

# A new hybrid differential evolution algorithm with self-adaptation for function optimization

Sukanta Nama<sup>1</sup> · Apu Kumar Saha<sup>1</sup>

Published online: 31 August 2017  
© Springer Science+Business Media, LLC 2017

**Abstract** In this study, a new hybrid algorithm, hDEBSA, is proposed with the aid of two evolutionary algorithms, Differential Evolution (DE) and Backtracking Search Optimization Algorithm (BSA). The control parameters of both algorithms are simultaneously considered as a self-adaptation basis such that the values of the parameters update automatically during the optimization process to improve performance and convergence speed. To validate the proposed algorithm, twenty-eight CEC2013 test functions are considered. The performance results of hDEBSA are validated by comparing them with several state-of-the-art algorithms that are available in literature. Finally, hDEBSA is applied to solve four real-world optimization problems, and the results are compared with the other algorithms, where it was found that the hDEBSA performance is better than that of the other algorithms.

**Keywords** Global optimization algorithm · Backtracking search optimization algorithm · Differential evolution · Hybrid algorithm · CEC2013

## 1 Introduction

Differential evolution (DE) [17] is a powerful evolutionary algorithm that is also easy to implement. DE incorporates

two control parameters, the scaling factor (F) and crossover rate (CR), and its performance depends on the choice of these two control parameters, which is why many authors have studied and are still studying DE to obtain suitable values of F and CR. Several of these studies are discussed below:

SPDE [25] is based on self-adaptation of the F and CR values, where the values of the control parameters are generated from a Gaussian distribution  $N(0, 1)$ . In jDE [26], self-adaptation of F and CR values are considered, where the value of F is generated within the range [0.1, 1.0] with probability  $\tau_1$ , and CR is generated within the range [0, 1] with probability  $\tau_2$ . In JADE [27], the value of F is generated by a Cauchy distribution, whereas the value of CR is generated using a normal distribution. In SaDE [28], the value of F is calculated from a normal distribution with mean of 0.5 and standard deviation of 0.3, denoted by  $N(0.5, 0.3)$ , and the CR value is calculated from a Gaussian distribution; these F and CR values are applied to each target vector. EPSDE [29] is based on the ensemble of multiple mutation strategies with multiple parameter settings of control parameters during different stages of evolution. CoDE [30] is based on the combination of three different trial vector generation strategies associated with three different parameter settings of F and CR. In MPEDE [37], a multi-population based ensemble of multiple strategies (i.e., “rand/1”, “current-to-rand/1”, and “current-to-pbest/1”) has been proposed.

However, recently, researchers have studied several hybrid algorithm that combines DE with other algorithms. Examples of these hybrid algorithms include DE-PSO [19], which is the combination of DE [17] and PSO [6, 15]; DESQI [18], a combination of DE and QA; BBDE [20], a combination of bare bones PSO and differential evolution; DE/BBO [21], a combination of DE and BBO [24];

---

✉ Apu Kumar Saha  
apusaha\_nita@yahoo.co.in  
Sukanta Nama  
sukanta1122@gmail.com

<sup>1</sup> Department of Mathematics, National Institute of Technology Agartala, Barjala, Jirania 799046, West Tripura, India

GA-DE [22], a combination of GA [8] and DE; ABC-DE [23], a combination of ABC [1] and DE; and CADE [38], a combination of the cultural algorithms (CA) and DE algorithm. Additional improved hybrid algorithms can be found in [53–55]. In this work, a novel hybrid algorithm, hDEBSA, is proposed by combining DE with a newly proposed evolutionary algorithm, called the backtracking search algorithm (BSA) [4, 39], where BSA is a population-based nature-inspired optimization technique that utilizes mutation, crossover, and selection operators during each evolution to move each individual towards the global optimum. Many researchers have improved the performance of BSA with respect to the different self-adaptive strategy designs [21, 39, 42] and hybridization [43–47], and these versions of BSA are widely used to solve various complex global optimization problems from a variety of fields, such as antenna array synthesis [48], power systems optimization [49], trusses structure [50], urban traffic network [51], and surface wave analysis [52].

The aforementioned facts have motivated us to work further work on DE and BSA, where we attempt to find a new hybrid algorithm that combines the features of DE and BSA. The primary contributions of this study are abridged as follows:

- i) A hybrid algorithm, i.e., hDEBSA, is proposed, which uses the components of DE and BSA,
- ii) A self-adaptation scheme for control parameters are used in hDEBSA to improve the performance as well as the convergence rate of the proposed algorithm,
- iii) The proposed hDEBSA is applied to solve four real-world optimization problems.

The remaining part of the paper is organized as follows: Section 2 discusses the two components of hDEBSA, i.e., the basic DE and BSA. The proposed hDEBSA is presented in Section 3. Section 4 presents the performance evaluation of twenty-eight CEC 2013 test functions. Section 5 presents the formulation of four real-world optimization problems, the results, and a discussion of these optimization problems. Finally, Section 6 summarizes the contribution of this study.

## 2 The basic DE and BSA algorithms

A brief description of the basic DE and BSA are given in the following sub-sections:

### 2.1 Differential evolution algorithm

Differential evolution is a population-based evolutionary algorithm, which incorporates two important algorithm specific control parameters. One is a weighting coefficient or scaling factor (F), and the other is the crossover rate (CR). The scaling factor (F) is used to generate new trial solutions

when executing the optimization process. The crossover rate (CR) is used to determine how much of a trial solution should be adopted into the current solution. It has been found that the performance of the DE algorithm depends on the proper values of F and CR [2], and varying the values of F and CR during the execution of the optimization process can improve its performance [3]. In the DE algorithm, the mutation operation is used in the current population to produce a mutant vector, where the crossover operator is used to produce the final form of the trial population; the selection operator is used between the trial population and target population to update the current population. By repeated cycles of the mutation operator, crossover operator, and selection operator, DE attempts to improve its performance. A detailed description of the DE algorithm is given in [17].

### 2.2 Backtracking search optimization algorithm

BSA is also a population-based stochastic evolutionary algorithm and incorporates two algorithm specific control parameters, i.e., the scaling factor (F) and mix rate (M). BSA uses the historical population to identify the search direction. The initial historical population is obtained by a uniform random generation strategy within the search space. BSA employs the one direction mutation strategy, which is different from other evolutionary algorithms. During the production of the trial population ‘Mutant’, parameter F controls the amplitude of the search-direction matrix. Once the mutant operation has ended, the crossover process is used to produce the final form of the trial population. The process of crossover strategy is divided into two steps. At first, a binary integer-valued matrix (map) of size  $NP \times D$  (where  $NP$  = number of population and  $D$  = dimension of the optimization problem) is calculated, which indicates the individuals of ‘T’ (trial population) to be manipulated using the relevant individuals of ‘P’ (current population). Secondly, using the relevant individual to the mutant individual, the relevant dimensions of the mutant individual are updated. A detailed description of BSA is given in [4].

## 3 Proposed hDEBSA algorithm

The combination of one meta-heuristic optimization technique with other optimization techniques or the component of any optimization technique is called a hybrid meta-heuristic optimization technique. An efficient hybrid algorithm enables more efficient behaviour and a higher flexibility when dealing with real-world and large-scale optimization problems.

However, the performance of any algorithm depends on the choice of the proper values of its algorithm control

parameters. DE incorporates two control parameters, F and CR, and the performance of DE depends on the proper choice of these to control parameters. Also, the performance of BSA depends on F. A lower value of F permits a fine search in small steps but is slow to converge, and a larger value of F speeds up the search but reduces the exploration capability.

Considering this fact, in this work, a hybrid algorithm, called hDEBSA, is proposed by combining the two algorithms, DE and BSA. Also, the control parameters are considered on a self-adaptive basis. In hDEBSA, first, the components of the DE algorithm are executed, and then the components of BSA are executed. When executing the component of the DE algorithm, the worst individual is updated

according to probability  $p_i$ . The value of probability  $p_i$  can be calculated by (1).

$$p_i = \frac{r_i}{NP} \tag{1}$$

Where  $NP$  is the population size, and  $r_i$  is the ranking value of each individual when the population is sorted from the worst fitness to the best one. It may be noted here that (1) is similar to the selection probability in DE with ranking-based mutation operators [7, 11]. This selection strategy can be defined as follows:

$$I_s = i, \text{ if } \text{rand}(0, 1) > p_i, i = 1, 2, 3, \dots, N \tag{2}$$

Where  $I_s$  is the selected individual and is optimized by DE.

**Fig. 1** Pseudo code of the proposed hDEBSA algorithm

```

Input:
NP: The number of individual in the population; D: Dimension of the problem; MAX_FES: Set the maximum function evaluations number;
ll1:D = lower limit; ul1:D = upper limit; FDEmin = 0.4; FDEmax = 0.95; CRDEmin = 0.3; CRDEmax = 0.9, FBSAmin = 0.45, FBSAmax = 2.0;
Initialization:
FOR i=1 to NP do
FOR j=1 to D
Pi,j = lli + rand(0,1) * (uli - lli); % initialized population set;
End
f(Pi) = ObjFun(Pi); % Evaluate the fitness values of initial population P.
End
f0max = max(f); f0min = min(f);
FES=NP; % Count the number of fitness evaluations %;
% Main loop %
WHILE (stopping criteria is not meet)
DO
FDE = FDEmin + (FDEmax - FDEmin) * (fimax - fimin) / (f0max - f0min); % Calculate the scaling factor for DE;
CRDE = CRDEmax - (CRDEmax - CRDEmin) * (fimax - fimin) / (f0max - f0min); % Calculate the crossover rate for DE;
pi =  $\frac{r_i}{NP}$  % Calculate the probability for update one individual;
Is = i, if rand(0,1) > pi, i = 1,2,3, ... N. % Is is selected individual DE.
vIs = PIsr1 + FDE * (PIsr2 - PIsr3); % Update individual by DE "DE/rand/1" mutation operator;
PIs,j =  $\begin{cases} v_{I_s,j} & \text{if } \text{rand}_j(0,1) < CR_{DE} \\ P_{I_s,j} & \text{Otherwise} \end{cases}$  % Calculate the crossover operator for DE; j=1, 2, 3, ... , D;
FES=FES+1; % Count the number of fitness evaluations %;
OldP = P; % initialized historical population set;
% Selection operator-I
if a < b, OldP = P; end; % a, b ∈ rand(0,1);
OldP = permutating(P); % changing the positions of two individuals in OldP;
% Mutation operator
FBSA = FBSAmax - rand(0,1) * (FBSAmax - FBSAmin) + rand(0,1) * (fimax - fimin) / (f0max - f0min); % Calculate the scaling factor for BSA;
FOR i=1 to NP
Mi = 0.5 * (1 + rand(0,1)); % calculates the mixrate parameter;
Mutanti,1:D = Pi,1:D + FBSA * (OldPi,1:D - Pi,1:D); % calculates the mutant vector;
% Crossover operator
map=zeros(popsiz,D);
if a < b, % where, a and b are uniformly distributed random number.
u=randperm(D); map(i,u(1:ceil(Mi*rand*D)))=1; end
else
map(i,randi(D))=1; end
end
% Generation of trial population T;
Ti,1:D = Mutanti,1:D;
if mapi,1:D = 1, Ti,1:D = Pi,1:D; end;
% Boundary control Operator
if Ti,1:D < ll1,1:D or Ti,1:D > ul1,1:D,
Ti,1:D = ll1,1:D + rand(0,1) * (ul1,1:D - ll1,1:D);
end
END FOR
FES=FES + NP; % Count the number of fitness evaluations %;
% Selection operator-II
FOR i=1 to NP
% Selection by competition between target vector and trial vector and evaluate the new trial vector.
if f(Ti) ≤ f(Pi) then Pi = Ti, f(Pi) = f(Ti);
if f(Ti) ≤ f(Pbest) then Pbest = Ti, f(Pbest) = f(Ti);
end;
end;
END FOR
END WHILE LOOP
Output: The best individual with the minimum fitness function value in the population (for minimization problem);
    
```

