 10-dimensional and 30-dimensional test problems, respectively. The reported results in Tables 7 and 8 are the best error value (BestE), mean best error value (MeanE) and standard deviation (Std). Results in Tables 7 and 8 show that DBBO outperforms the LdDE algorithm both for 10-dimensional and 30-dimensional test problems. Table 9 reports the comparison between the hybrid artificial bee colony optimizer (HABC) algorithm and the DBBO algorithm. Here the average fitness value (MeanF) and standard deviation (Std) are reported. Results clearly show that the DBBO algorithm outperforms the HABC algorithm. The overall comparisons thus



(a) Diversity measure of griewank function from $1 - 120$ iterations
(b) Diversity measure of griewank function from $200 - 1000$ iterations



(c) Diversity measure of griewank function from $800 - 1000$ iterations
(d) Diversity measure of griewank function from $1 - 1000$ iterations

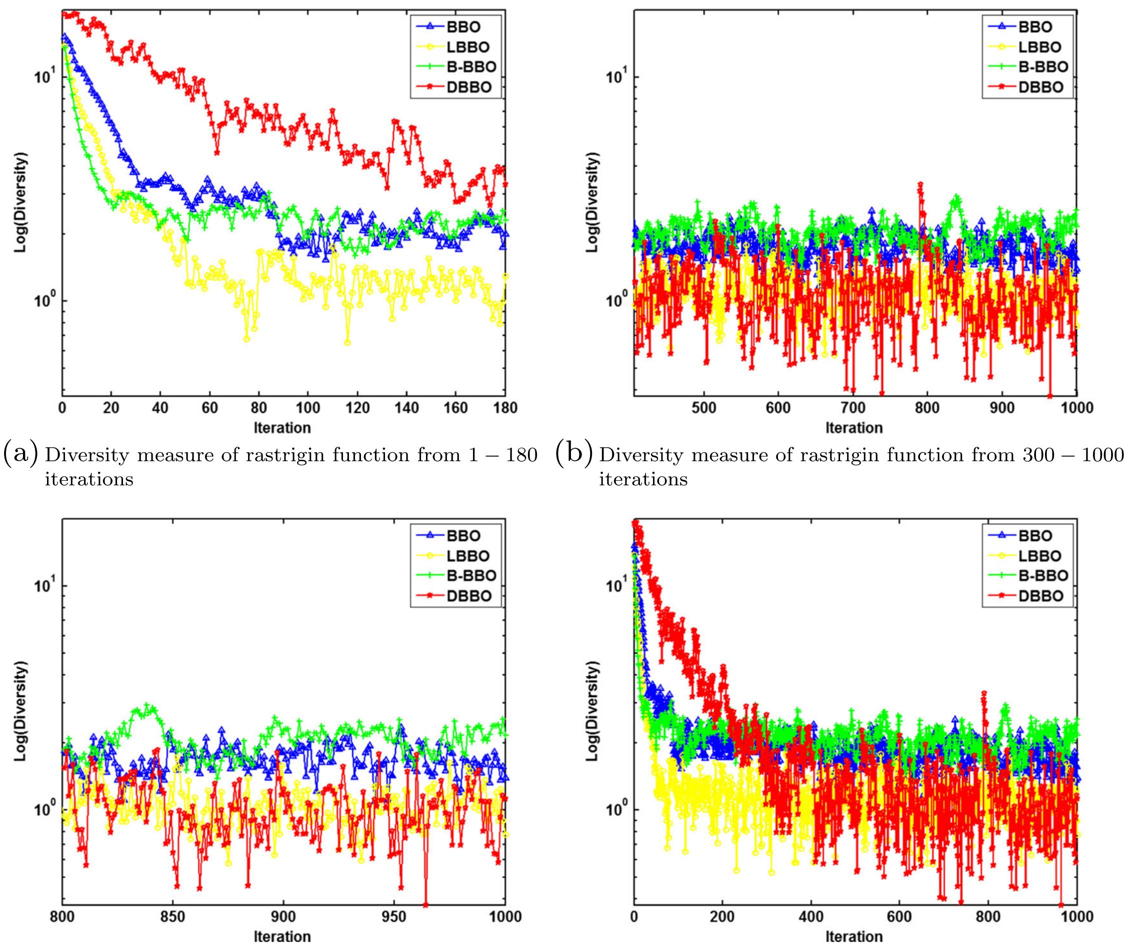
Fig. 8 Diversity comparison for griewank function in a single run

show that the proposed DBBO algorithm is competing with the current state-of-the-art metaheuristics and is highly recommended for multi-dimensional complex optimization problems.

5 Application of DBBO to optimal power flow problem

5.1 Optimal power flow problem

Optimal power flow (OPF) is an important tool for power system operators that aims to optimize a certain objective, subject to the network power flow equations of the system, and equipment within operating limits. The minimization of fuel cost is widely considered an objective function subject to equality and inequality constraints among many objectives. Equality constraints in the form of power flow equations and inequality constraints in the form of limits on control variables and operational limits on system dependent variables. The control variables for the OPF problem are generator real powers, generator bus voltages, the reactive powers for shunt VAR compensations, and transformer tap settings. The system dependent variables are the load bus voltages, transmission line loadings, and generators reactive powers. In general, the OPF problem is nonlinear and non-convex optimization problem with a large number of constraints. The OPF optimization problem is generally solved by the conventional method and intelligent method. Disrupt biogeography-based optimization (DBBO) algorithm is one of the intelligent methods for solving OPF optimization problem. The main purpose of solving optimal power flow optimization problem is to find an optimal setting of control variable while satisfying the equality and inequality constraints.



