



A Bio-Inspired Multi-Population-Based Adaptive Backtracking Search Algorithm

Sukanta Nama^{1,2} · Apu Kumar Saha²

Received: 26 May 2021 / Accepted: 17 December 2021 / Published online: 30 January 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

Backtracking search algorithm (BSA) is a nature-based optimization technique extensively used to solve various real-world global optimization problems for the past few years. The present work aims to introduce an improved BSA (ImBSA) based on a multi-population approach and modified control parameter settings to apprehend an ensemble of various mutation strategies. In the proposed ImBSA, a new mutation strategy is suggested to enhance the algorithm's performance. Also, for all mutation strategies, the control parameters are updated adaptively during the algorithm's execution. Extensive experiments have been performed on CEC2014 and CEC2017 single-objective benchmark functions, and the results are compared with several state-of-the-art algorithms, improved BSA variants, efficient differential evolution (DE) variants, particle swarm optimization (PSO) variants, and some other hybrid variants. The nonparametric Friedman rank test has been conducted to examine the efficiency of the proposed algorithm statistically. Moreover, six real-world engineering design problems have been solved to examine the problem-solving ability of ImBSA. The experimental results, statistical analysis, convergence graphs, complexity analysis, and the results of real-world applications confirm the superior performance of the suggested ImBSA.

Keywords Backtracking search algorithm · Mutation strategies · Multi-population strategy · CEC2014 · CEC2017

Introduction

To solve the nonlinear complex optimization problems, the backtracking search algorithm (BSA) [1] is one of the most efficient and powerful tools because of the simplicity and user-friendly approach of the algorithm. Due to these advantages, numerous attempts have been made towards further improving the BSA, and at the same time, these improved versions have been applied to solve various optimization problems in the past few years [2–9]. For example, based on self-adaptive strategy design [10–13] and hybridization mechanism [2, 3, 11, 14], many variants of BSA have been studied by several authors in the recent past. These improved and hybridized variants of BSA have been applied to obtain the optimum solution of various problems from diverse

fields, such as parameters estimation of photovoltaic models [11], identifying soil parameters [15], job-shop scheduling [9], and fuzzy multiproduct multistage scheduling problem [16].

On the other hand, other than BSA, many biologically inspired algorithms have been used in numerous real-life applications. Due to the advantages of these algorithms to solve various types of optimization problems which are otherwise very hard to solve by the conventional methods, many researchers have introduced a variety of bio-inspired algorithms. Together with their variants, those algorithms have been used to solve numerous problems from both academia and industry. For example, Saha et al. [17] applied the HSOS algorithm on the pseudo-dynamic-bearing capacity of shallow strip footing; the critical failure mechanisms were determined via optimization with a genetic algorithm (GA). Different bio-inspired algorithms have been used to optimize the factor of safety and other related parameters of slope stability analysis in [18]. The weight and cost of the cantilever retaining wall have been minimized using a hybrid algorithm, namely, h-BOASOS in [19]. Sharma et al. proposed MPBOA [20], appending mutualism and commensalism

✉ Apu Kumar Saha
apusaha_nita@yahoo.co.in

¹ Department of Mathematics, MBB University, Agartala, Tripura, India

² Department of Mathematics, National Institute of Technology Agartala, Agartala, Tripura, India

phase in BOA, and used it to segment images using multi-level thresholding strategy. Chakraborty et al. [21] updated the whale optimization algorithm (WOA), modifying the parameters, increasing the exploration ability with random initialization of solution during the global search phase, and reducing the population. Later, the updated method mWOAPR was used to segment COVID-19 X-ray images. Mohammadzadeh et al. [22] improved the grey wolf optimization (GWO) algorithm with the hill-climbing method and chaos theory to solve workflow scheduling in green cloud computing. Trinh et al. [23] used moth-flame optimization algorithm (MFO) to tune the fuzzy logic controllers to improve life span, reduce retransmission, and normalize packet loss percentage in wireless sensor networks. Some other modifications of bio-inspired algorithms can be found in [24–26].

One of the exciting findings in the field of nature-inspired algorithms was “No Free Lunch (NFL) theorems” [27]. Logically, it has been proven that no optimization algorithm is ideal for solving all optimization problems. Alternatively, a specific algorithm can produce promising effects on a set of optimization problems, but it can perform poorly in another set of problems. This theorem unlocks the field for further studies on these algorithms, strengthening existing methods and suggesting new algorithms.

BSA is executed through four significant steps: selection-I, mutation, crossover, and selection-II operators. It has two algorithm-specific control parameters: scaling factor (F) and mixrate parameter (M). The scaling factor controls the direction of the population in the mutation strategy, and the mixrate parameter controls the crossover process. The proper execution of BSA can be done by setting the appropriate value of these parameters [13]. The value of these parameters can be set in many ways [8]. Some parameter settings can enhance the convergence speed [12], whereas some other settings are effective for solving separable functions [28]. Besides, solving one specific optimization problem may require a parameter setting where parameters vary automatically during the execution process [4].

The works carried out on BSA indicate that the algorithm produces adequate results in solving unimodal and multimodal problems but does not always provide optimum solutions. In some cases, it tends to converge in local optima. Also, in some test functions, when dimensions are increased, the performance of the BSA decreases. Therefore, in this work, the core search technique of the traditional BSA has been updated to answer all these questions and increase the efficiency to make a better trade-off between exploration and exploitation.

Also, some of the researchers studied the “multi-swarm” and “multi-population island models” [29] concept on some of the state-of-the-art algorithms for improving the performance of those algorithms. Basically, in those works, the

initial population is partitioned into several subpopulations [17]. The core objective behind these works is to maintain the entire population’s assortment in the search space and maintain better local and global search abilities.