**Table 1** Performance results of PSO, DE ABC, BSA, ABSA and hDEBSA at dimension(D) 30 after reaching 3000 FEs of CEC2013 test functions over 30 runs with 50 population size

Function	PSO [15]		DE [17]		ABC [1]		BSA [4]		ABSA [31]		hDEBSA	
	Mean (STD.)	Mean (STD.)	Mean (STD.)	Mean (STD.)	Mean (STD.)	Mean (STD.)	Mean (STD.)	Mean (STD.)	Mean (STD.)	Mean (STD.)	Mean (STD.)	Mean (STD.)
F1	3.76e+004 (5.37e+003)	1.33e+004 (3.04e+003)	1.67e+004 (7.79e+003)	2.41e+004 (4.65e+003)	3.39e+004 (7.56e+003)	<b>8.15e+003</b> (2.21e+003)						
F2	7.04e+008 (2.29e+008)	4.71e+008 (1.21e+008)	1.74e+008 (7.71e+007)	4.14e+008 (1.11e+008)	5.43e+008 (1.08e+008)	<b>2.80e+008</b> (6.52e+007)						
F3	4.42e+017 (1.67e+018)	3.82e+011 (1.59e+011)	7.59e+012 (4.08e+013)	1.79e+013 (2.80e+013)	3.52e+012 (5.40e+012)	<b>1.51e+011</b> (1.40e+011)						
F4	<b>6.84e+004</b> (1.39e+004)	2.15e+005 (3.79e+004)	1.07e+005 (1.86e+004)	1.09e+005 (1.72e+004)	1.37e+005 (2.27e+004)	1.02e+005 (1.83e+004)						
F5	8.74e+003 (3.00e+003)	3.26e+003 (1.01e+003)	3.47e+003 (4.14e+003)	9.74e+003 (3.89e+003)	1.64e+004 (4.74e+003)	<b>2.06e+003</b> (7.64e+002)						
F6	7.55e+003 (1.81e+003)	1.22e+003 (5.98e+002)	1.08e+003 (6.30e+002)	3.14e+003 (1.10e+003)	4.50e+003 (9.10e+002)	<b>9.29e+002</b> (3.04e+002)						
F7	3.35e+005 (7.49e+005)	5.48e+002 (1.24e+002)	1.53e+003 (2.55e+003)	2.27e+003 (2.86e+003)	1.49e+003 (1.66e+003)	<b>3.67e+002</b> (2.22e+002)						
F8	2.12e+001 (6.35e+002)	<b>2.11e+001</b> (6.69e+002)	<b>2.11e+001</b> (5.58e+002)	2.12e+001 (5.65e+002)	<b>2.11e+001</b> (6.18e+002)	<b>2.11e+001</b> (4.61e+002)						
F9	4.35e+001 (2.47e+000)	4.44e+001 (1.27e+000)	<b>3.90e+001</b> (1.71e+000)	4.17e+001 (1.40e+000)	4.12e+001 (1.89e+000)	4.17e+001 (1.53e+000)						
F10	5.68e+003 (1.34e+003)	2.61e+003 (5.14e+002)	2.28e+003 (8.13e+002)	4.30e+003 (1.12e+003)	5.20e+003 (7.39e+002)	<b>2.14e+003</b> (4.34e+002)						
F11	5.62e+002 (7.94e+001)	4.21e+002 (3.56e+001)	<b>3.06e+002</b> (7.74e+001)	5.81e+002 (9.31e+001)	6.94e+002 (8.24e+001)	3.20e+002 (4.52e+001)						
F12	6.35e+002 (6.51e+001)	4.51e+002 (5.33e+001)	7.26e+002 (9.49e+001)	6.09e+002 (8.93e+001)	7.75e+002 (8.02e+001)	<b>4.21e+002</b> (5.83e+001)						
F13	6.51e+002 (7.84e+001)	4.73e+002 (5.71e+001)	7.93e+002 (9.96e+001)	6.42e+002 (1.19e+002)	7.85e+002 (9.90e+001)	<b>4.00e+002</b> (4.75e+001)						
F14	7.47e+003 (6.71e+002)	8.68e+003 (2.65e+002)	<b>4.18e+003</b> (4.74e+002)	6.77e+003 (2.80e+002)	6.54e+003 (3.75e+002)	6.02e+003 (4.33e+002)						
F15	7.81e+003 (4.75e+002)	8.79e+003 (3.57e+002)	<b>7.30e+003</b> (3.77e+002)	8.32e+003 (3.38e+002)	8.53e+003 (2.76e+002)	8.12e+003 (4.81e+002)						
F16	3.89e+000 (5.32e-001)	4.00e+000 (5.81e-001)	<b>3.27e+000</b> (5.01e-001)	3.88e+000 (3.24e-001)	4.09e+000 (3.90e-001)	4.06e+000 (5.28e-001)						
F17	6.16e+002 (5.90e+001)	6.79e+002 (9.44e+001)	6.40e+002 (1.84e+002)	9.90e+002 (2.69e+002)	1.46e+003 (1.70e+002)	<b>5.08e+002</b> (1.13e+002)						
F18	6.53e+002 (7.82e+001)	6.48e+002 (6.92e+001)	1.10e+003 (1.86e+002)	1.03e+003 (2.82e+002)	1.60e+003 (2.17e+002)	<b>5.89e+002</b> (1.13e+002)						
F19	1.34e+005 (6.71e+004)	7.09e+004 (5.93e+004)	7.23e+004 (2.60e+005)	3.05e+005 (3.83e+005)	5.52e+005 (4.12e+005)	<b>7.99e+003</b> (6.60e+003)						
F20	<b>1.50e+001</b> (3.30e-003)	<b>1.50e+001</b> (8.96e-002)	<b>1.50e+001</b> (2.23e-005)	<b>1.50e+001</b> (1.69e-002)	<b>1.50e+001</b> (9.53e-003)	<b>1.50e+001</b> (2.59e-002)						
F21	2.25e+003 (9.40e+001)	2.31e+003 (1.83e+002)	2.35e+003 (3.93e+002)	2.99e+003 (4.58e+002)	4.03e+003 (4.61e+002)	<b>2.11e+003</b> (1.53e+002)						
F22	8.22e+003 (5.40e+002)	9.04e+003 (3.65e+002)	<b>4.99e+003</b> (4.60e+002)	7.53e+003 (3.70e+002)	7.43e+003 (3.51e+002)	6.94e+003 (4.07e+002)						
F23	8.57e+003 (5.85e+002)	9.11e+003 (3.46e+002)	<b>8.11e+003</b> (6.55e+002)	8.83e+003 (4.15e+002)	8.92e+003 (2.76e+002)	8.84e+003 (3.99e+002)						
F24	4.50e+002 (4.21e+001)	<b>3.14e+002</b> (4.85e+000)	3.36e+002 (1.03e+001)	3.28e+002 (6.64e+000)	3.25e+002 (5.05e+000)	3.18e+002 (6.91e+000)						
F25	4.30e+002 (1.86e+001)	<b>3.10e+002</b> (4.46e+000)	3.63e+002 (1.06e+001)	3.40e+002 (5.76e+000)	3.29e+002 (3.49e+000)	3.48e+002 (5.01e+000)						
F26	3.83e+002 (6.76e+001)	4.07e+002 (1.60e+001)	2.30e+002 (1.91e+001)	2.61e+002 (2.43e+001)	2.98e+002 (4.48e+001)	2.45e+002 (2.03e+001)						
F27	1.60e+003 (1.13e+002)	1.43e+003 (2.72e+001)	1.46e+003 (5.77e+001)	1.44e+003 (4.00e+001)	1.42e+003 (4.67e+001)	<b>1.41e+003</b> (4.03e+001)						
F28	4.88e+003 (4.82e+002)	3.75e+003 (4.99e+002)	5.05e+003 (8.07e+002)	5.32e+003 (5.67e+002)	5.64e+003 (6.07e+002)	<b>3.60e+003</b> (5.13e+002)						

“Mean” and “STD” show the average and standard deviation of the test function respectively. Boldface represents the best result among the compared algorithms

**Table 2** Counting the mean performances result of hDEBSA, is better than, less than or equal to the other compared algorithms (PSO, DE, ABC, BSA, ABSA) on CEC2013 real world benchmark functions with 30 dimensions

Algorithms	Better than (hDEBSA>Algorithms)	Less than (hDEBSA<Algorithms)	Equal to (hDEBSA=Algorithms)
PSO	23	4	1
DE	23	3	2
ABC	18	2	8
BSA	23	3	2
ABSA	24	2	2

A detailed description of self-adaptive-based control parameters setting and the proposed hDEBSA algorithm is given below:

### 3.1 Scaling factor/ weighting coefficient (F) of the DE algorithm

In the DE algorithm, the scaling factor (F) is used to produce a new set of the trial vector. It has been found that a smaller value (less than 0.4) and a larger value (greater than 1.0) of F are occasionally effective [17]. Several researchers have also observed that a large control parameter F reduces the local optimum [12, 13]. Gämperle et al. [12] found that F= 0.6 or 0.5 may be the proper initial value, whereas Rönkkönen et al. [13] found it to be F = 0.9. According to Rönkkönen et al. [13], the value of control parameter F lies in the range 0.4–0.95. Varying the value of control parameter F during the optimization process, one can improve the performance of the DE algorithm [3, 17]. Thus, the modification of the scaling factor (F) can be defined by (3). For clarity, instead of F, the variable is denoted  $F_{DE}$

$$F_{DE} = F_{DE}^{max} - (F_{DE}^{max} - F_{DE}^{min}) * \frac{f_i^{max} - f_i^{min}}{f_0^{max} - f_0^{min}} \quad (3)$$

where  $F_{DE}^{min} = 0.4$  and  $F_{DE}^{max} = 0.95$ ;  $f_0^{max}$  and  $f_0^{min}$  are the maximum and minimum fitness values of the initial population, respectively;  $f_i^{max}$  and  $f_i^{min}$  are the maximum and minimum fitness values of the population in the  $i^{th}$  iteration, respectively.