(a) Diversity measure of rastrigin function from 1 − 180 iterations

(b) Diversity measure of rastrigin function from 300 − 1000 iterations

(c) Diversity measure of rastrigin function from 800 − 1000 iterations

(d) Diversity measure of rastrigin function from 1 − 1000 iterations

Fig. 9 Diversity comparison for rastrigin function in a single run

5.2 Mathematical formulation of OPF formulation

Mathematical formulation of a general optimal power flow problem can be seen as:

$$\text{Min } F(u, x) \tag{22}$$

subject to:

$$G(u, x) = 0 \tag{23}$$

$$H^{min} \leq H(u, x) \leq H^{max} \tag{24}$$

Where F is the objective function to be optimized, G is set of equality constraints representing nodal power injections, and H is set of inequality constraints. The vector u consists of independent variables or control variables which include generator bus voltages (V_g), generator real powers (P_g) except at slack bus, transformer tap settings (T), shunt VAR compensation (Q_c). Vector x consists of dependent variables or state variable which includes generator active power at slack bus (P_{g1}), load bus voltages (V_l), generator reactive powers (Q_g) and transmission line loading (line flow) (S_l). Hence, u and x can be expressed as:

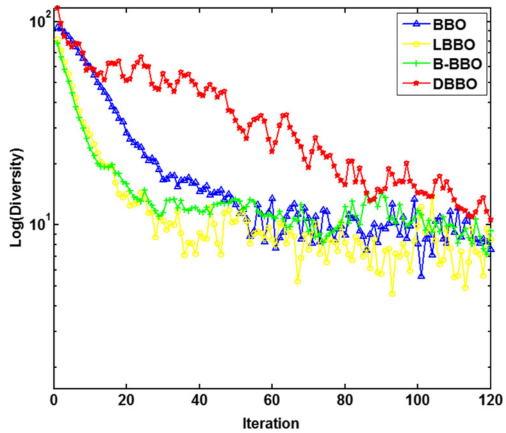
$$u = [P_{g2} \ldots . P_{gng}, V_{g1} \ldots . V_{gng}, Q_{c1} \ldots . Q_{cnc}, T_1 \ldots . T_{nt}] \tag{25}$$

$$x = [P_{g1}, V_{l1} \ldots . V_{lnl}, Q_{g1} \ldots . Q_{gng}, S_{l1} \ldots . S_{lNl}] \tag{26}$$

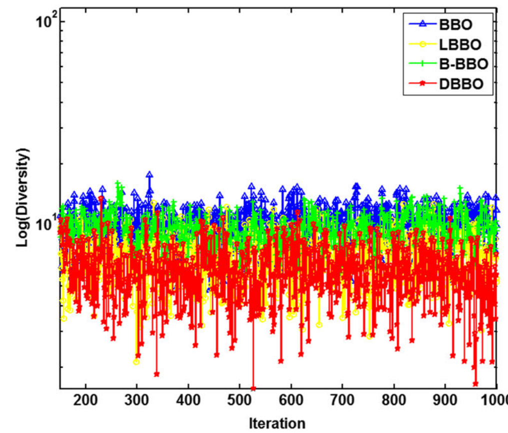
where ng, nc, nt, nl and Nl are the numbers of generators, shunt VAR compensators, regulating transformers, load buses and transmission lines, respectively.

$G(u, x)$ is the set of equality constraints which represents typical load flow equations of the type:

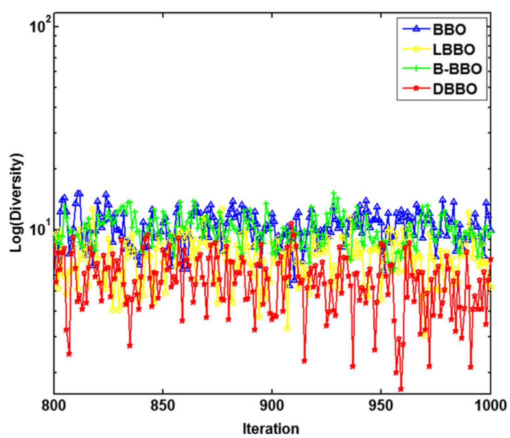
$$P_{gi} - P_{di} - V_i \sum_{k=1}^{nb} V_k (G_{ik} \cos \theta_{ik} + B_{ik} \sin \theta_{ik}) = 0 \tag{27}$$



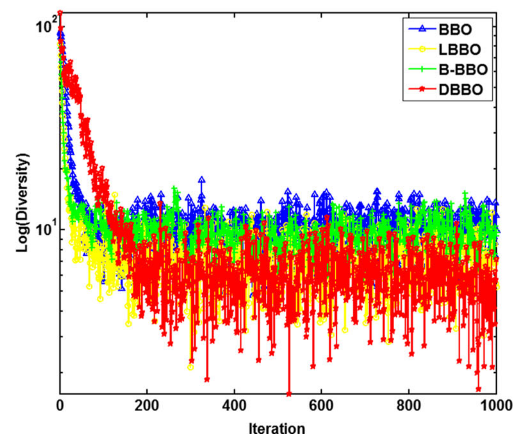
(a) Diversity measure of ellipsoidal function from $1-120$ iterations



(b) Diversity measure of ellipsoidal function from $150-1000$ iterations



(c) Diversity measure of ellipsoidal function from $800-1000$ iterations



(d) Diversity measure of ellipsoidal function from $1-1000$ iterations

Fig. 10 Diversity comparison for ellipsoidal function in a single run

and

$$Q_{gi} - Q_{di} - V_i \sum_{k=1}^{nb} V_k (G_{ik} \sin \theta_{ik} + B_{ik} \cos \theta_{ik}) = 0 \quad (28)$$

where P_{gi} and Q_{gi} are the active and reactive powers at i^{th} generator, P_{di} and Q_{di} are the active and reactive power demands at the i^{th} bus, G_{ik} and B_{ik} are the transfer conductance and susceptance between buses i and k, respectively, θ_{ik} is the phase angle difference between the voltages at buses i and k and nb is the total number of bus bars. Also, $H(u, x)$ is the set of system operational limiting constraints which include following inequality constraints:

– Generator constraints:

$$P_{gi}^{min} \leq P_{gi} \leq P_{gi}^{max}, \text{ for } i = 1, 2, ..., ng \quad (29)$$

$$Q_{gi}^{min} \leq Q_{gi} \leq Q_{gi}^{max}, \text{ for } i = 1, 2, .., ng \quad (30)$$

$$V_{gi}^{min} \leq V_{gi} \leq V_{gi}^{max}, \text{ for } i = 1, 2, ..., ng \quad (31)$$

– Security constraints:

$$V_{li}^{min} \leq V_{li} \leq V_{li}^{max}, \text{ for } i = 1, 2, ..., nl \quad (32)$$

$$S_{li} \leq S_{li}^{max}, \text{ for } i = 1, 2, ..., Nl \quad (33)$$

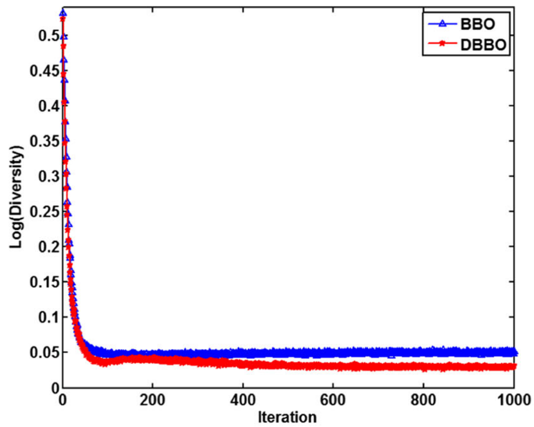
– Transformer constraints:

$$T_i^{min} \leq T_i \leq T_i^{max}, \text{ for } i = 1, 2, ..., nt \quad (34)$$

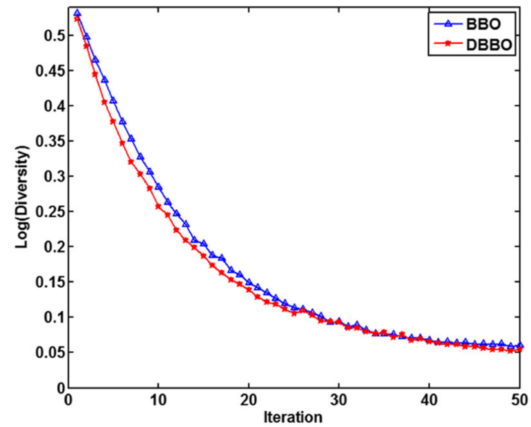
– Shunt VAR compensator constraints:

$$Q_{ci}^{min} \leq Q_{ci} \leq Q_{ci}^{max}, \text{ for } i = 1, 2, ..., nc \quad (35)$$

In the present study, the penalty function method has been adopted to handle constraints with respect to dependent variables i.e, if any dependent variable violates its bound then the square of that violation amount multiplied by a fix penalty factor is added to its corresponding fitness function so that infeasible solutions can be rejected. On the other