Motivated by the arguments mentioned above, in this work, a novel-enhanced BSA variant (ImBSA, in short) is proposed by utilizing the multi-population-based approach with multiple mutation operators. During the evolutionary process, the whole population is split into multiple subpopulations. The study’s main objective is to apprehend the automated parameter tuning approach among each mutation strategy over the populations to produce a new variant of BSA. ImBSA has applied on single-objective CEC 2014 [30] and CEC 2017 [31] test functions with dimension 50. The extensive comparative performance of ImBSA is exhibited with several state-of-the-art standard algorithms, improved BSA, efficient DE variants, PSO variants, and some other hybrid variants. Moreover, the proposed algorithm is applied to solve six real-world engineering design problems, namely frequency modulation sounds parameter identification problem, Lennard–Jones potential problem, Tersoff potential function minimization problem, speed reducer design problem, I-beam design problem, cantilever beam design problem, and speed reducer design problem. Through the experimental research, we seek to better understand the actions of the algorithm, in particular how exploratory and exploitative processes are balanced within this study.

The remaining part of the paper is arranged in the following way: a brief discussion of the basic BSA is presented in “[Overview of the Backtracking Search Algorithm](#)” section. The “[The Proposed Algorithm](#)” section offers a detailed description of ImBSA. The “[Performance Results Analysis of CEC 2014 and CEC 2017 Test Functions](#)” section reports the performance analysis of the proposed ImBSA on CEC 2014 and CEC 2017 test functions. The application of ImBSA to real-world optimization problems has been presented in the “[Application of ImBSA to Real-World Problem](#)” section. Finally, the “[Conclusion](#)” section reviews the influence of the present study.

Overview of the Backtracking Search Algorithm

The BSA is a population-based iterative, evolutionary algorithm that executes the process through five significant components: initialization, selection-I, mutation, crossover, and selection II.

To start with an initial population (Pop), a uniform distribution $U(lb, ub)$ is used in BSA, represented by Eq. (1).

$$Pop_{i,j} = lb + rand(0, 1) * (ub - lb) \quad (1)$$

where $i = 1, 2, 3, \dots \dots \dots, NP; j = 1, 2, 3, \dots \dots \dots, D$; lb and ub are the lower and upper bound of the optimization problem; and $rand$ is a uniformly distributed random

number. NP is the number of population; D is the dimension of the optimization problem.

Initial, historical population (OldPop) is obtained by utilizing the Eq. (2). Then selection I operator is used to redefine OldPop at the beginning of each evolutionary process through Eq. (3). Once OldPop is determined, a random shuffling function is applied to transform each individual’s position using Eq. (4).

$$OldPop_{i,j} = lb + rand(0, 1) * (ub - lb) \tag{2}$$

$$If a < b, then, OldPop = Pop; a, b \in (0, 1) \tag{3}$$

$$OldPop = permuting(OldPop) \tag{4}$$

Here, *permuting* change the position of the *OldPop*.

To generate the search-direction matrix, *OldPop* is used, which means that BSA takes partial advantage of previous experiences for generating a new trial population. BSA uses the mutation operator to generate a trial population (Mutant) during the evolutionary process using Eq. (5).

$$Mutant = Pop + F * (OldPop - Pop) \tag{5}$$

Here, (*OldPop* – *Pop*) is called the search-direction matrix, and F controls it. The value of F is defined by Eq. (6). F is the scaling factor.

$$F = 3 * rndn, \text{ where } rndn \in N(0, 1) \tag{6}$$

Here, *N*(0, 1) is the normally distributed random number.

After completing the mutation operator, the BSA crossover operator is applied to determine the trial population’s appropriate system (T). Two steps execute it. An NP*D matrix (map) of binary integer value in the crossover operator is initially calculated. The *map* is defined by Eqs. (7) and (8). Moreover, it is controlled by the mixrate parameter (M). In the second step, the trial population (T) is determined according to the Eq. (9).

$$map_{i,u(1: [M*rand*D])} = 0 | u = permutating(< 1, 2, 3, \dots, D >); \tag{7}$$

$$map_{i,randi(D)} = 1 \tag{8}$$

Here M is the mixrate parameter, and its value is 1; 'rand' is a uniformly distributed random number; randi is an integer between 1 and D.

$$T_{i,j} = \begin{cases} Mutant_{i,j} & \text{if } map_{i,j} = 1 \\ Pop_{i,j} & \text{if } map_{i,j} = 0 \end{cases} \tag{9}$$

Next, the determined trial population T is updated into the population set *Pop* using the selection II operator. In the selection II operator, the better objective function value of the individuals of T is updated to the corresponding

individual of *Pop* for producing the current population and to use in the next step through Eq. (10).

$$Pop_{i,j} = \begin{cases} T_{i,j} & \text{if } f(T_{i,j}) < f(Pop_{i,j}) \\ Pop_{i,j} & \text{Otherwise} \end{cases} \tag{10}$$

The Proposed Algorithm

This section discusses the newly suggested improved backtracking search algorithm (ImBSA) incorporating the multi-population strategy, adaptive parameter setting, and a new mutation operator in the original BSA.