**Table 3** Ranks obtained by Friedman’s test in which Bonferroni–Dunn’s procedure was used as a post hoc procedure on the mean performance of CEC2013 test function with dimension 30. The Friedman’s test was conducted based on the SPSS software

Algorithms	Ranks
PSO	4.32
DE	3.39
ABC	2.89
BSA	4.02
ABSA	4.46
hDEBSA	1.91

### 3.2 Crossover rate (CR) of the DE algorithm

In DE, the crossover rate (CR) is used to produce the final form of the trial vector set. It is used to determine how much of a trial solution should be adopted into the current solution as well as the DE scheme. This is one of the crucial ideas behind DE for generating trial vectors [9, 16]. Researchers have verified that a large CR speeds up convergence but reduces the local search ability [12–14]. The value of CR = 0.1 is the proper initial choice, whereas a CR = 0.9 or 1.0 can improve the convergence speed [17]. The proper value of CR can be between 0.3 and 0.9 [12]. When CR =1, the number of the trial vectors may be reduced dramatically, which may lead to immobility [10, 13]. By varying the value of CR during the execution of the optimization process, one can improve the performance of DE [3]. The modification of the crossover rate (CR) can be defined by (4). Instead of CR, it is written as  $CR_{DE}$

$$CR_{DE} = CR_{DE}^{max} - (CR_{DE}^{max} - CR_{DE}^{min}) * \frac{f_i^{max} - f_i^{min}}{f_0^{max} - f_0^{min}} \quad (4)$$

where  $CR_{DE}^{min} = 0.3$  and  $CR_{DE}^{max} = 0.9$ ;  $f_0^{max}$  and  $f_0^{min}$  are the maximum and minimum fitness values of the initial population, respectively;  $f_i^{max}$  and  $f_i^{min}$  are the maximum and minimum fitness values of the population in the  $i^{th}$  iteration, respectively. Also, the value of another BSA control parameter, i.e., the mixrate, is considered as

$$mixrate_{BSA} = 0.3 * (1 + rand(0, 1)) \quad (5)$$

The pseudocode of the proposed hDEBSA algorithm is presented in Fig. 1.

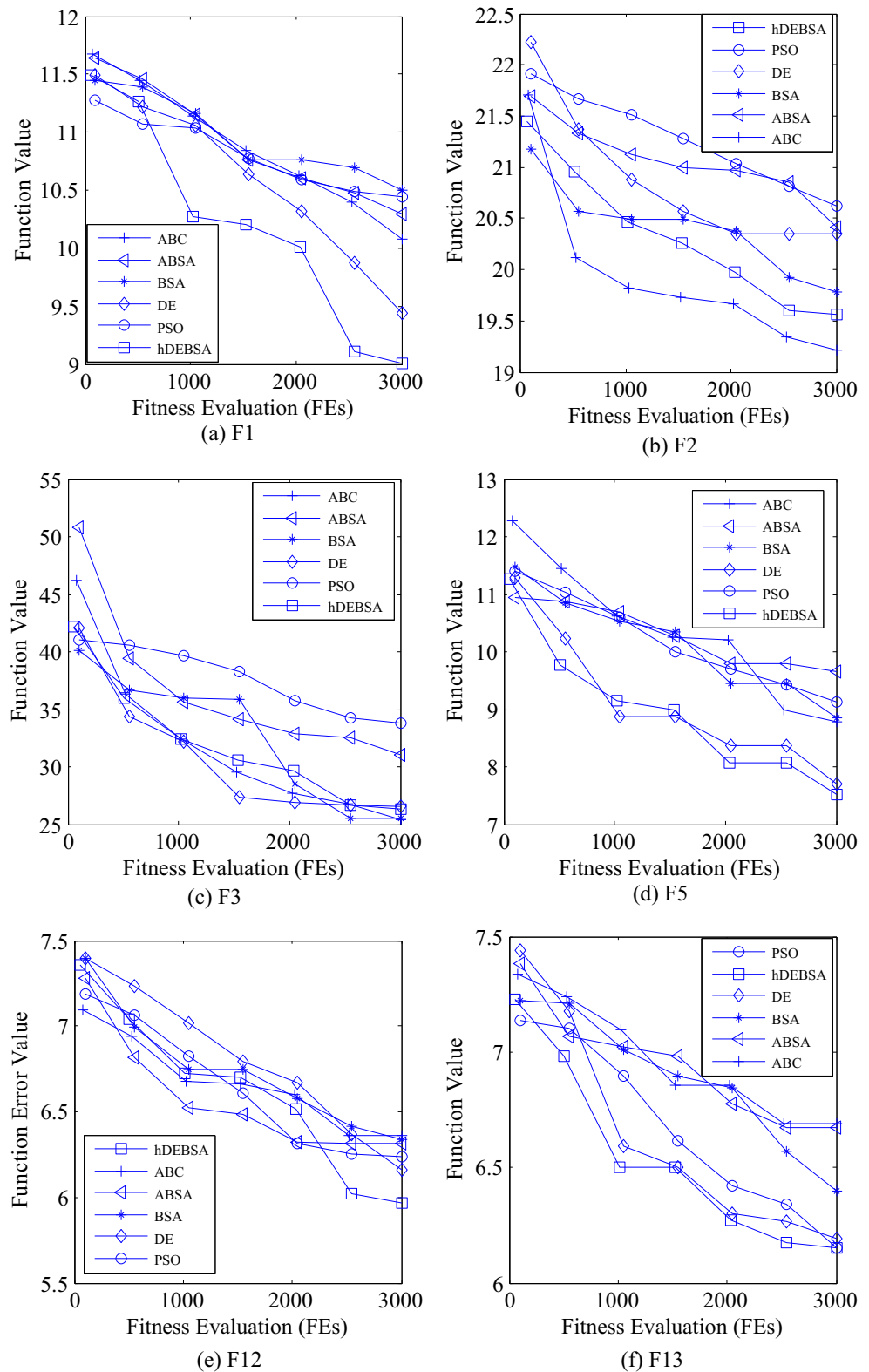
## 4 Performance evaluation on the CEC 2013 test functions

To validate the performance of hDEBSA, twenty-eight benchmark functions from the CEC 2013 special session on real-parameter optimization [5] are considered. These test functions consist of three different types of functions: (i) unimodal functions F1–F5, (ii) basic multimodal functions F6– F20, and (iii) composition functions F21– F28. A detailed description of these twenty-eight test functions can

be seen in [5]. For this study, the dimension (D) of each test function is considered as 30 and 50. The algorithm is run for 30 times with 3000 (for D = 30) and 5000 (for D = 50) fitness evaluations (FEs) for a population size of 50. The

range of each test function is considered as  $[-100, 100]$ . The performance results are presented in terms of the mean and standard deviation of each test function. For the statistical analysis, the Friedman Rank Test is used to obtain the

**Fig. 2** Convergence graph of F1, F2, F3, F5, F12 and F13



**Table 4** Performance results of CoDE, DE/rand/2/bin, CLPSO, CPSON-H, FI-PSO, DE-PSO and hDEBSA at dimension(D) 30 after reaching 3000 FEs of CEC2013 test functions over 30 runs with 50 population size