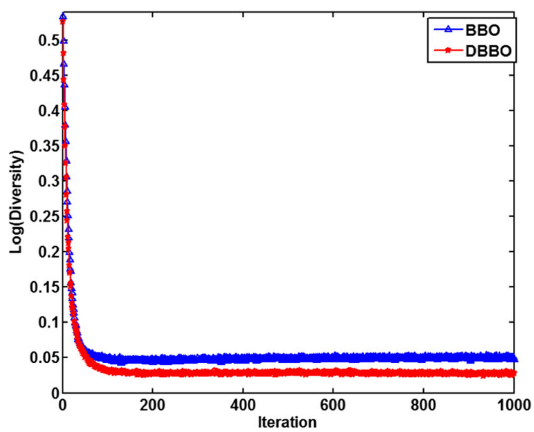


(a) Diversity measure of sphere function from $1-1000$ iterations

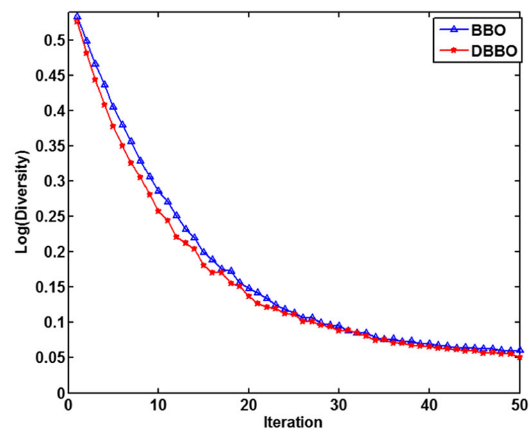


(b) Diversity measure of sphere function from $1-50$ iterations

(c) Diversity measure of sphere function over 30 run

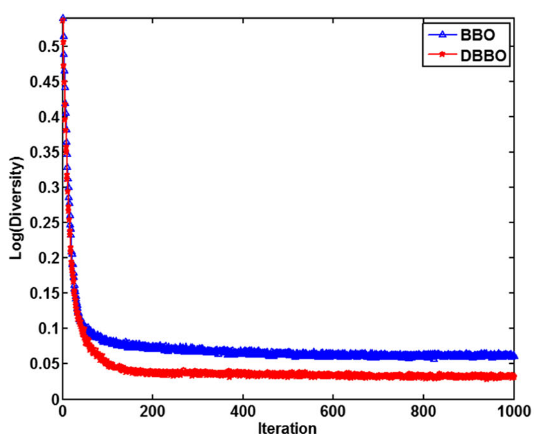


(d) Diversity measure of griewank function from $1-1000$ iterations

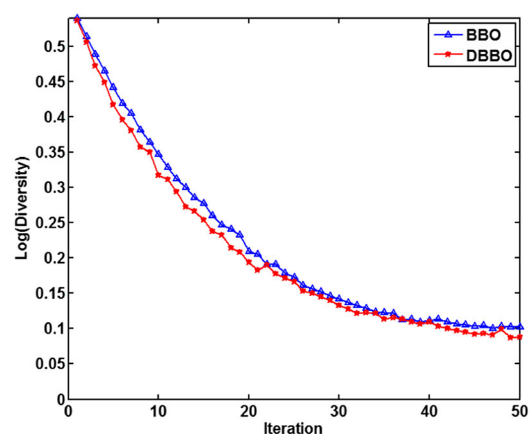


(e) Diversity measure of griewank function from $1-50$ iterations

(f) Diversity measure of griewank function over 30 run



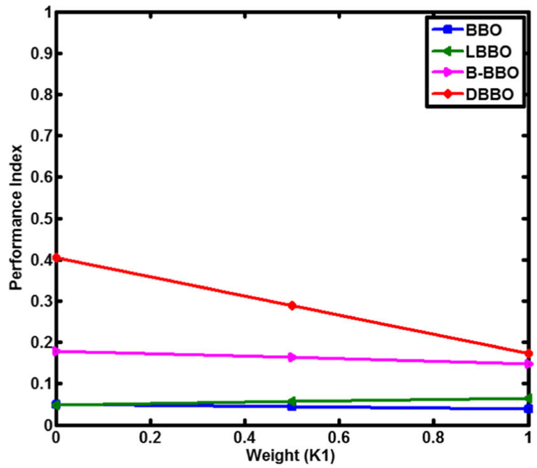
(g) Diversity measure of rastrigin function from $1-1000$ iterations



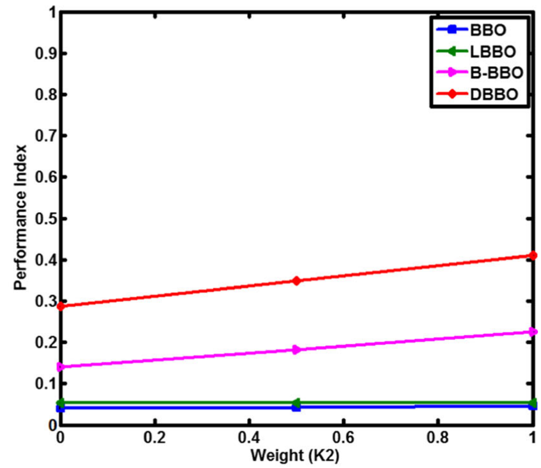
(h) Diversity measure of rastrigin function from $1-50$ iterations

(i) Diversity measure of rastrigin function over 30 run

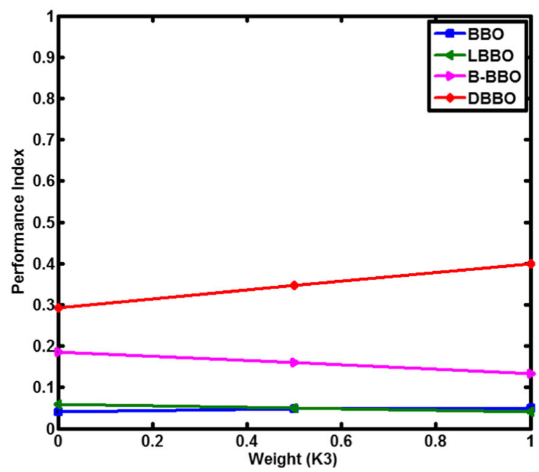
Fig. 11 Comparing diversity BBO and DBBO over average of 30 runs



(a) Performance index for case (1)



(b) Performance index for case (2)



(c) Performance index for case (3)

Fig. 12 Performance index for test problems in several cases

hand, the constraints corresponding to the control variables are handled by generating them between the given bounds in its initialization phase. In this way, the modified objective function for OPF is of the following form:

$$\begin{aligned}\text{Min } F_{mod} = {} & F(u,x) + \lambda_p (P_{g1} - P_{g1}^{lim})^2 \\ & + \lambda_v \sum_{i=1}^{nl} (V_{li} - V_{li}^{lim})^2 \\ & + \lambda_q \sum_{i=1}^{ng} (Q_{gi} - Q_{gi}^{lim})^2 \\ & + \lambda_s \sum_{i=1}^{Nl} (S_{li} - S_{li}^{lim})^2 \end{aligned} \tag{36}$$

where λ_p, λ_v, λ_q and λ_s are the penalty factors (all were set to 10^5 in this study) and a^{lim} is the limit value of the dependent variable a and is given as:

$$a^{lim} = \begin{cases} a^{max}; & a > a^{max} \\ a^{min}; & a < a^{min} \end{cases} \tag{37}$$

5.2.1 Quadratic fuel cost function

The generator cost characteristics are defined as a quadratic cost function of the generator power output and (36) is selected as the objective function to be optimized. In (36), the $F(u,x)$ is designed for this purpose as:

$$F(u,x) = \sum_{i=1}^{ng} f_i(P_{gi}) = \sum_{i=1}^{ng} (a_i + b_i P_{gi} + c_i P_{gi}^2) \tag{38}$$