Modified Mutation Operator

For an optimization algorithm’s success, the significant development of the algorithm’s local and global search capabilities is essential [36, 37]. Global exploration capability indicates that the population visited the entire search region. In contrast, local exploitation capability specifies that the population visited nearer of those potential regions of the search space where the population visited previously. BSA mutation operation uses a historical population (*OldPop*) to produce new mutants to avoid the local optimums through their exploration capability. BSA remembers the historical population until the new population is created. To find the direction of the population, *OldPop* is used; thus, BSA takes partial advantage of the previous best-performing individual (*BestPop*) for producing a new trial population. Therefore, BSA can remember the best previous population (*BestPop*). For the above reasons, we have modified the mutation operator of BSA and formulated the new mutation operator as follows:

$$Mutant = Pop + F * (OldPop - Pop) + F * (BestPop - Pop) \tag{11}$$

Multi-Population-Based Ensemble Mutation Strategy Approach

The ensemble of the BSA mutation strategy can provide an efficient algorithm for solving various complex global optimization problems. In this study, two mutation strategies, “Mutation strategy 1” and “Mutation strategy 2,” are considered for producing a mutant, each of which has respective advantage. The “Mutation strategy 1” is used to explore the whole search space of the optimization problem, and “Mutation strategy 2” is used to exploit the population nearer the potential solutions.

Mutation strategy 1:

$$V_1 = \text{Mutant} = \text{Pop} + F * (\text{OldPop} - \text{Pop}) \quad (12)$$

Mutation strategy 2:

$$V_2 = \text{Mutant} = \text{Pop} + F * (\text{OldPop} - \text{Pop}) + F * (\text{BestPop} - \text{Pop}) \quad (13)$$

To appreciate the effectiveness of the ensemble of two mutation strategies, the whole population is divided into two sub-populations in each evaluation, which may be equal or unequal in size. In this study, we have considered equal sub-populations. If Pop denotes the entire population and Pop_1 and Pop_2 represent the two sub-populations, then, $\text{Pop} = \text{Pop}_1 \cup \text{Pop}_2$. After that, a shuffle function is applied to the whole population to randomly change each population's position. Thus, during the implementation of ImBSA,

each population can exchange information by utilizing different mutation strategies in other evaluations.

Parameter Adaptation of BSA

BSA has three algorithm-specific parameters (NP, M, and F). F controls the direction of the matrix ($\text{OldPop} - \text{Pop}$). The crossover operator is controlled by the mixrate parameter M. The mixrate parameter M is adopted in the BSA crossover operator to control the population that will mutate in a new trial population [3]. However, for finding BSA's significant performance on different optimization problems, the appropriate value of control parameters, viz. scaling factor "F" and mixrate "M," has until now not been stated by some authors [4, 7, 8, 10]. Hence finding the appropriate values

Algorithm-1: Pseudo-code of ImBSA

1. **Initialization:** Initialize the value of the parameter: NP, MAX_FEs, D;
2. Evaluate population (Pop) and historical population (OldPop) according to the equations (1) and (2).
3. Evaluate value of F using equation (14)
4. Evaluate fitness of each solution in Pop .
5. Identify the best solution "BestPop."
6. FE= NP; % count the number of fitness value calculation.
7. **WHILE** (termination criteria is not met)
8. **DO**
9. Divide the population (Pop) and historical population (OldPop) into two sub-population such that $\text{Pop} = \text{Pop}_1 \cup \text{Pop}_2$; $\text{OldPop} = \text{OldPop}_1 \cup \text{OldPop}_2$;
10. $F = F_1 \cup F_2$;
- // For sub-population Pop_1
11. Determine the final form of OldPop_1 according to equations (3) and (4).
12. **For** each solution in Pop_1 , apply equation (12) to determine the mutant vector.
13. Determine mixrate parameter value using equation (15)
14. Apply BSA crossover operator using equations (7), (8) and (9) to obtain the final form of trial population.
15. FE = FE+1; % count the number of fitness value calculation.
16. Apply selection operator using equation (10).
17. Update value of F using equation (14).
18. **End for**
19. //For sub-population Pop_2
20. Determine the final form of OldPop_2 according to equations (3) and (4).
21. **For** each Pop_2 , apply equation (12) to determine the mutant vector.
22. Determine mixrate parameter value using equation (15)
23. Apply BSA crossover operator using equations (7), (8) and (9) to obtain the final form of trial population.
24. FE = FE+1; % count the number of fitness value calculation.
25. Apply selection operator using equation (10).
26. Update value of F using equation (14).
27. **End for**
28. $\text{Pop} = \text{Pop}_1 \cup \text{Pop}_2$;
29. $\text{OldPop} = \text{OldPop}_1 \cup \text{OldPop}_2$;
30. Update the best performing individual "BestPop" according to the objective function values of population Pop .
31. **END WHILE**
32. **Output:** The best objective function value and the corresponding optimum solution.

Algorithm 1 Pseudo-code of ImBSA

of F and M are a difficult task in the study of BSA. Under this consideration, self-adaption schemes for F and M have been suggested in this work. The self-adaption scheme can automatically maintain the appropriate value of these two parameters during the execution. The objective is to develop a highly tuneable algorithm based on BSA to solve a wide range of real-world optimization problems. The initial value of scaling factor F is determined within 0.45 and 2.0 for the population *Pop*. Pop^i defines the i^{th} solution in the population, and the value of “F” is determined by Eq. (14) [4].

$$F_i^{new} = \begin{cases} F_l + rand(0, 1) * (F_u - F_l), & f(T_i) > f(Pop_i) \\ F_i, & otherwise \end{cases} \tag{14}$$

If $F = 0$, then the new trial population is generated by utilizing the crossover operator only but no mutation operator. The lower and upper values of F are considered as $F_l = 0.45$ and $F_u = 2.0$. Here, $f(T_i)$ and $f(Pop_i)$ represent the value of the i^{th} trial population and the population’s objective function. From the observation of the Eq. (14), it can be said that the proper value of F is evaluated between 0.45 and 2.0. The

value of F is obtained before the mutation operator of BSA is performed.