Function	CoDE [30]	DE/rand/2/bin [28]	CLPSO [32]	CPSON-H [33]	FI-PSO [34]	DE-PSO [19]	hDEBSA
	Mean(STD.)	Mean(STD.)	Mean (STD.)	Mean (STD.)	Mean (STD.)	Mean (STD.)	Mean(STD.)
F1	1.88e+004 (3.28e+003)	2.78e+004 (4.40e+003)	4.26e+004 (6.22e+003)	<b>2.51e+002</b> (1.02e+002)	1.90e+004 (2.74e+003)	1.12e+004 2.85e+003	8.15e+003 (2.21e+003)
F2	3.83e+008 (9.74e+007)	5.28e+008 (8.98e+007)	7.68e+008 (2.56e+008)	4.45e+008 (2.15e+008)	4.03e+008 (6.19e+007)	<b>2.00e+008</b> ±5.79e+007	2.80e+008 (6.52e+007)
F3	1.61e+012 (3.06e+012)	3.07e+013 (6.32e+013)	1.37e+015 (2.75e+015)	1.83e+015 (7.23e+015)	<b>1.43e+011</b> (3.03e+010)	3.73e+011 ±5.96e+011	1.51e+011 (1.40e+011)
F4	1.03e+005 (2.02e+004)	1.11e+005 (1.90e+004)	1.25e+005 (1.98e+004)	1.72e+005 (3.64e+004)	1.41e+005 (2.49e+004)	<b>8.04e+004</b> ±1.09e+004	1.02e+005 (1.83e+004)
F5	4.35e+003 (9.97e+002)	4.17e+003 (7.83e+002)	1.60e+004 (4.63e+003)	<b>1.37e+002</b> (1.04e+002)	5.44e+003 (1.30e+003)	1.61e+003 ±3.82e+002	2.06e+003 (7.64e+002)
F6	2.32e+003 (5.38e+002)	3.51e+003 (7.94e+002)	5.47e+003 (1.65e+003)	1.16e+003 (4.05e+002)	1.57e+003 (3.13e+002)	1.36e+003 ±3.52e+002	<b>9.29e+002</b> (3.04e+002)
F7	8.19e+002 (7.39e+002)	4.82e+003 (4.02e+003)	3.38e+004 (4.13e+004)	2.34e+004 (4.75e+004)	<b>3.36e+002</b> (4.51e+001)	3.43e+002 ±1.66e+002	3.67e+002 (2.22e+002)
F8	<b>2.11e+001</b> (6.64e-002)	2.12e+001 (6.55e-002)	<b>2.11e+001</b> (8.13e-002)	2.12e+001 (7.34e-002)	<b>2.11e+001</b> (5.54e-002)	2.12e+001 ±4.99e-002	<b>2.11e+001</b> (4.61e-002)
F9	4.37e+001 (9.37e-001)	4.45e+001 (9.74e-001)	4.19e+001 (1.82e+000)	<b>4.14e+001</b> (2.76e+000)	4.40e+001 (1.51e+000)	4.24e+001 ±1.73e+000	4.17e+001 (1.53e+000)
F10	3.16e+003 (4.95e+002)	4.43e+003 (7.92e+002)	6.32e+003 (1.12e+003)	2.74e+003 (8.60e+002)	3.13e+003 (5.11e+002)	<b>1.98e+003</b> ±4.23e+002	2.14e+003 (4.34e+002)
F11	4.88e+002 (4.05e+001)	5.45e+002 (5.01e+001)	6.88e+002 (6.87e+001)	4.94e+001 (9.13e+000)	5.02e+002 (3.96e+001)	4.01e+002 ±4.09e+001	<b>3.20e+002</b> (4.52e+001)
F12	5.24e+002 (5.13e+001)	6.18e+002 (4.82e+001)	8.09e+002 (9.58e+001)	6.06e+002 (1.31e+002)	5.02e+002 (3.52e+001)	4.19e+002 ±3.13e+001	<b>4.21e+002</b> (5.83e+001)
F13	5.33e+002 (4.49e+001)	6.26e+002 (4.41e+001)	7.89e+002 (1.04e+002)	5.85e+002 (9.72e+001)	5.07e+002 (4.25e+001)	4.42e+002 ±5.21e+001	<b>4.00e+002</b> (4.75e+001)
F14	7.24e+003 (3.72e+002)	8.31e+003 (3.27e+002)	7.29e+003 (5.10e+002)	6.24e+002 (1.63e+002)	8.32e+003 (2.27e+002)	8.01e+003 ±4.20e+002	<b>6.02e+003</b> (4.33e+002)
F15	8.56e+003 (2.67e+002)	8.62e+003 (3.78e+002)	8.32e+003 (4.21e+002)	<b>7.21e+003</b> (7.54e+002)	8.56e+003 (2.81e+002)	8.09e+003 ±5.39e+002	8.12e+003 (4.81e+002)
F16	4.02e+000 (4.96e-001)	3.94e+000 (6.51e-001)	3.99e+000 (6.28e-001)	<b>3.70e+000</b> (7.15e-001)	4.11e+000 (4.42e-001)	4.08e+000 ±4.67e-001	4.06e+000 (5.28e-001)
F17	8.77e+002 (1.22e+002)	1.07e+003 (9.64e+001)	1.16e+003 (1.48e+002)	<b>1.19e+002</b> (1.62e+001)	8.34e+002 (1.00e+002)	5.22e+002 ±4.72e+001	5.08e+002 (1.13e+002)
F18	8.64e+002 (1.11e+002)	1.03e+003 (9.35e+001)	1.25e+003 (1.79e+002)	5.70e+002 (7.95e+001)	8.71e+002 (7.47e+001)	<b>5.12e+002</b> ±4.92e+001	5.89e+002 (1.13e+002)
F19	8.30e+004 (5.24e+004)	1.94e+005 (1.11e+005)	8.94e+005 (5.17e+005)	3.64e+004 (8.30e+004)	8.99e+004 (4.26e+004)	1.32e+004 ±7.03e+003	<b>7.99e+003</b> (6.60e+003)
F20	<b>1.50e+001</b> (1.67e-002)	<b>1.50e+001</b> (6.24e-007)	<b>1.50e+001</b> (2.93e-004)	<b>1.50e+001</b> (6.39e-002)	<b>1.50e+001</b> (2.37e-014)	<b>1.50e+001</b> ±1.13e-002	<b>1.50e+001</b> (2.59e-002)
F21	2.63e+003 (1.61e+002)	3.06e+003 (2.44e+002)	3.43e+003 (3.12e+002)	2.11e+003 (2.91e+002)	2.56e+003 (2.20e+002)	<b>2.01e+003</b> ±1.43e+002	2.11e+003 (1.53e+002)
F22	7.78e+003 (3.87e+002)	9.07e+003 (3.39e+002)	7.98e+003 (4.64e+002)	<b>9.63e+002</b> (2.19e+002)	8.72e+003 (3.58e+002)	8.66e+003 ±5.05e+002	6.94e+003 (4.07e+002)
F23	8.95e+003 (4.45e+002)	9.34e+003 (2.71e+002)	8.99e+003 (3.47e+002)	<b>8.17e+003</b> (8.84e+002)	9.13e+003 (3.93e+002)	8.69e+003 ±6.43e+002	8.84e+003 (3.99e+002)
F24	3.29e+002 (3.39e+000)	3.46e+002 (6.27e+000)	3.35e+002 (7.79e+000)	3.27e+002 (8.57e+000)	<b>3.12e+002</b> (3.25e+000)	3.25e+002 ±7.48e+000	3.18e+002 (6.91e+000)
F25	3.53e+002 (8.37e+000)	3.71e+002 (6.37e+000)	3.23e+002 (4.35e+000)	3.55e+002 (1.01e+001)	<b>3.10e+002</b> (4.55e+000)	3.64e+002 ±6.49e+000	3.48e+002 (5.01e+000)
F26	2.92e+002 (4.92e+001)	3.18e+002 (4.11e+001)	3.47e+002 (5.67e+001)	3.98e+002 (5.03e+001)	3.51e+002 (6.41e+001)	2.69e+002 ±4.01e+001	<b>2.45e+002</b> (2.03e+001)
F27	1.47e+003 (4.58e+001)	1.52e+003 (3.38e+001)	1.48e+003 (4.06e+001)	1.50e+003 (8.47e+001)	1.42e+003 (2.75e+001)	1.42e+003 ±5.80e+001	<b>1.41e+003</b> (4.03e+001)
F28	4.34e+003 (4.46e+002)	4.92e+003 (3.55e+002)	6.51e+003 (9.15e+002)	4.98e+003 (1.58e+003)	3.91e+003 (3.68e+002)	3.70e+003 ±3.89e+002	<b>3.60e+003</b> (5.13e+002)

“Mean” and “STD.” show the average and standard deviation of the test function values, respectively. Boldface represents the best result among the compared algorithms

**Table 5** Counting the mean performances result of hDEBSA, is better than, less than or equal to the other compared algorithms (CoDE, DE/rand/2/bin, CLPSO, CPSO-H, FI-PS, DE-PSO, hDEBSA) on CEC2013 real world benchmark functions with 30 dimensions

Algorithms	Better than (hDEBSA>Algorithms)	Less than (hDEBSA<Algorithms)	Equal to (hDEBSA=Algorithms)
CoDE	25	1	2
DE/rand/2/bin	26	1	1
CLPSO	23	3	2
CPSO-H	17	9	2
FI-PS	22	4	2
DE-PSO	18	9	1

overall rank of all the algorithms using the mean results, where the Bonferroni-Dunn approach is taken as a post hoc procedure.

Table 1 shows the performance results of PSO [15], DE [17], ABC [1], BSA [4], ABSA [31], and hDEBSA on the twenty-eight benchmark functions of the CEC 2013 special session on real-parameter optimization problems with dimension 30. Table 2 shows the number of test functions where the performance of hDEBSA is better than, worse than, or similar to the compared algorithms. From Table 2, it can be seen that hDEBSA performs better than PSO on twenty-three test functions; DE on twenty-three test functions; ABC on eighteen test functions; BSA on twenty-three test functions, and ABSA on twenty-four test functions. Table 3 shows the ranks obtained by the Friedman rank test with respect to the mean performance of all algorithms for each test functions. From Table 3, it is clear that the rank of hDEBSA is the lowest. Therefore, it can be claimed that hDEBSA performs better than the compared algorithms. Several of the convergence graphs of hDEBSA are shown in Fig. 2.

Table 4 shows the performance results of CoDE [30], DE/rand/2/bin [18], CLPSO [32], CPSO-H [33], FI-PS [34], DE-PSO [19], and hDEBSA on the twenty-eight benchmark functions of the CEC 2013 special session on real-parameter optimization problems with a dimension of 30. Table 5 shows the number of test functions where the performance of hDEBSA is better than, worse than, or similar to the compared algorithms. From Table 5, it is seen that hDEBSA performs better than CoDE on twenty-five test functions; DE/rand/2/bin on twenty-six test functions; CLPSO on twenty-three test functions; CPSO-H on seventeen test functions; FI-PS on twenty-two test functions, and DE-PSO on eighteen test functions. Table 6 shows the ranks obtained by the Friedman rank test with respect to the mean performance of all algorithms of each test functions. From Table 6, it is clear that the rank of hDEBSA is the lowest compared with that of the other algorithms. Thus, it can be claimed that hDEBSA performs better than the other algorithms.

Table 7 shows the performance results of CoDE [30], EPSDE [29], DE/rand/2/bin [18], CLPSO [32], CPSO-H

[33], FI-PS [34], and hDEBSA on twenty-eight benchmark functions of the CEC 2013 special session on real-parameter optimization problems with a dimension of 50. Table 8 shows the number of occasions where the performance of hDEBSA is better than, worse than, or similar to the other algorithms. From Table 8, it is seen that hDEBSA performs better than CoDE on twenty-six test functions; EPSDE on fourteen test functions; DE/rand/2/bin on twenty-six test functions; CLPSO on twenty five-test functions; CPSO-H on nineteen test functions, and FI-PS on twenty-four test functions. Table 9 shows the ranks obtained by the Friedman rank test with respect to the mean performance of all algorithms for each test function and observed that the rank of hDEBSA is the lowest compared with that of the compared algorithms. Hence, it can be said that the performance of hDEBSA is better than that of the other algorithms.