Table 4 Comparison of DBBO and BBO-MBLX-PM

S. No.	Test Problem	BBO-MBLX-PM (MeanF)	DBBO (Mean)
1	Sphere	1.63E−13	9.2500E−06
2	De Jongs f4	0.00E+00	9.3100E−05
3	Griewank	4.93E−03	**3.91E−03**
4	Rosenbrok	2.51E+01	**2.38E+01**
5	Rastrigin	2.37E−10	1.94E−03
6	Ackley	2.12E−06	9.24E−06
7	Alpine	3.79E−08	9.58E−04
8	Michalewicz	−8.63E+00	**−8.66E+00**
9	Cosine Mixture	**−3.00E+00**	**−3.00E+00**
10	Exponential	**−1.00E+00**	**−1.00E+00**
11	Zakharov's	1.37E−02	**1.36E−02**
12	Cigar	1.60E−07	**7.09E−08**
13	Brown3	1.68E−13	9.53E−10
14	Schewel prob 3	4.74E−07	**3.66E−08**
15	Salomon Problem	3.70E−01	**2.70E−01**
16	Axis parallel hyperellipsoid	3.44E−12	**3.19E−12**
17	Pathological	9.48E−01	**1.71E−01**
18	Rotated hyper-ellipsoid function	3.78E−10	**2.67E−11**
19	Step function	**0.00E+00**	**0.00E+00**
20	Quartic function	8.45E+00	**6.54**

Where f_i and P_{gi} are fuel cost and active power of i^{th} generator, respectively. The variables a_i, b_i and c_i represent the cost coefficients of i^{th} generator whose values have been considered from standard IEEE 30-bus system and are given here in Table 10. The total number of generators is represented by ng in the system.

5.3 DBBO implementation for OPF

DBBO first initializes the $SIVs$ of each island which represents a potential solution. The structure of each island is $u = (P_{g2}....P_{gng}, V_{g1}....V_{gng}, Q_{c1}....Q_{cnc}, T_1....T_{nt})$ where each involved variable is self-constrained i.e., each

Table 5 Comparison of DBBO and BBBO [10]

S.No.	Test Problem	BBBO (MeanE)	BBBO (BestE)	DBBO (MeanE)	DBBO (BestE)
1	Ackley	**4.0410E−08**	**3.5382E−10**	9.6108E−04	6.5183E−04
2	Fletcher	1.0815E+03	8.9584E+02	**1.0461E+03**	**1.3848E+02**
3	Griewank	1.0000E+00	1.0000E+00	**1.4609E−02**	**2.2166E−03**
4	Penalty1	2.3358E−10	**1.2771E−17**	**9.6871E−11**	8.9706E−13
5	Penalty2	4.0188E−10	**9.9903E−16**	**2.2299E−11**	9.4637E−14
6	Quartic	**4.9392E−28**	**5.7239E−33**	1.6467E−20	1.0796E−22
7	Generalised Rastrigin	**3.0731E−03**	**1.7764E−15**	5.0184E−03	2.7020E−03
8	Generalized Rosenbrock	**1.7909E+01**	1.4269E+01	2.3197E+01	**1.3138E−02**
9	Schwefel1.2	1.1579E−02	**2.5455E−04**	**1.1180E−02**	4.4731E−04
10	Schwefel2.21	3.2209E+00	1.3212E+00	**1.2325E−01**	**9.2813E−02**
11	Schwefel2.22	**1.8714E−07**	**2.7737E−08**	2.9836E−05	2.0926E−05
12	Schwefel2.26	6.6094E−01	3.3967E−02	**4.7871E−04**	**3.0605E−04**
13	Sphere	**1.9718E−14**	**4.6913E−17**	3.3336E−06	2.2111E−07
14	Step	3.6727E−01	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**

Table 6 Comparison of DBBO, SG-PSO [24], and SP-PSO [24]

S.No.	Test Problem	SG-PSO (MeanF)	SP-PSO (MeanF)	DBBO (MeanF)
1	Sphere	**0.00E+00**	1.48E−06	1.2076E−04
2	Rosenbrock	2.08E+02	2.20E+01	**2.0163E+01**
3	Schwefel	4.20E+03	4.03E+03	**1.5816E−02**
4	Rastrigin	2.86E+01	3.28E+00	**8.5885E−02**
5	Weierstrass	1.60E+00	**0.00E+00**	6.5885E−06
6	Shifted Sphere	1.24E+04	1.63E+01	**−9.9000E+01**
7	Shifted Schwefel?s Problem 1.2	5.70E+06	1.73E+05	**−1.0100E+02**
8	Shifted Rosenbrock	2.99E+09	1.45E+05	**7.5632E+01**
9	Shifted Rastrigin	1.57E+02	1.29E+02	**−3.2987E+02**
10	Shifted Rotated Rastrigin	2.31E+02	1.72E+02	**−321.05411**
11	Shifted Rotated Weierstrass	2.77E+01	2.89E+01	**1.82E+01**

control variable of the individual island u is initialized randomly within the allowable limits given in Table 11. If any control variable violates its lower or upper limits, then that is adjusted to the corresponding violated limit. To handle the inequality constraints of dependent variables, the modified objective function (36) is considered to be optimized.