The value of the mixrate parameter “M” for the i^{th} population is determined by Eq. (15).

$$M_i = 1 - 0.1 * (1 - r_i) \tag{15}$$

Here, $r_i \in [0, 1]$. From Eq. (15), the value of “M” is always less than 1. Thus, in the proposed ImBSA, the user does not need to guess the better values of F and M.

Boundary Control Operator

If an individual moves outside the search space, it should be returned into the search space (feasible region). To keep the individuals in the feasible region, the following rule has been introduced:

$$Pop_{i,j} = \begin{cases} lb + 0.5 * rand(0, 1) * (ub - lb), & \text{if } Pop_{i,j} < lb \\ ub - 0.5 * rand(0, 1) * (ub - lb), & \text{if } Pop_{i,j} > ub \end{cases} \tag{16}$$

Table 1 Performance results of DE, PSO, ABC, BSA, ABSA, and ImBSA at dimension (D) 50 after reaching D*1000 FEs of CEC 2014 test functions over 51 runs with 100 population size

F	DE	PSO	ABC	BSA	ABSA	ImBSA
F1	5.74e+08±1.39e+08	1.50e+07±3.21e+07	4.25e+07±1.16e+07	8.23e+07±1.65e+07	3.07e+08±6.87e+07	8.74e+07±2.02e+07
F2	7.90e+07±3.25e+07	1.56e+05±1.55e+05	4.84e+06±1.38e+07	1.26e+09±4.82e+08	2.28e+10±3.12e+09	1.01e+08±4.01e+07
F3	1.45e+04±3.61e+03	6.90e+04±1.00e+04	1.04e+04±3.64e+03	1.95e+04±3.77e+03	1.03e+05±1.49e+04	1.87e+04±4.15e+03
F4	7.17e+02±1.81e+01	7.27e+02±1.28e+02	7.26e+02±4.92e+01	9.20e+02±6.46e+01	3.23e+03±4.75e+02	7.15e+02±4.69e+01
F5	5.22e+02±3.94e-02	5.22e+02±8.30e-02	5.22e+02±4.25e-02	5.22e+02±4.36e-02	5.22e+02±3.60e-02	5.21e+02±3.49e-02
F6	6.75e+02±1.65e+00	6.56e+02±5.12e+00	6.51e+02±2.48e+00	6.51e+02±1.64e+00	6.52e+02±2.00e+00	6.49e+02±1.48e+00
F7	7.04e+02±1.65e-01	7.00e+02±1.82e-01	7.03e+02±4.25e-01	7.10e+02±3.70e+00	9.09e+02±3.82e+01	7.02e+02±3.70e-01
F8	1.21e+03±1.54e+01	1.03e+03±2.12e+01	8.37e+02±6.20e+00	9.92e+02±1.68e+01	1.03e+03±1.94e+01	9.08e+02±9.45e+00
F9	1.32e+03±1.47e+01	1.81e+03±3.38e+01	1.22e+03±2.42e+01	1.24e+03±2.74e+01	1.35e+03±2.83e+01	1.21e+03±1.46e+01
F10	1.51e+04±3.73e+02	7.45e+03±1.16e+03	7.87e+03±1.89e+02	6.36e+03±3.15e+02	6.22e+03±3.19e+02	4.06e+03±3.30e+02
F11	1.58e+04±4.69e+02	5.59e+04±2.09e+03	7.58e+04±4.68e+02	1.08e+04±3.83e+02	1.10e+04±3.14e+02	1.07e+04±3.75e+02
F12	1.20e+03±3.94e-01	1.22e+03±7.92e-01	1.21e+03±8.74e-02	1.23e+03±1.34e-01	1.20e+03±1.41e-01	1.20e+03±1.32e-01
F13	1.33e+03±6.91e-02	1.34e+03±5.43e-02	1.32e+03±5.11e-02	1.35e+03±4.48e-02	1.31e+03±7.61e-01	1.30e+03±5.32e-02
F14	1.41e+03±2.11e-01	1.42e+03±2.92e-02	1.43e+03±2.95e-02	1.44e+03±5.48e-02	1.46e+03±1.26e+01	1.40e+03±3.32e-02
F15	1.55e+03±3.13e+00	1.57e+03±1.34e+01	1.65e+03±8.89e+00	1.58e+03±2.76e+01	3.91e+05±2.01e+05	1.54e+03±5.35e+00
F16	1.64e+03±1.49e-01	1.65e+03±6.73e-01	1.66e+03±3.65e-01	1.67e+03±2.60e-01	1.63e+03±2.73e-01	1.62e+03±3.76e-01
F17	2.17e+07±6.40e+05	2.95e+07±6.24e+06	2.08e+07±6.86e+06	1.79e+07±6.14e+06	1.97e+07±1.76e+07	1.66e+07±5.47e+06
F18	8.54e+05±3.67e+05	3.54e+06±2.37e+07	6.44e+05±6.48e+05	3.10e+06±2.99e+06	9.39e+08±2.40e+08	2.56e+04±2.76e+04
F19	2.94e+03±1.05e+00	2.59e+03±3.14e+01	2.95e+03±1.53e+01	1.99e+03±5.43e+00	2.14e+03±3.91e+01	1.98e+03±1.08e+01
F20	3.20e+04±2.02e+03	2.76e+04±8.78e+03	3.30e+04±8.95e+03	2.94e+04±6.01e+03	2.97e+04±2.88e+04	2.07e+04±5.52e+03
F21	2.77e+05±1.11e+05	5.67e+06±9.58e+05	8.27e+06±3.62e+06	5.63e+06±1.59e+06	2.26e+07±6.23e+06	5.60e+06±1.61e+06
F22	4.48e+03±1.87e+02	3.89e+03±3.82e+02	4.57e+03±1.96e+02	3.85e+03±2.02e+02	4.47e+03±1.81e+02	3.84e+03±1.98e+02
F23	2.74e+03±1.39e+00	2.75e+03±7.96e+00	2.78e+03±8.83e+00	2.76e+03±9.68e+00	2.68e+03±4.09e+01	2.65e+03±2.63e+00
F24	2.87e+03±3.31e+00	2.89e+03±1.08e+01	2.98e+03±2.53e+00	2.69e+03±3.81e+00	2.79e+03±1.67e+01	2.68e+03±2.33e+00
F25	2.77e+03±1.11e+01	2.79e+03±6.89e+00	2.83e+03±3.03e+00	2.75e+03±3.99e+00	2.76e+03±7.19e+00	2.74e+03±3.41e+00
F26	2.72e+03±8.54e-02	2.79e+03±2.85e+01	2.73e+03±9.77e-02	2.74e+03±4.85e-02	2.71e+03±3.54e-01	2.70e+03±4.77e-02
F27	4.89e+03±4.71e+01	4.84e+03±2.55e+02	4.94e+03±3.81e+02	4.31e+03±8.72e+01	4.21e+03±4.04e+01	4.12e+03±8.99e+01
F28	5.24e+03±9.07e+00	5.66e+03±9.98e+02	5.71e+03±3.04e+02	5.72e+03±1.43e+02	5.61e+03±8.43e+01	5.17e+03±1.51e+02
F29	3.23e+05±5.51e+01	5.96e+08±8.30e+08	3.26e+06±2.24e+04	4.31e+06±1.96e+06	9.52e+06±7.96e+06	1.59e+05±6.09e+04
F30	6.00e+04±2.76e+02	4.64e+06±7.42e+06	5.85e+04±5.92e+03	8.78e+04±2.41e+04	1.41e+05±3.08e+04	5.40e+04±8.99e+03