Table 10 compares the performance results obtained by IBSA [38], MPEDE [39], I-SOS [40], SOS-ABF1 [41], SOS-ABF2 [41], SOS-ABF1&2 [41], and hDEBSA on twenty-eight CEC2013 test functions with dimension 50. Table 11 shows the number of occasions where the mean performance of hDEBSA is better than, worse than, or similar to the other algorithms. From Table 11, it can be observed that the performance of hDEBSA is better than

**Table 6** Ranks obtained by Friedman's test in which Bonferroni-Dunn's procedure was used as a post hoc procedure of CoDE, DE/rand/2/bin, CLPSO, CPSO-H, FI-PS, DE-PSO, hDEBSA with respect to the mean performance on CEC2013 test functions with dimension 30

Algorithms	Ranks
CoDE	4.07
DE/rand/2/bin	5.71
CLPSO	5.59
CPSO-H	3.41
FI-PS	4.20
DE-PSO	2.80
hDEBSA	2.21

The Friedman's test was conducted based on the SPSS software



**Table 7** Performance results of CoDE, EPSDE, jDE, CLPSO, CPSO-H, FI-PS and hDEBSA at dimension(D) 50 after reaching 5000 FEs of CEC2013 test functions over 30 runs with 50 population size

Function	CoDE [30]	EPSDE [29]	DE/rand/2/bin [18]	CLPSO [32]	CPSO-H [33]	FI-PS [34]	hDEBSA
	Mean (STD.)	Mean (STD.)	Mean (STD.)	Mean (STD.)	Mean (STD.)	Mean (STD.)	Mean (STD.)
F1	2.82e+004 ±5.41e+003	2.36e+003 ±1.42e+003	6.17e+004 ±7.01e+003	5.77e+004 ±8.17e+003	<b>3.48e+002</b> ±8.94e+001	3.72e+004 ±3.56e+003	8.35e+003 ±2.80e+003
F2	6.98e+008 ±1.26e+008	4.93e+008 ±2.67e+008	1.38e+009 ±2.01e+008	1.60e+009 ±3.83e+008	6.30e+008 ±2.09e+008	9.90e+008 ±1.16e+008	<b>4.20e+008</b> ±1.09e+008
F3	5.07e+011 ±5.48e+011	2.61e+011 ±1.25e+011	2.05e+013 ±2.25e+013	2.48e+014 ±3.59e+014	8.36e+013 ±2.93e+014	3.23e+011 ±6.95e+010	<b>1.23e+011</b> ±4.42e+010
F4	1.36e+005 ±1.96e+004	3.63e+005 ±1.08e+005	1.77e+005 ±1.88e+004	2.14e+005 ±2.39e+004	2.57e+005 ±5.25e+004	2.30e+005 ±2.58e+004	<b>1.59e+005</b> ±2.05e+004
F5	6.76e+003 ±1.27e+003	1.01e+003 ±4.08e+002	8.71e+003 ±1.27e+003	1.69e+004 ±3.19e+003	<b>1.16e+002</b> ±8.85e+001	8.41e+003 ±1.03e+003	2.12e+003 ±9.34e+002
F6	2.60e+003 ±5.43e+002	<b>3.25e+002</b> ±1.11e+002	6.32e+003 ±1.11e+003	5.78e+003 ±1.39e+003	1.90e+003 ±5.35e+002	3.18e+003 ±5.35e+002	7.72e+002 ±1.95e+002
F7	5.22e+002 ±2.64e+002	<b>3.16e+002</b> ±1.07e+002	2.90e+003 ±1.83e+003	5.98e+003 ±5.10e+003	9.47e+003 ±2.04e+004	3.56e+002 ±4.40e+001	2.42e+002 ±5.28e+001
F8	<b>2.13e+001</b> ±4.06e+002	<b>2.13e+001</b> ±4.80e+002	2.13e+001 ±4.82e+002	<b>2.13e+001</b> ±4.06e+002	<b>2.13e+001</b> ±5.03e+002	<b>2.13e+001</b> ±3.50e+002	<b>2.13e+001</b> ±2.92e+002
F9	7.77e+001 ±1.68e+000	7.85e+001 ±2.16e+000	7.90e+001 ±1.68e+000	7.46e+001 ±2.30e+000	7.42e+001 ±3.94e+000	7.89e+001 ±1.82e+000	<b>7.38e+001</b> ±1.97e+000
F10	4.86e+003 ±6.41e+002	8.73e+002 ±4.53e+002	8.86e+003 ±1.33e+003	9.65e+003 ±1.16e+003	5.21e+003 ±1.38e+003	6.11e+003 ±6.00e+002	<b>2.59e+003</b> ±7.29e+002
F11	8.00e+002 ±5.73e+001	<b>4.67e+002</b> ±3.28e+001	1.09e+003 ±7.13e+001	1.11e+003 ±1.19e+002	8.02e+001 ±1.22e+001	9.35e+002 ±3.95e+001	4.98e+002 ±4.38e+001
F12	8.51e+002 ±7.18e+001	<b>5.20e+002</b> ±3.02e+001	1.17e+003 ±7.11e+001	1.18e+003 ±8.39e+001	1.25e+003 ±2.15e+002	9.06e+002 ±5.26e+001	6.67e+002 ±5.47e+001
F13	8.68e+002 ±8.27e+001	<b>5.24e+002</b> ±3.61e+001	1.18e+003 ±8.23e+001	1.18e+003 ±9.13e+001	1.19e+003 ±1.63e+002	9.17e+002 ±5.38e+001	6.75e+002 ±4.66e+001
F14	1.27e+004 ±5.27e+002	1.23e+004 ±5.90e+002	1.50e+004 ±4.72e+002	1.31e+004 ±4.73e+002	<b>9.02e+002</b> ±1.73e+002	1.47e+004 ±4.54e+002	1.11e+004 ±2.95e+002
F15	1.56e+004 ±4.83e+002	1.60e+004 ±5.94e+002	1.56e+004 ±4.93e+002	1.53e+004 ±4.83e+002	<b>1.32e+004</b> ±9.44e+002	1.57e+004 ±4.95e+002	1.49e+004 ±6.47e+002
F16	5.02e+000 ±3.86e+001	4.89e+000 ±4.69e+001	4.82e+000 ±4.42e+001	4.69e+000 ±5.10e+001	<b>4.44e+000</b> ±5.86e+001	4.69e+000 ±5.09e+001	4.57e+000 ±3.37e+001
F17	1.42e+003 ±1.38e+002	5.42e+002 ±4.87e+001	2.25e+003 ±1.80e+002	2.00e+003 ±1.91e+002	<b>2.03e+002</b> ±1.87e+001	1.71e+003 ±8.58e+001	6.95e+002 ±1.03e+002
F18	1.40e+003 ±2.21e+002	<b>5.69e+002</b> ±3.89e+001	2.25e+003 ±1.98e+002	1.96e+003 ±2.18e+002	1.17e+003 ±1.60e+002	1.69e+003 ±1.23e+002	8.37e+002 ±1.47e+002
F19	1.09e+005 ±4.97e+004	1.54e+004 ±4.31e+004	8.73e+005 ±3.05e+005	1.41e+006 ±5.35e+005	1.36e+004 ±2.54e+004	4.48e+005 ±1.63e+005	<b>7.33e+003</b> ±1.15e+004
F20	<b>2.50e+001</b> ±3.50e+002	<b>2.50e+001</b> ±1.45e+001	2.50e+001 ±9.95e+002	<b>2.50e+001</b> ±7.98e+003	<b>2.50e+001</b> ±1.01e+001	<b>2.50e+001</b> ±4.96e+011	<b>2.50e+001</b> ±1.72e+002
F21	4.86e+003 ±3.91e+002	<b>2.45e+003</b> ±6.12e+002	6.43e+003 ±2.76e+002	5.99e+003 ±4.35e+002	3.87e+003 ±6.62e+002	5.06e+003 ±1.79e+002	3.69e+003 ±1.40e+002
F22	1.40e+004 ±6.26e+002	1.31e+004 ±6.83e+002	1.61e+004 ±4.78e+002	1.43e+004 ±6.80e+002	1.71e+003 ±5.39e+002	1.55e+004 ±5.41e+002	<b>1.25e+004</b> ±5.17e+002
F23	1.66e+004 ±2.78e+002	1.64e+004 ±4.30e+002	1.66e+004 ±4.31e+002	1.65e+004 ±4.63e+002	<b>1.51e+004</b> ±8.70e+002	1.61e+004 ±3.83e+002	1.61e+004 ±4.05e+002
F24	4.22e+002 ±7.08e+000	4.01e+002 ±3.52e+000	4.73e+002 ±1.24e+001	4.21e+002 ±7.01e+000	4.31e+002 ±1.61e+001	3.99e+002 ±4.91e+000	<b>3.92e+002</b> ±1.19e+001
F25	4.75e+002 ±1.07e+001	<b>3.96e+002</b> ±6.04e+000	5.31e+002 ±9.88e+000	4.10e+002 ±5.74e+000	4.84e+002 ±2.14e+001	3.98e+002 ±4.73e+000	4.66e+002 ±9.46e+000
F26	4.46e+002 ±5.62e+001	4.99e+002 ±4.65e+000	4.88e+002 ±3.95e+001	4.62e+002 ±5.58e+001	4.84e+002 ±6.60e+001	4.11e+002 ±7.28e+001	<b>2.90e+002</b> ±3.29e+001
F27	2.40e+003 ±5.13e+001	2.27e+003 ±5.65e+001	2.53e+003 ±8.45e+001	2.38e+003 ±5.68e+001	2.42e+003 ±1.21e+002	2.29e+003 ±5.34e+001	<b>2.24e+003</b> ±6.26e+001
F28	6.04e+003 ±7.60e+002	<b>2.35e+003</b> ±1.21e+003	8.40e+003 ±5.25e+002	9.19e+003 ±9.61e+002	6.56e+003 ±1.37e+003	6.07e+003 ±5.24e+002	4.56e+003 ±6.68e+002

“Mean ” and “STD.” show the average and standard deviation of the test function values, respectively. Boldface represents the best result among the compared algorithms

**Table 8** Counting the mean performances result of hDEBSA, is better than, less than or equal to the other compared algorithms (CoDE, EPSDE, DE/rand/2/bin, CLPSO, CPSO-H, FI-PS, hDEBSA) on CEC2013 real world benchmark functions with 50 dimensions

Algorithms	Better than (hDEBSA>Algorithms)	Less than (hDEBSA<Algorithms)	Equal to (hDEBSA=Algorithms)
CoDE	26	0	2
EPSDE	14	12	2
DE/rand/2/bin	26	0	2
CLPSO	25	1	2
CPSO-H	19	7	2
FI-PS	24	1	3

IBSA on nineteen test functions, MPEDE on twenty-four test functions, I-SOS on twenty-three test functions, SOS-ABF1 on fourteen test functions, SOS-ABF2 on fifteen test functions, and SOS-ABF1&2 on nineteen test functions. Table 12 shows the ranks obtained by Friedman rank test with respect to the mean performances, where it is shown that the rank of hDEBSA is the lowest. Hence, it may be concluded that the performance of hDEBSA is better than that of the other algorithms.