5.4 Experimental setting, results and discussion

To validate the effectiveness and robustness of the DBBO algorithm, it has been applied on standard IEEE 30-bus system. This test system consists of 6 generating units at buses 1, 2, 5, 8, 11 and 13 interconnected with 41 transmission lines with a total load of 283.4 MW and 126.2 MVAR

Table 7 Comparison of DBBO and LdDE [12] for 10-dimensional problems

S.No.	Test Problem	LdDE (BestE)	LdDE (MeanE)	LdDE (Std)	DBBO (BestE)	DBBO (MeanE)	DBBO (Std)
1	Shifted sphere function	3.82E−07	6.21E−07	**1.65E−08**	**3.3073E−07**	**4.04E−07**	**2.20E−03**
2	Shifted Schwefels problem 1.2	**4.78E−07**	**7.12E−07**	**1.06E−07**	1.14E−04	2.11E−03	3.92E−05
3	Shifted rotated high conditional elliptic function	9.72E−07	3.03E+01	1.30E+02	**2.81E−07**	**2.27E−01**	**1.68E+01**
4	Shifted Schwefels problem 1.2 with noise in fitness	**2.71E−07**	**8.13E−07**	**1.23E−07**	0.10290097	5.91E+00	1.02E+01
5	Schwefels problem 2.6 with global optimum on bounds	5.78E−07	**8.99E−07**	**1.06E−08**	**2.32E−06**	2.38E−05	6.32E−04
6	Shifted Rosenbrocks function	**6.25E−03**	**4.86E−01**	**2.52E−01**	3.30E+00	7.09E+01	7.27E+01
7	Shifted rotated Griewanks function without bounds No bounds	**5.98E−04**	**1.27E+03**	**3.88E−13**	4.67E−02	1.62E−01	1.01E−01
8	Shifted rotated Ackleys function with global optimum on bounds	2.01E+01	2.02E+01	6.21E−02	**1.91E+01**	**2.00E+01**	**5.23E−02**
9	Shifted Rastrigins function	7.71E−03	3.73E+00	4.89E+00	**7.60E−03**	**1.29E−02**	**5.42E−03**
10	Shifted rotated Rastrigins function	4.98E+00	1.28E+01	7.91E+00	**4.97E+00**	**1.21E+01**	**7.51E+00**
11	Shifted rotated Weierstrass function	8.32E−03	1.69E+00	1.42E+00	**2.15E−03**	**2.98E−01**	**4.68E−02**
12	Schwefels problem 2.13	5.89E−03	6.58E+00	7.61E+00	**3.80E−03**	**3.32E+00**	**4.96E+00**
13	Expanded extended Griewanks plus Rosenbrocks function	3.30E−03	2.23E+00	8.61E−02	**8.16E−04**	**2.27E−03**	**1.51E−03**
14	Shifted rotated expanded Scaffers	2.20E+00	2.07E+00	2.58E−01	**2.10E+00**	**2.07E+00**	**2.24E−01**
15	Hybrid composite function	1.35E+02	4.88E+02	1.86E+02	**1.09E−02**	**2.03E+02**	**2.07E+02**

Table 8 Comparison of DBBO and LdDE [12] for 30-dimensional problems

S.No.	Test Problem	LdDE (BestE)	LdDE (MeanE)	LdDE (Std)	DBBO (BestE)	DBBO (MeanE)	DBBO (Std)
1	Shifted sphere function	7.48E−07	9.14E−07	**5.08E−08**	**6.22E−07**	**8.91E−07**	4.82E−03
2	Shifted Schwefels problem 1.2	**8.82E−07**	**9.69E−07**	**3.05E−08**	6.46E−04	1.55E−03	5.99E−05
3	Shifted rotated high conditional elliptic function	2.58E+03	1.57E+05	2.06E+05	**2.50E+03**	**1.35E+05**	**2.02E+05**
4	Shifted Schwefels problem 1.2 with noise in fitness	**9.81E-07**	**6.35E−01**	**1.80E+00**	2.55E+00	7.48E+00	2.91E+01
5	Schwefels problem 2.6 with global optimum on bounds	1.03E+02	1.28E+03	3.54E+02	**9.94E+01**	**1.11E+03**	**1.01E+02**
6	Shifted Rosenbrocks function	**8.69E−03**	**1.44E−01**	**4.83E−01**	9.49E+01	8.77E+02	2.31E+03
7	Shifted rotated Griewanks function without bounds No bounds	4.69E+03	4.69E+03	**2.91E−12**	**6.31E−01**	**9.66E−01**	1.56E−01
8	Shifted rotated Ackleys function with global optimum on bounds	2.09E+01	**2.09E+01**	2.98E−02	**2.08E+01**	**2.09E+01**	**1.46E−02**
9	Shifted Rastrigins function	3.98E+00	4.82E+01	2.92E+01	**3.29E−02**	**9.77E−02**	**5.33E−02**
10	Shifted rotated Rastrigins function	3.58E+01	7.65E+01	3.85E+01	**3.47E+01**	**6.14E+01**	**2.71E+01**
11	Shifted rotated Weierstrass function	1.32E+01	2.41E+01	5.60E+00	**1.21E+01**	**1.52E+01**	**1.66E+00**
12	Schwefels problem 2.13	2.99E−01	2.05E+03	2.85E+03	**1.99E−01**	**1.25E+03**	**2.75E+03**
13	Expanded extended Griewanks plus Rosenbrocks function	1.37E+00	3.08E+00	1.31E+00	**8.31E−01**	**1.19E+00**	**2.63E−01**
14	Shifted rotated expanded Scaffers	1.09E+01	1.05E+01	1.86E−01	**1.03E+01**	**1.04E+01**	**1.69E−01**
15	Hybrid composite function	1.74E−01	3.37E+02	7.07E+01	**1.12E−01**	**3.24E+02**	**1.14E+01**

and 4 transformers with off-nominal tap ratios in lines 6 − 9, 6 − 10, 4 − 12 and 28 − 27. The shunt injections are provided at buses 10, 12, 15, 17, 20, 21, 23, 24 and 29. Experimental setting is same as the given in Section 4.1 except the total number of simulations. Here 50 simulations are considered for solving OPF problem.