Table 2 The number of occasions where ImBSA is superior to, inferior to, or similar to DE, PSO, ABC, BSA, and ABSA on CEC 2014 functions with dimension 50

Algorithms	Superior to (ImBSA > algorithms)	Inferior to (ImBSA < algorithms)	Equal to (ImBSA = algorithms)
DE	27	3	0
PSO	27	3	0
ABC	26	4	0
BSA	29	1	0
ABSA	29	0	1

where $i = 1, 2, 3, \dots \dots \dots, NP; j = 1, 2, 3, \dots \dots \dots, D$.

Incorporating the above modifications in the basic BSA, an improved BSA algorithm called ImBSA has been proposed to search the maximum possible region and exploit the population information. The framework of ImBSA is given in Algorithm 1.

Performance Results Analysis of CEC 2014 and CEC 2017 Test Functions

The proposed ImBSA is applied on CEC 2014 and CEC 2017 functions to justify the better performance of the algorithm and compared with various state-of-the-art algorithms. In the following subsections, we will discuss the results obtained by ImBSA and other algorithms on different test functions cited above.

Result and Discussion on CEC 2014 Test Function

The proposed ImBSA is executed on a suit of thirty test functions of the CEC 2014 special session and competition.

Fig. 1 Ranks obtained by Friedman test on CEC 2014 test function at $D=50$

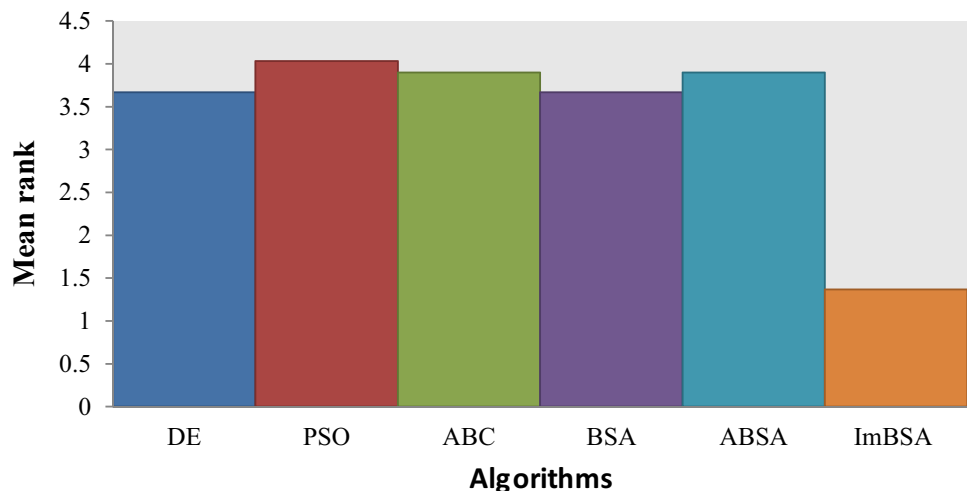


Fig.1. Ranks obtained by Friedman test on CEC2014 test function at $D = 50$

Table 3 Friedman test on CEC 2014 benchmark functions with 50 dimensions of the algorithms DE, PSO, ABC, BSA, ABSA, and ImBSA using their mean performances

Algorithms	Average rank	Variance of rank	Average rank difference w.r.t. average rank
DE	3.666666667	2.229885057	2.3
PSO	4.033333333	2.309195402	2.666666667
ABC	3.9	2.989655172	2.533333333
BSA	3.666666667	2.022988506	2.3
ABSA	3.9	2.851724138	2.533333333
ImBSA	1.366666667	0.722988506	
F-Score			3.625
Critical difference (CD)			1.239978804
F-statistics			2.567

A detailed description of these thirty functions can be seen in [30]. For a comprehensive comparison, each comparative algorithm runs 51 times over the test functions with dimension 50, population size 100, and maximum function evaluations $D*1000$. All the algorithms are run in the MATLAB R2010a platform. The lower and upper bound of each test function is considered as -100 and 100 , respectively.