**5 Formulation of the real-world optimization problems**

In this section, the formulation of four real-world problems and the performance results of these four optimization problems are discussed.

**5.1 Problem formulation**

To apply the hDEBSA on real-world optimization problems, two real-world problems, namely the Gas Transmission Compressor Design problem and Optimal Capacity of Gas Production facilities, are taken from [35] and another two problems, the Frequency Modulation Sounds Parameter

**Table 9** Ranks obtained by Friedman’s test in which Bonferroni-Dunn’s procedure was used as a post hoc procedure of CoDE, EPSDE, DE/rand/2/bin, CLPSO, CPSO-H, FI-PS, DE-PSO, hDEBSA with respect to the mean performance on CEC2013 test functions with dimension 50

Algorithms	Ranks
CoDE	4.04
EPSDE	2.86
DE/rand/2/bin	5.95
CLPSO	5.29
CPSO-H	3.50
FI-PSO	4.29
hDEBSA	2.09

The Friedman’s test was conducted based on the SPSS software

Identification problem and the Spread Spectrum Radar Polyphase Code Design, problem are taken from [36]. The formulation of these real-world optimization problems are presented below:

**P1. Gas transmission compressor design problem**

$$\begin{aligned}
 \text{Minf}(x) = & 8.61 \times 10^5 \times x_1^{\frac{1}{2}} \times x_2 \times x_3^{\frac{-2}{3}} \times (x_2^2 - 1)^{-\frac{1}{2}} \\
 & + 3.69 \times 10^4 \times x_3 + 7.72 \times 10^8 \times x_1^{-1} \\
 & \times x_2^{0.219} - 765.43 \times 10^6 \times x_1^{-1} \quad (6)
 \end{aligned}$$

Such that,  $10 \leq x_1 \leq 55, 1.1 \leq x_2 \leq 2, 10 \leq x_3 \leq 40;$

**P2. Optimal capacity of gas production facilities**

$$\begin{aligned}
 \text{Minf}(x) = & 61.8 + 5.72 \times x_1 \times 0.2623 \\
 & \times \left[ (40 - x_1) \times \ln\left(\frac{x_2}{200}\right) \right]^{-0.85} + 0.087 \\
 & \times (40 - x_1) \times \ln\left(\frac{x_2}{200}\right) + 700.23 \times x_2^{-0.75} \quad (7)
 \end{aligned}$$

Such that,  $x_1 \geq 17.5, x_2 \geq 200, 17.5 \leq x_1 \leq 40, 300 \leq x_2 \leq 600;$

**P3. Frequency modulation sounds parameter identification problem:**

In the modern sound system, the frequency-modulated (FM) sound wave synthesis has an important role to generate the target sound in the FM synthesizer. This optimization problem has six parameters, i.e., six dimensions given by  $X = \{a1, \omega1, a2, \omega2, a3, \omega3\}$ . The objective is to determine the optimum values of these parameters in such a way that the sound that is generated is similar to that of the target sound. The sound waves, i.e., the estimated sound and the target sound waves using these parameters, are given by:

$$y(t) = a_1 \sin(\omega_1 t \theta + a_2 \sin(\omega_2 t \theta + a_3 \sin(\omega_3 t \theta))) \quad (8)$$

$$\begin{aligned}
 y_0(t) = & 1.0 \times \sin(5.0 \times t \theta + 1.5 \times \sin(4.8 \times t \theta \\
 & + 2.0 \sin(4.9 \times t \theta))) \quad (9)
 \end{aligned}$$

**Table 10** Performance results of IBSA, MPEDE, I-SOS, SOS-ABF1, SOS-ABF2, SOS-ABF1&2 and hDEBSA at dimension (D) 50 after reaching 5000 FEs of CEC2013 test functions over 30 runs with 50 population size

Function	IBSA [39] Mean (STD.)	MPEDE [37] Mean (STD.)	I-SOS [40] Mean (STD.)	SOS-ABF1 [41] Mean (STD.)	SOS-ABF2 [41] Mean (STD.)	SOS-ABF1 & 2 [41] Mean (STD.)	hDEBSA Mean (STD.)
F1	9.49e+003 ±1.56e+003	2.33e+004 ±2.70e+003	4.20e+004 ±6.38e+003	7.65e+003 ±1.88e+003	7.20e+003 ±1.54e+003	9.40e+003 ±9.96e+002	8.35e+003 ±2.80e+003
F2	4.38e+008 ±8.53e+007	5.76e+008 ±9.50e+007	3.41e+008 ±1.01e+008	2.38e+008 ±4.39e+007	2.25e+008 ±5.40e+007	8.01e+008 ±1.75e+008	4.20e+008 ±1.09e+008
F3	1.83e+011 ±1.57e+011	1.76e+011 ±6.91e+010	3.01e+011 ±2.73e+011	1.05e+011 ±6.13e+010	8.36e+010 ±3.47e+010	9.73e+013 ±1.73e+014	1.23e+011 ±4.42e+010
F4	1.53e+005 ±1.97e+004	1.42e+005 ±1.69e+004	1.22e+005 ±2.05e+004	1.59e+005 ±2.95e+004	1.62e+005 ±2.24e+004	2.19e+005 ±2.44e+004	1.59e+005 ±2.05e+004
F5	2.07e+003 ±4.62e+002	6.32e+003 ±1.41e+003	6.57e+003 ±1.97e+003	1.69e+003 ±3.61e+002	1.57e+003 ±3.10e+002	5.23e+003 ±1.28e+003	2.12e+003 ±9.34e+002
F6	9.13e+002 ±1.43e+002	1.69e+003 ±3.16e+002	2.84e+003 ±7.74e+002	7.81e+002 ±1.14e+002	7.75e+002 ±1.33e+002	2.02e+003 ±4.49e+002	7.72e+002 ±1.95e+002
F7	2.47e+002 ±4.35e+001	3.96e+002 ±1.22e+002	3.94e+002 ±4.05e+002	1.97e+002 ±3.21e+001	1.92e+002 ±2.96e+001	2.76e+003 ±2.84e+003	2.42e+002 ±5.28e+001
F8	2.13e+001 ±4.71e-002	2.13e+001 ±4.55e-002	2.13e+001 ±3.71e-002	2.13e+001 ±4.23e-002	2.13e+001 ±4.84e-002	2.13e+001 ±3.46e-002	2.13e+001 ±2.92e-002
F9	7.39e+001 ±1.76e+000	7.85e+001 ±1.65e+000	7.80e+001 ±2.73e+000	7.70e+001 ±2.65e+000	7.63e+001 ±2.26e+000	7.43e+001 ±3.88e+000	7.38e+001 ±1.97e+000
F10	2.72e+003 ±3.99e+002	3.78e+003 ±5.86e+002	4.98e+003 ±9.81e+002	1.48e+003 ±2.47e+002	1.38e+003 ±2.09e+002	2.38e+003 ±2.95e+002	2.59e+003 ±7.29e+002
F11	5.18e+002 ±3.48e+001	7.35e+002 ±5.02e+001	8.90e+002 ±7.15e+001	6.35e+002 ±4.70e+001	6.70e+002 ±7.58e+001	6.31e+002 ±5.69e+001	4.98e+002 ±4.38e+001
F12	6.83e+002 ±4.77e+001	7.58e+002 ±3.34e+001	9.39e+002 ±7.46e+001	6.71e+002 ±5.56e+001	6.69e+002 ±5.11e+001	6.73e+002 ±6.50e+001	6.67e+002 ±5.47e+001
F13	7.10e+002 ±5.51e+001	7.66e+002 ±4.99e+001	9.71e+002 ±1.10e+002	6.62e+002 ±5.74e+001	6.77e+002 ±4.51e+001	6.63e+002 ±5.42e+001	6.75e+002 ±4.66e+001
F14	1.10e+004 ±4.87e+002	1.42e+004 ±4.84e+002	1.42e+004 ±4.50e+002	1.38e+004 ±5.19e+002	1.36e+004 ±6.03e+002	1.37e+004 ±6.20e+002	1.11e+004 ±2.95e+002
F15	1.52e+004 ±4.36e+002	1.55e+004 ±4.23e+002	1.57e+004 ±4.96e+002	1.56e+004 ±4.01e+002	1.57e+004 ±5.70e+002	1.54e+004 ±5.62e+002	1.49e+004 ±6.47e+002
F16	4.55e+000 ±4.38e-001	4.78e+000 ±4.52e-001	4.83e+000 ±5.74e-001	4.71e+000 ±5.75e-001	4.78e+000 ±4.32e-001	4.77e+000 ±5.83e-001	4.57e+000 ±3.37e-001
F17	7.82e+002 ±6.26e+001	1.19e+003 ±9.86e+001	1.23e+003 ±7.37e+001	7.83e+002 ±7.92e+001	7.98e+002 ±6.31e+001	7.91e+002 ±7.12e+001	6.95e+002 ±1.03e+002
F18	8.89e+002 ±6.48e+001	1.18e+003 ±1.06e+002	1.22e+003 ±9.19e+001	8.12e+002 ±7.29e+001	8.12e+002 ±7.69e+001	8.21e+002 ±6.15e+001	8.37e+002 ±1.47e+002
F19	8.58e+003 ±4.80e+003	5.40e+004 ±2.25e+004	9.43e+004 ±4.86e+004	2.06e+004 ±1.14e+004	1.27e+004 ±8.33e+003	6.85e+005 ±2.83e+005	7.33e+003 ±1.15e+004
F20	2.49e+001 ±9.24e-002	2.50e+001 ±5.87e-002	2.47e+001 ±2.42e-001	2.44e+001 ±4.29e-001	2.43e+001 ±4.10e-001	2.42e+001 ±5.04e-001	2.50e+001 ±1.72e-002
F21	3.87e+003 ±2.84e+002	4.54e+003 ±2.19e+002	4.16e+003 ±1.10e+002	2.82e+003 ±4.11e+002	2.60e+003 ±4.13e+002	2.49e+003 ±4.62e+002	3.69e+003 ±1.40e+002
F22	1.23e+004 ±6.18e+002	1.54e+004 ±4.04e+002	1.53e+004 ±5.18e+002	1.46e+004 ±6.43e+002	1.48e+004 ±6.39e+002	1.48e+004 ±6.27e+002	1.25e+004 ±5.17e+002
F23	1.60e+004 ±4.11e+002	1.65e+004 ±3.84e+002	1.68e+004 ±3.13e+002	1.62e+004 ±6.16e+002	1.65e+004 ±3.78e+002	1.63e+004 ±5.76e+002	1.61e+004 ±4.05e+002
F24	3.92e+002 ±7.88e+000	4.08e+002 ±8.60e+000	4.09e+002 ±7.13e+000	3.94e+002 ±1.50e+001	3.92e+002 ±1.04e+001	3.95e+002 ±1.40e+001	3.92e+002 ±1.19e+001
F25	4.63e+002 ±7.11e+000	3.99e+002 ±5.39e+000	4.01e+002 ±4.86e+000	4.25e+002 ±1.62e+001	4.25e+002 ±1.46e+001	4.20e+002 ±1.40e+001	4.66e+002 ±9.46e+000
F26	3.23e+002 ±5.96e+001	4.43e+002 ±6.50e+001	4.69e+002 ±6.41e+001	4.61e+002 ±7.06e+001	4.24e+002 ±1.01e+002	4.47e+002 ±8.04e+001	2.90e+002 ±3.29e+001
F27	2.25e+003 ±5.05e+001	2.34e+003 ±6.74e+001	2.46e+003 ±9.42e+001	2.29e+003 ±1.21e+002	2.29e+003 ±1.03e+002	2.25e+003 ±1.27e+002	2.24e+003 ±6.26e+001
F28	4.62e+003 ±6.34e+002	5.43e+003 ±7.11e+002	6.89e+003 ±4.98e+002	2.57e+003 ±9.73e+002	2.58e+003 ±1.14e+003	2.52e+003 ±8.13e+002	4.56e+003 ±6.68e+002