Table 9 Comparison of DBBO and HABC [16]

Sr. No.	Test Problem	HABC (MeanF)	HABC (Std)	DBBO (MeanF)	DBBO (Std)
1	Sphere	0.0000E+00	0.0000E+00	**1.2076E−03**	4.1856E−04
2	Rosenbrock	1.1011E+00	1.5612E+00	2.0193E+01	1.7700E+01
3	Quadric	1.8232 E−02	2.1319 E-02	**1.6612E−02**	**1.4200E−02**
4	Sin	6.6598E−15	6.8549E-15	1.9785E−14	3.1297E−14
5	Rastrigin	0.0000E+00	0.0000E+00	6.5885E−02	3.4415E−02
6	Schwefel	3.9011E+02	1.6016 E+02	**3.5816E−02**	**2.0600E−02**
7	Ackley	7.1161E−07	3.68E−07	**7.8430E−08**	**1.7061E−08**
8	Griewank	3.285 E−04	1.6344 E−03	1.2020E−02	1.3321E−02
9	Shifted Sphere	**−4.5000E+02**	**1.4978E−14**	−4.4992E+02	4.7644E−02
10	Shifted Schwefel	**−1.2042E+02**	**5.4848E−14**	−1.2000E+02	7.8970E−04
11	Shifted rosenbrock	4.0390E+02	3.7741E+01	**1.5632E+01**	**2.1716E+01**
12	Shifted rotated Griewank without bounds	**−1.7999E+02**	**7.7096E−03**	−1.7917E+02	7.7565E−02
13	Shifted rotated Ackley	**−1.1905E+02**	4.8932E−01	**−1.1905E+02**	**4.4297E−02**
14	Shifted Rastrigin	**−3.3000E+02**	**0.0000E+00**	−3.2995E+02	1.6600E−02
15	Shifted rotated Rastrigin	−2.4042E+02	2.4662E+01	**−325.0551**	**6.51E+00**

Table 10 Generator cost coefficients for quadratic fuel cost function

Cost coefficients	Bus No.					
	1	2	5	8	11	13
a	0.00	0.00	0.00	0.00	0.00	0.00
b	2.00	1.75	1.00	3.25	3.00	3.00
c	0.00375	0.01750	0.06250	0.00834	0.02500	0.02500

Table 11 The upper and lower limits of control variables

Control variables	Min	Max	Control variables	Min	Max
P_1	50	200	T_{11}	0.9	1.1
P_2	20	80	T_{12}	0.9	1.1
P_5	15	50	T_{15}	0.9	1.1
P_8	10	35	T_{36}	0.9	1.1
P_{11}	10	30	Q_{c10}	0	5
P_{13}	12	40	Q_{c12}	0	5
V_1	0.95	1.1	Q_{c15}	0	5
V_2	0.95	1.1	Q_{c17}	0	5
V_5	0.95	1.1	Q_{c20}	0	5
V_8	0.95	1.1	Q_{c21}	0	5
V_{11}	0.95	1.1	Q_{c23}	0	5
V_{13}	0.95	1.1	Q_{c24}	0	5
			Q_{c29}	0	5

Table 12 Minimum fuel cost for different methods for IEEE 30-bus system

Optimization methods	Minimum fuel cost ($/h)
ITS [19]	804.5560
IEP[19]	802.4650
SADE-ALM [27]	802.4040
DE-OPF [22]	802.3940
IBF [2]	802.325
MSFLA [18]	802.287
ABC [1]	800.660
BBO	800.4852
DBBO	800.4564

Table 13 shows the optimal control parameter settings achieved by the DBBO and BBO algorithms for OPF problem in the quadratic fuel cost function. Table 12 presents the comparison results of BBO, DBBO and other results from the literature. It can be observed from the Tables 12 and 13 that DBBO gives better results compared to other considered algorithms and is competing with the BBO algorithm with regard to achieving fuel cost.

Table 13 Best control variable settings by BBO and DBBO algorithms for quadratic fuel cost function

Control variables	BBO	DBBO
P_1	177.0859	177.293
P_2	48.6604499	48.69543291
P_5	21.36316102	21.33046969
P_8	21.27492825	21.34609834
P_{11}	11.90827271	11.75659277
P_{13}	12.12408737	12.00449685
V_1	1.081898126	1.086100863
V_2	1.041281442	1.034739184
V_5	1.032920165	1.034431247
V_8	1.039071401	1.037189401
V_{11}	1.066625182	1.07733275
V_{13}	1.053257874	1.041313005
T_{11}	1.022827852	1.080297079
T_{12}	0.942469605	0.9
T_{15}	0.977002946	0.958954479
T_{36}	0.973209406	0.973701054
Q_{c10}	4.997354561	5
Q_{c12}	4.287750133	4.753830765
Q_{c15}	4.189618845	5
Q_{c17}	4.990626993	4.999713444
Q_{c20}	4.330989237	4.656365932
Q_{c21}	4.995054786	5
Q_{c23}	3.501303645	3.494672339
Q_{c24}	4.986178075	5
Q_{c29}	2.430692314	2.502092122
Fuel cost ($/h)	800.4852501	800.4564996

6 Conclusion

This paper presents a modified BBO named as DisruptBBO (DBBO) to improve the exploration and exploitation ability of BBO algorithms. The performance of DBBO is compared with BBO, LBBO, and B-BBO. Through intensive statistical analysis, improvement is shown in terms of reliability, efficiency, and accuracy. To further validate the performance of DBBO over variants of BBO and other state-of-the-art metaheuristics, its performance has been compared with BBO variants such as BBO-MBLX-PM, and BBBO as well as with other state-of-the-art algorithms such as SG-PSO, SP-PSO, LdDE (10-dimensional), LdDE (30-dimensional) and HABC. The comparison of the results show that on average performance of DBBO is comparative with all other considered algorithms. DBBO has also been used to optimize the OPF problem. The results show that performance of DBBO is competitive with that of BBO for solving OPF optimization problem.