Comparison with State-of-the-Art Algorithms

Computational results of thirty CEC 2014 test functions with dimension 50, obtained by BSA [1], PSO [32], ABC[33], DE [34], ABSA [13], and ImBSA, are presented in Table 1. Results obtained by ImBSA are given in boldface if they are the best among the compared algorithms. From this table, several observations and conclusions can be made. Table 2 presents the performance results of ImBSA where it is inferior to, similar to, and superior to other compared

algorithms. From Table 2, it is seen that the performance of ImBSA is superior to DE, PSO, ABC, BSA, and ABSA on 27, 27, 26, 29, and 29 test functions, respectively; inferior to DE, PSO, ABC, and BSA on 3, 3, 4, and 1 test functions respectively; and equal to ABSA on one occasion.

The Friedman rank test is conducted by utilizing each benchmark function’s mean results obtained by all algorithms. For each algorithm, the average ranks and variance of positions are obtained and presented in Table 3. At 95% confidence level, the critical value, i.e., the F-statistic value, is 2.576. Based on the average rank of all CEC 2014 test functions, the calculated Friedman statistics, i.e., F-Score values, are reported in Table 3. In Table 3, it is noticed that,

at a 95% confidence level, the F-statistics value (2.576) is less than the calculated F-Score value (3.625). Hence, the null hypothesis is rejected, and it indicated that the average ranks obtained by ImBSA utilizing the average performance of all the benchmark test functions are statistically significant.

Next, we apply a pairwise post hoc Bonferroni-Dunn test [35] to identify the significant performance of the proposed ImBSA compared to other existing algorithms. Bonferroni-Dunn test states that “if the average ranks difference is superior to the critical difference (CD) value, then the algorithm is statistically significant.” From the observation of Table 3, it is seen that, at a 95% level of

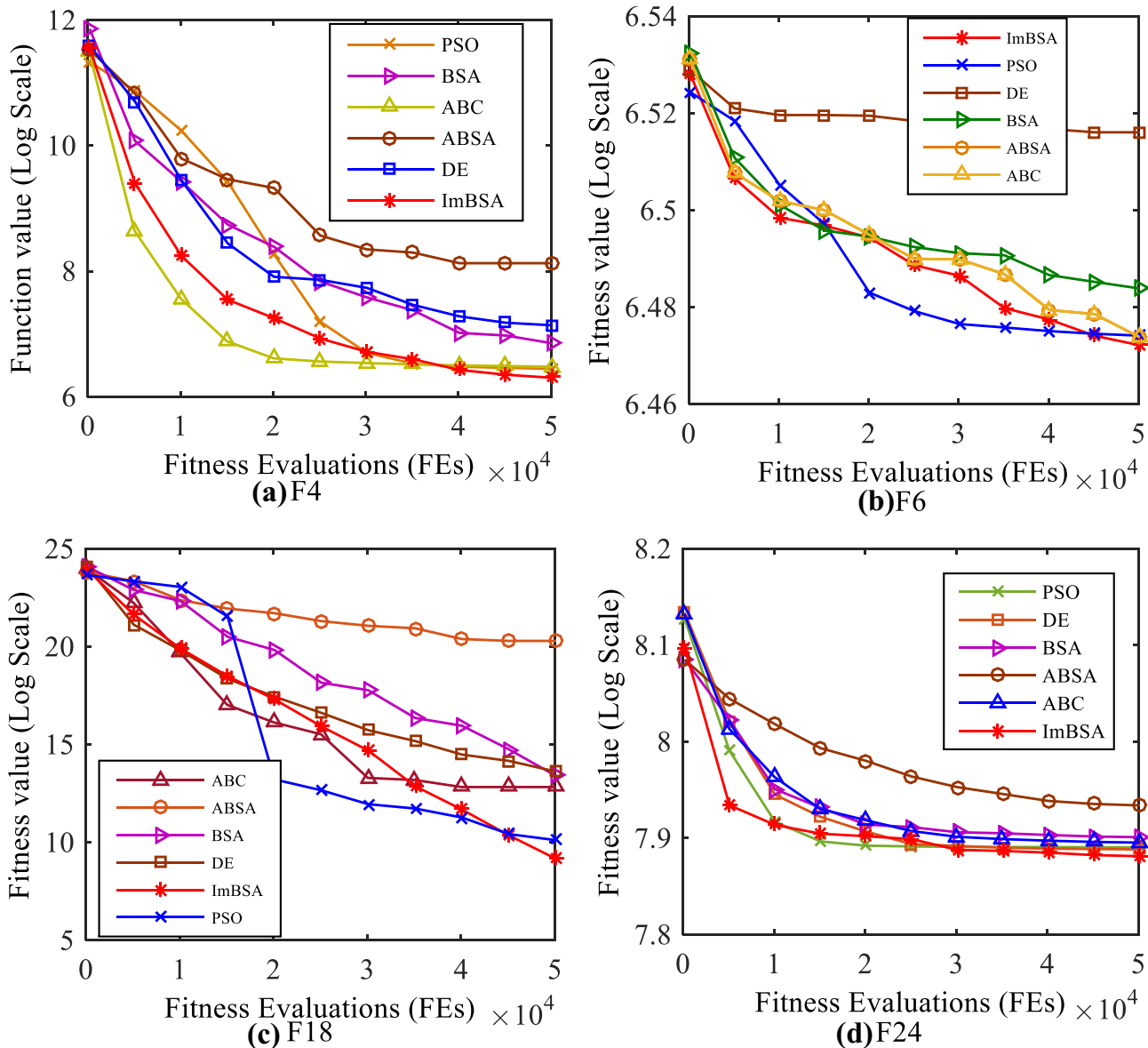


Fig. 2 Convergence graph of four functions (F4, F6, F18, and F24) with dimension $D=50$ and $D \times 1000$ FEs over fitness value vs. fitness evaluation