“Mean” and “STD.” show the average and standard deviation of the test function values, respectively

**Table 11** Counting the mean performances result of hDEBSA, is better than, less than or equal to the other compared algorithms (IBSA, MPEDE, I-SOS, SOS-ABF1, SOS-ABF2, SOS-ABF1&2 and hDEBSA) on CEC2013 real world benchmark functions with 50 dimensions

Algorithms	Better than (hDEBSA>Algorithms)	Less than (hDEBSA<Algorithms)	Equal to (hDEBSA=Algorithms)
IBSA	19	7	2
MPEDE	24	2	2
I-SOS	23	4	1
SOS-ABF1	14	12	2
SOS-ABF2	15	11	2
SOS-ABF1&2	19	8	1

**Table 12** Ranks obtained by Friedman’s test in which Bonferroni–Dunn’s procedure was used as a post hoc procedure of IBSA, MPEDE, I-SOS, SOS-ABF1, SOS-ABF2, SOS-ABF1&2 and hDEBSA with respect to the mean performance on CEC2013 test functions with dimension 50

Algorithms	Ranks
IBSA	3.38
MPEDE	5.43
I-SOS	5.96
SOS-ABF1	3.25
SOS-ABF2	3.16
SOS-ABF1&2	4.07
hDEBSA	2.75

The Friedman’s test was conducted based on the SPSS software

**Table 13** Performance results of P1 and P2 real world optimization problems

Item	Beightler and Phillips [35]	DE	BSA	hDEBSA
P1. Performance results of DE, BSA and hDEBSA on gas transmission compressor design				
X <sub>1</sub>	55	53.4467	<b>53.4239</b>	<b>53.4478</b>
X <sub>2</sub>	1.195	<b>1.1901</b>	<b>1.19021</b>	<b>1.1901</b>
X <sub>3</sub>	25.026	<b>24.7186</b>	<b>24.6793</b>	<b>24.7191</b>
f(X)	2.96455E+06	<b>2.96438e+006</b>	<b>2.96438e+006</b>	<b>2.96438e+006</b>
P2. Performance results of DE, BSA and hDEBSA on Optimal Capacity of Gas production facilities.				
X <sub>1</sub>	<b>17.5</b>	<b>17.5</b>	<b>17.5</b>	<b>17.5</b>
X <sub>2</sub>	465	<b>600</b>	<b>600</b>	<b>600</b>
f(X)	173.76	<b>169.8437</b>	<b>169.8437</b>	<b>169.8437</b>

Boldface represents the best result among the compared algorithms

respectively (where  $\theta = 2\pi/100$ ), where each parameter is bounded by the range  $[-6.4, 6.35]$ . The fitness function, i.e., the objective function for this optimization problem is given by

$$f(\vec{X}) = \sum_{t=0}^{100} (y(t) - y_0(t))^2 \tag{10}$$

The optimum value, i.e., the minimum value of the frequency modulation sound parameter identification optimization problem is  $f(X^*) = 0$ .

**P4. Spread spectrum radar polyphase code design problem:**

The pulse compression technique is an important technique widely used to design a radar-system. The polyphase compression code synthesis offers convenience and is easier to implement the digital processing technique. This optimization problem is a continuous min–max global optimization problem in continuous variables with numerous local optima. Based on the properties of the aperiodic auto-correlation function and the assumption of coherent radar pulse processing in the receiver, the min–max model can be defined as

$$Global \min f(X) = \max \{\varphi_1(X), \varphi_2(X), \dots, \varphi_{2m}(X)\} \tag{11}$$

**Table 14** Best, Mean and SD results of P3 and P4 real life optimization problems compared to state of the art other algorithms

Function		Best	Median	Worst	Mean	SD
P3	hDEBSA	<b>1.37e+001</b>	2.25e+001	2.43e+001	2.17e+001	2.37e+000
	DE	1.54e+001	2.16e+001	2.41e+001	<b>2.14e+001</b>	1.86e+000
	BSA	1.78e+001	2.35e+001	2.56e+001	2.29e+001	1.96e+000
P4	hDEBSA	<b>1.50e+000</b>	1.77e+000	2.12e+000	<b>1.80e+000</b>	1.47e-001
	DE	1.69e+000	2.15e+000	2.33e+000	2.10e+000	1.57e-001
	BSA	1.51e+000	1.86e+000	2.11e+000	1.84e+000	1.66e-001

Boldface represents the best result among the compared algorithms

where  $X = \{(x_1, x_2, x_3, \dots, x_D) \in R^D | 0 \leq x_j \leq 2\pi, j = 1, 2, 3, \dots, D\}$  and  $m = 2D - 1$ , with

$$\varphi_{2i-1}(X) = \sum_{j=i}^D \cos \left( \sum_{k=|2i-j-1|+1}^j X_k \right), i = 1, 2, 3, \dots, D \tag{12}$$

$$\varphi_{2i}(X) = 0.5 + \sum_{j=i+1}^D \cos \left( \sum_{k=|2i-j|+1}^j X_k \right), i = 1, 2, 3, \dots, D - 1 \tag{13}$$

$$\varphi_{m+i}(X) = -\varphi_i(X), i = 1, 2, 3, \dots, m \tag{14}$$

Here, the objective is to obtain the minimum value of the module of the largest among the samples of the auto-correlation function  $\phi$  that are related to the complex envelope of the compressed radar pulse at the optimal receiver output.

### 5.2 Result and discussion of the real-world problems

To analyze the performance of hDEBSA on four real-world problems, the algorithm was run for 30 times with 5000 fitness evaluations and 50 population sizes. Table 13 shows the performance results of Beightler and Phillips [35], DE [17], BSA [4], and hDEBSA for two real-world problems (P1 and P2). From this table, it is seen that the performance of DE, BSA, and hDEBSA are the same. The performance of hDEBSA is better than that of Beightler and Phillips [35]. Table 14 shows the performance results of DE, BSA, and hDEBSA on real-world problems P3 and P4. From this table, it is seen that for P3, the performance of hDEBSA is better than that of BSA; for P4, the performance of hDEBSA is better than that of DE and BSA.

### 6 Conclusion

In this paper, a hybrid algorithm hDEBSA is presented using two popular optimization techniques, DE and BSA. In hDEBSA, self-adaptation schemes for control parameters are suggested, in which the value of control parameters vary

automatically during the optimization process. The proposed hDEBSA is applied on twenty-eight CEC 2013 test functions, two industrial engineering design problems, and two real-world optimization problems for validation. The obtained results were compared with several standard algorithms, such as PSO, DE, ABC, BSA, and ABSA; several improved variants of DE (CoDE, EPSDE, DE/rand/2/bin); several improved variants of PSO (CLPSO, CPSO-H, FI-PS), and also one hybrid algorithm DE-PSO were also used as a comparison. The comparison results in terms of the numerical result and statistical analysis show that the proposed method is superior to the aforementioned algorithms and thus, is acceptable. Hence, the proposed method may be recommended to solve optimization problems in different branches of humanities, science, and engineering.

**Acknowledgments** The authors are extremely thankful to anonymous referees and the editor for their valuable comments towards improving the manuscript.