The future scope of this work is the implementation of the proposed disruption operator in other nature inspired algorithms to make them more efficient.

Acknowledgment The second author acknowledges the funding from South Asian University, New Delhi, India to carry out this research. Both authors also acknowledges anonymous reviewers Prof. Chander Mohan, IIT Roorkee and Dr. Ashok Pal, Chandigarh University for their intensive reviews.

References

1. Rezaei Adaryani M, Karami A (2013) Artificial bee colony algorithm for solving multi-objective optimal power flow problem. Int J Electr Power Energy Syst 53:219–230
2. Amjady N, Fatemi H, Zareipour H (2012) Solution of optimal power flow subject to security constraints by a new improved bacterial foraging method. IEEE Trans Power Syst 27(3):1311–1323
3. Bansal JC (2016) Modified blended migration and polynomial mutation in biogeography-based optimization. In Harmony Search Algorithm. Springer, pp 217–225
4. Bansal JC, Sharma H, Arya KV, Deep K, Pant M (2014) Self-adaptive artificial bee colony. Optimization 63(10):1513–1532
5. Bansal JC, Sharma H, Jadon SS, Clerc M (2014) Spider monkey optimization algorithm for numerical optimization. Memetic Comput 6(1):31–47
6. Deep K, Thakur M (2007) A new crossover operator for real coded genetic algorithms. Appl Math Comput 188(1):895–911
7. Feng Q, Liu S, Zhang J, Yang G, Yong L (2014) Biogeography-based optimization with improved migration operator and self-adaptive clear duplicate operator. Appl Intell 41(2):563–581
8. Garg V, Deep K (2015) A state-of-the-art review of biogeography-based optimization. Proceedings of Fourth International Conference on Soft Computing for Problem Solving, pages 533–549. Springer
9. Guo W, Chen M, Wang L, Mao Y, Wu Q (2016) A survey of biogeography-based optimization. Neural Comput & Applic:1–18
10. Guo W, Chen M, Wang L, Wu Q (2015) Backtracking biogeography-based optimization for numerical optimization and mechanical design problems. Appl Intell:1–10
11. Harwit M (2006) Astrophysical concepts. Springer Science & Business Media
12. Jana ND, Sil J (2015) Levy distributed parameter control in differential evolution for numerical optimization. Natural Comput:1–14
13. Liu H, Ding G, Wang B (2014) Bare-bones particle swarm optimization with disruption operator. Appl Math Comput 238:106–122
14. Lohokare MR, Pattnaik SS, Panigrahi BK, Das S (2013) Accelerated biogeography-based optimization with neighborhood search for optimization. Appl Soft Comput 13(5):2318–2342
15. Ma H, Simon D (2011) Blended biogeography-based optimization for constrained optimization. Eng Appl Artif Intell 24(3):517–525
16. Ma L, Hu K, Zhu Y, Chen H (2015) A hybrid artificial bee colony optimizer by combining with life-cycle, powells search and crossover. Appl Math Comput 252:133–154
17. MacArthur RH, Wilson EO (1967) The theory of island biogeography, vol 1. Princeton University Press
18. Niknam T, Narimani MR, Jabbari M, Malekpour AR (2011) A modified shuffle frog leaping algorithm for multi-objective optimal power flow. Energy 36(11):6420–6432
19. Ongsakul W, Tantimaporn T (2006) Optimal power flow by improved evolutionary programming. Electr Power Compon Syst 34(1):79–95
20. Riget J, Vesterstrøm JS (2002) A diversity-guided particle swarm optimizer-the arpso. Dept. Comput. Sci., Univ. of Aarhus, Aarhus, Denmark, Tech. Rep, vol 2
21. Sarafrazi S, Nezamabadi-Pour H, Saryazdi S (2011) Disruption: a new operator in gravitational search algorithm. Scientia Iranica 18(3):539–548
22. Sayah S, Zehar K (2008) Modified differential evolution algorithm for optimal power flow with non-smooth cost functions. Energy Convers Manag 49(11):3036–3042
23. Sharma H, Bansal JC, Arya KV (2014) Self balanced differential evolution. J Comput Sci 5(2):312–323
24. Shin Y-B, Kita E (2014) Search performance improvement of particle swarm optimization by second best particle information. Appl Math Comput 246:346–354
25. Simon D (2008) Biogeography-based optimization. IEEE Trans Evol Comput 12(6):702–713
26. Simon D, Omran MGH, Clerc M (2014) Linearized biogeography-based optimization with re-initialization and local search. Inf Sci 267:140–157
27. Vaisakh K, Srinivas LR (2011) Evolving ant direction differential evolution for opf with non-smooth cost functions. Eng Appl Artif Intell 24(3):426–436
28. Wang G-G, Gandomi AH, Alavi AH (2014) An effective krill herd algorithm with migration operator in biogeography-based optimization. Appl Math Model 38(9):2454–2462
29. Xiong G, Shi D, Duan X (2014) Enhancing the performance of biogeography-based optimization using polyphyletic migration operator and orthogonal learning. Comput Oper Res 41:125–139

Dr. Jagdish Chand Bansal is an Assistant Professor at South Asian University Delhi, India. He has worked as an Assistant Professor at ABV-Indian Institute of Information Technology and Management Gwalior. He has obtained his Ph.D. in Mathematics from IIT Roorkee. He is the editor in chief of “International Journal of Swarm Intelligence (IJSI)” published by Inderscience. His primary area of interest is Nature Inspired Optimization Techniques.

Pushpa Farswan received her B. Sc. Degree in Physics, Chemistry, Mathematics and M. Sc. Degree in Mathematics from Kumaun University Nainital, Uttarakhand, India in 2007 and 2009, respectively. She is currently a Research Scholar at South Asian University, New Delhi, India.