Table 4 Performance results of DE/rand/2/bin, DE/best/1/bin, DE/target-to-best/1/bin, DE/target-to-best/2/bin/, jDE, CoDE, and dimension (D) 50 after reaching D*1000 FEs of CEC 2014 test functions over 51 runs with 100 population size. Boldface represents the best result among the compared algorithms

F	DE/rand/2/bin	DE/best/1/bin	DE/target-to-best/1/bin	DE/target-to-best/2/bin/	jDE	CoDE	ImBSA
F1	1.03e+09 ± 1.35e+08	1.31e+08 ± 6.35e+07	9.16e+07 ± 3.92e+07	9.80e+07 ± 4.21e+07	3.12e+09 ± 5.59e+08	1.73+08 ± 3.39e+07	8.74e+07 ± 2.02e+07
F2	5.25e+10 ± 5.48e+09	2.69e+10 ± 7.70e+09	2.24e+10 ± 6.22e+09	2.09e+10 ± 5.09e+09	1.81e+11 ± 1.62e+10	3.87e+09 ± 9.18e+08	1.01e+08 ± 4.01e+07
F3	1.03e+05 ± 9.41e+03	1.04e+04 ± 7.55e+03	6.46e+03 ± 5.04e+03	7.24e+03 ± 3.65e+03	2.56e+05 ± 2.67e+04	2.37e+04 ± 3.65e+03	1.87e+04 ± 4.15e+03
F4	9.87e+03 ± 3.34e+03	3.46e+03 ± 1.49e+03	2.96e+03 ± 8.57e+02	3.04e+03 ± 9.55e+02	5.32e+04 ± 1.06e+04	1.22e+03 ± 1.00e+02	7.15e+02 ± 4.69e+01
F5	5.23e+02 ± 4.12e-02	5.24e+02 ± 4.10e-02	5.25e+02 ± 4.23e-02	5.26e+02 ± 3.94e-02	5.22e+02 ± 3.78e-02	5.27e+02 ± 4.82e-02	5.21e+02 ± 3.49e-02
F6	6.72e+02 ± 1.18e+00	6.42e+02 ± 4.67e+00	6.30e+02 ± 2.98e+00	6.31e+02 ± 3.00e+00	6.75e+02 ± 1.80e+00	6.63e+02 ± 1.69e+00	6.49e+02 ± 1.48e+00
F7	1.22e+03 ± 4.23e+01	9.88e+02 ± 6.56e+01	9.27e+02 ± 5.97e+01	9.12e+02 ± 5.13e+01	2.43e+03 ± 1.31e+02	7.37e+02 ± 8.04e+00	7.02e+02 ± 3.70e-01
F8	1.35e+03 ± 2.48e+01	1.05e+03 ± 4.13e+01	9.71e+02 ± 2.63e+01	9.62e+02 ± 2.45e+01	1.57e+03 ± 2.54e+01	1.11e+03 ± 1.65e+01	9.08e+02 ± 9.45e+00
F9	1.51e+03 ± 2.47e+01	1.42e+03 ± 5.29e+01	1.88e+03 ± 3.04e+01	1.98e+03 ± 3.31e+01	1.41e+03 ± 5.41e+01	1.35e+03 ± 1.85e+01	1.21e+03 ± 1.46e+01
F10	1.44e+04 ± 3.09e+02	6.70e+03 ± 8.95e+02	1.21e+04 ± 1.87e+03	1.20e+04 ± 1.57e+03	1.51e+03 ± 4.25e+02	9.67e+03 ± 2.93e+02	4.06e+03 ± 3.30e+02
F11	1.51e+04 ± 2.97e+02	1.08e+04 ± 3.27e+03	1.47e+04 ± 4.16e+02	1.46e+04 ± 5.70e+02	1.52e+04 ± 4.64e+02	1.36e+04 ± 4.05e+02	1.07e+04 ± 3.75e+02
F12	1.20e+03 ± 3.57e-01	1.20e+03 ± 4.31e-01	1.20e+03 ± 3.62e-01	1.20e+03 ± 3.59e-01	1.20e+03 ± 3.78e-01	1.20e+03 ± 2.81e-01	1.20e+03 ± 1.32e-01
F13	1.32e+03 ± 2.66e-01	1.37e+03 ± 7.88e-01	1.33e+03 ± 7.97e-01	1.31e+03 ± 9.07e-01	1.35e+03 ± 8.30e-01	1.34e+03 ± 8.03e-02	1.30e+03 ± 5.32e-02
F14	1.53e+03 ± 1.13e+01	1.47e+03 ± 1.84e+01	1.45e+03 ± 1.54e+01	1.46e+03 ± 1.45e+01	1.87e+03 ± 4.24e+01	1.41e+03 ± 3.15e+00	1.40e+03 ± 3.32e-02
F15	6.92e+05 ± 2.53e+05	5.19e+04 ± 5.97e+04	9.45e+03 ± 7.89e+03	9.15e+03 ± 7.09e+03	2.07e+07 ± 7.90e+06	2.50e+03 ± 7.72e+02	1.54e+03 ± 5.35e+00
F16	1.64e+03 ± 2.33e-01	1.65e+03 ± 3.98e-01	1.66e+03 ± 3.10e-01	1.67e+03 ± 3.84e-01	1.63e+03 ± 1.65e-01	1.66e+03 ± 1.93e-01	1.62e+03 ± 3.76e-01
F17	3.96e+07 ± 1.16e+07	8.65e+05 ± 1.26e+06	3.84e+05 ± 3.62e+05	4.45e+05 ± 6.01e+05	2.67e+08 ± 5.70e+07	6.28e+06 ± 1.89e+06	1.66e+07 ± 5.47e+06