### References

1. Akay B, Karaboga D (2012) Artificial bee colony algorithm for large-scale problem and engineering design optimization. *J Intell Manuf* 23:1001–1014
2. Smuc T (2002) Sensitivity of differential evolution algorithm to value of control parameters. In: *Proceedings of the international conference on artificial intelligence*, pp 108–1093
3. Smuc T (2002) Improving convergence properties of the differential evolution algorithm. In: *Proceedings of MENDEL 2002, 8th international Mendel conference on soft computing*, pp 80–86
4. Civicioglu P (2013) Backtracking search optimization algorithm for numerical optimization problems. *Appl Math Comput* 219(14):8121–8144
5. Liang JJ, Qu BY, Suganthan P, Hernández-Díaz AG (2013) Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization, Technical Report 201212, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China And Technical Report, Nanyang Technological University, Singapore
6. Eberhart R, Shi Y (2001) Particle swarm optimization: Developments, Applications and resources. In: *Proceedings of the 2001 congress on evolutionary computation*, vol 81, pp 81–86
7. Gong W, Cai Z (2013) Differential evolution with ranking based mutation operators. *IEEE Trans Cybern* 43(5):2066–2081

8. Holland JH (1975) *Adaptation in natural and artificial systems*. University of Michigan Press
9. Zaharie D (2009) Influence of crossover on the behavior of differential evolution algorithms. *Appl Soft Comput* 9(3):1126–1138
10. Lampinen J, Zelinka I (2000) On stagnation of the differential evolution algorithm. In: *Proceedings of MENDEL 2000, 6th international Mendel conference on soft computing*, pp 76–83
11. Wang L, Zhong Y, Yin Y, Zhao W, Wang B, Xu Y (2015) A hybrid backtracking search optimization algorithm with differential evolution. In: *Mathematical problems in engineering*, Volume 2015, Article ID 769245, <https://doi.org/10.1155/2015/769245>
12. Gämperle R, Müller SD, Koumoutsakos P (2002) A parameter study for differential evolution. *Adv Intell Syst Fuzzy Syst Evol Comput* 10:293–298
13. Ronkkonen J, Kukkonen S, Price K (2005) Real-parameter optimization with differential evolution. *Proc IEEE CEC* 1:506–513
14. Storn R (1996) On the usage of differential evolution for function optimization. In: *Biennial conference of the North American fuzzy information processing society (NAFIPS)*. IEEE, Berkeley, pp 519–523
15. Shi Y, Eberhart R (1998) A modified particle swarm optimizer. *Evolutionary Computation Proceedings, IEEE World Congress on Computational Intelligence*
16. Zhang C, Ning J, Lu S, Ouyang D, Ding T (2009) A novel hybrid differential evolution and particle swarm optimization algorithm for unconstrained optimization. *Oper Res Lett* 37:117–122
17. Storn R, Price K (1997) Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim* 11(4):341–359
18. Zhang L, Li H, Jiao Y-C, Zhang F-S (2009) *Hybrid differential evolution and the simplified quadratic interpolation for global optimization*, Copyright is held by the author/owner(s). GEC'09. ACM, Shanghai, pp 12–14. 978-1-60558-326-6/09/06
19. Pant M, Thangaraj R (2011) DE-PSO: A new hybrid meta-heuristic for solving global optimization problems. *Math Nat Comput* 7(3):363–381
20. Omran M, Engelbrecht AP, Salman A (2008) Bare bones differential evolution. *Eur J Oper Res* 196(1):128–139
21. Gong W, Cai Z, Ling CX (2010) DE/BBO: a hybrid differential evolution with biogeography-based optimization for global numerical optimization. *Soft Comput* 15(4):645–665
22. Wen-Yi L (2010) A GA–DE hybrid evolutionary algorithm for path synthesis off our- bar linkage. *Mech Mach Theory* 45:1096–1107
23. Rao RV, Sivasani VJ (2012) *Mechanical design optimization using advanced optimization technique*. Springer, London
24. Simon D (2008) Biogeography-based optimization. *IEEE Trans Evol Comput* 12:702–713
25. Abbas H (2002) The self-adaptive pareto differential evolution algorithm. In: *Proceedings of 2002 congress on evolutionary computation*, vol 1, pp 831–836
26. Brest J, Greiner S, Boškovic B, Mernik M, Žumer V (2006) Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Trans Evol Comput* 10(5):646–657
27. Zhang J, Sanderson A (2009) JADE: Adaptive differential evolution with optional external archive. *IEEE Trans Evol Comput* 13(5):945–958
28. Qin AK, Huang VL, Suganthan P (2009) Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans Evol Comput* 13(2):398–417
29. Mallipeddi R, Suganthan P, Pan Q, Tasgetiren M (2011) Differential evolution algorithm with ensemble of parameters and mutation strategies. *Appl Soft Comput* 11(2):1679–1696
30. Wang Y, Cai Z, Zhang Q (2011) Differential evolution with composite trial vector generation strategies and control parameters. *IEEE Trans Evol Comput* 15(1):55–66
31. Duan H, Luo Q (2014) Adaptive backtracking search algorithm for induction magnetometer optimization. *IEEE Trans Magn* 50(11):6001206
32. Liang JJ, Qin AK, Suganthan P, Baskar S (2006) Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation*, 10(3)
33. Van den Bergh F, Engelbrecht AP (2004) A cooperative approach to particle swarm optimization. *IEEE Trans Evol Comput* 8:225–239
34. Mendes R, Kennedy J, Neves J (2004) The fully informed particle swarm: Simpler, may be better. *IEEE Trans Evol Comput* 8:204–210
35. Beightler CS, Phillips DT (1976) *Applied geometric programming*. Wiley, New York, p 1976
36. Das S, Suganthan P (2010) Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems. Technical Report, <http://www.ntu.edu.sg/home/EPNSugan>
37. Wu G, Mallipeddi R, Suganthan PN, Wang R, Chen H (2016) Differential evolution with multi-population based ensemble of mutation strategies. *Inf Sci* 329:329–345
38. Awad NH, Ali MZ, Suganthan PN, Reynolds RG (2017) CADE: A hybridization of cultural algorithm and differential evolution for numerical optimization. *Inf Sci* 378:215–241
39. Nama S, Saha AK, Ghosh S (2016) Improved backtracking search algorithm for pseudo dynamic active earth pressure on retaining wall supporting c-θ backfill. *Appl Soft Comput* 52:885–897
40. Nama S, Saha AK, Ghosh S (2016) Improved symbiotic organisms search algorithm for solving unconstrained function optimization. *Decis Sci Lett* 5:361–380
41. Tejani GG, Sivasani VJ, Patel VK (2016) Adaptive symbiotic organisms search (SOS) algorithm for structural design optimization. *J Comput Des Eng* 3(3):226–249
42. Chen D, Zou F, Lu R, Wang P (2017) Learning backtracking search optimisation algorithm and its application. *Inf Sci* 376:71–94
43. Lin Q, Gao L, Li X, Zhang C (2015) A hybrid backtracking search algorithm for permutation flow-shop scheduling problem. *Computers & Industrial Engineering*, <https://doi.org/10.1016/cie.2015.04.009>
44. Askarzadeh A, Coelho LdS (2014) A backtracking search algorithm combined with Burger's chaotic map for parameter estimation of PEMFC electrochemical model. *International Journal of Hydrogen Energy*, <https://doi.org/10.1016/j.ijhydene.2014.05.052>
45. Wang B, Wang L, Yin Y, Xu Y, Zhao W, Tang Y (2015) An improved neural network with random weights using backtracking search algorithm. *Neural Process Letter*, <https://doi.org/10.1007/s11063-015-9480-z>
46. Nama S, Saha AK, Ghosh S (2016) A new ensemble algorithm of differential evolution and backtracking search optimization algorithm with adaptive control parameter for function optimization. *Int J Indust Eng Comput* 7:323–338
47. Wang L, Zhong Y, Yin Y, Zhao W, Wang B, Xu Y (2015) A hybrid backtracking search optimization algorithm with differential evolution. In: *Mathematical problems in engineering*, Volume 2015, Article ID 769245, <https://doi.org/10.1155/2015/769245>
48. Guney K, Durmus A (2016) Elliptical antenna array synthesis using backtracking search optimisation algorithm. *Def Sci J* 66:272–277
49. Modiri-Delshad M, Kaboli Aghay S, Hr, Taslimi-Renani E, Abd Rahim N (2016) Backtracking search algorithm for solving economic dispatch problems with valve-point effects and multiple fuel options. *Energy* 116:637–649
50. Souza RR, Fadel Miguel L, Lopez RH, Torii AJ, Miguel LFF (2016) A backtracking search algorithm for the simultaneous size, shape and topology optimization of trusses. *Latin Amer J Solids Struct* 13:2622–2651

51. Khooban MH, Vafamand N, Liaghat A, Dragicevic T (2016) An optimal general type-2 fuzzy controller for Urban Traffic Network, ISA Transactions, <https://doi.org/10.1016/j.isatra.2016.10.011>
52. Song X, Zhang X, Zhao S, Li L (2015) Backtracking search algorithm for effective and efficient surface wave analysis. J Appl Geophys 114:19–31
53. Wang S, Li Y, Yang H (2017) Self-adaptive differential evolution algorithm with improved mutation mode, Applied Intelligence, <https://doi.org/10.1007/s10489-017-0914-3>
54. Li X, Ma S, Hu J (2017) Multi-search differential evolution algorithm, Applied Intelligence, <https://doi.org/10.1007/s10489-016-0885-9>
55. Yi W, Gao L, Li X (2015) A new differential evolution algorithm with a hybrid mutation operator and self-adapting control parameters for global optimization problems. Appl Intell 42:642–660



**Sukanta Nama** received his M.Sc. degree in Mathematics from National Institute of Technology Agartala, India in 2013. Currently, he is working as a Research Scholar at Department of Mathematics, National Institute of Technology Agartala. His current research interest area includes operation research, evolutionary computation, application of evolutionary computation in different environment. He has published 6 journal and 1 conference papers.



**Dr. Apu Kumar Saha** is working as Assistant Professor at National Institute of Technology Agartala, India. He has obtained his Ph.D. in Mathematics from National Institute of Technology Agartala. He has more than 10 years of experience of teaching the undergraduate and postgraduate level. He has published more than 40 journal and conference papers. His primary area of interest is Evolutionary Computation, Nature Inspired Heuristic Global Optimization

Techniques, Multi Criteria Decision Making, Computational Fluid Dynamics etc.