F18	1.35e+07 ± 3.42e+06	8.89e+06 ± 2.37e+07	2.08e+06 ± 7.24e+06	9.79e+05 ± 3.97e+06	1.08e+10 ± 3.20e+09	6.65e+06 ± 1.83e+06	2.56e+04 ± 2.76e+04
F19	2.06e+03 ± 1.34e+01	2.02e+03 ± 3.23e+01	2.30e+03 ± 2.82e+01	2.31e+03 ± 3.30e+01	2.24e+03 ± 3.83e+02	1.98e+03 ± 7.49e+00	1.98e+03 ± 1.08e+01
F20	5.14e+04 ± 1.54e+04	6.28e+03 ± 2.99e+03	2.92e+03 ± 7.26e+02	2.74e+03 ± 5.46e+02	5.33e+05 ± 2.55e+05	9.51e+03 ± 2.27e+03	2.07e+04 ± 5.52e+03
F21	3.97e+06 ± 1.30e+06	2.59e+05 ± 2.41e+05	8.71e+04 ± 6.48e+04	1.02e+05 ± 8.93e+04	6.01e+07 ± 1.92e+07	9.22e+05 ± 2.81e+05	5.60e+06 ± 1.61e+06
F22	4.15e+03 ± 2.53e+02	3.17e+03 ± 2.61e+02	3.98e+03 ± 3.28e+02	3.97e+03 ± 3.29e+02	8.64e+03 ± 2.24e+03	3.85e+03 ± 1.62e+02	3.84e+03 ± 1.98e+02
F23	2.77e+03 ± 1.43e+01	2.88e+03 ± 5.18e+01	2.78e+03 ± 4.12e+01	2.87e+03 ± 4.02e+01	4.20e+03 ± 2.40e+02	2.75e+03 ± 0.00e+00	2.65e+03 ± 2.63e+00
F24	2.85e+03 ± 1.09e+01	2.75e+03 ± 1.35e+01	2.71e+03 ± 8.24e+00	2.73e+03 ± 8.94e+00	2.69e+03 ± 5.02e+01	3.69e+03 ± 0.00e+00	2.68e+03 ± 2.33e+00
F25	2.82e+03 ± 9.63e+00	2.76e+03 ± 1.01e+01	2.84e+03 ± 4.07e+00	2.86e+03 ± 4.13e+00	2.75e+03 ± 3.20e+01	2.90e+03 ± 0.00e+00	2.74e+03 ± 3.41e+00
F26	2.72e+03 ± 5.39e-01	2.81e+03 ± 3.81e+01	2.87e+03 ± 3.43e+01	2.89e+03 ± 2.65e+01	2.75e+03 ± 4.12e+01	2.71e+03 ± 8.82e-02	2.70e+03 ± 4.77e-02
F27	4.83e+03 ± 5.17e+01	4.81e+03 ± 1.32e+02	4.80e+03 ± 9.20e+01	4.71e+03 ± 9.13e+01	4.19e+03 ± 1.65e+02	4.90e+03 ± 0.00e+00	4.12e+03 ± 8.99e+01
F28	1.17e+04 ± 5.07e+02	7.62e+03 ± 9.33e+02	5.92e+03 ± 3.56e+02	5.90e+03 ± 5.52e+02	1.47e+04 ± 8.57e+02	5.20e+03 ± 9.19e-13	5.17e+03 ± 1.51e+02
F29	1.86e+08 ± 5.05e+07	4.74e+06 ± 1.05e+07	5.20e+06 ± 2.41e+07	6.26e+06 ± 2.17e+07	1.13e+09 ± 3.02e+08	3.10e+05 ± 0.00e+00	1.59e+05 ± 6.09e+04
F30	1.11e+06 ± 2.82e+05	1.29e+05 ± 1.00e+05	1.65e+05 ± 1.47e+05	1.50e+05 ± 1.15e+05	1.48e+07 ± 3.91e+06	5.50e+04 ± 0.00e+00	5.40e+04 ± 8.99e+03

Table 5 The number of occasions where ImBSA is superior to, inferior to, or similar to DE/rand/2/bin, DE/best/1/bin, DE/target-to-best/1/bin, DE/target-to-best/2/bin/, jDE, and CoDE on CEC 2014 benchmark suite with dimension 50

Algorithms	Superior to (ImBSA > algorithms)	Inferior to (ImBSA < algorithms)	Equal to (ImBSA = algorithms)
DE/rand/2/bin	28	1	1
DE/best/1/bin	23	6	1
DE/target-to-best/1/bin	24	5	1
DE/target-to-best/2/bin	24	5	1
jDE	28	1	1
CoDE	26	3	1

confidence, the obtained average rank difference is superior to the calculated CD (1.239978804). Therefore, from the above discussion and as shown in Fig. 1, it can be concluded that the performance of ImBSA on thirty CEC 2014 special session and competition test functions is statistically significant. Some graphical representations of the convergence performance of ImBSA are reported in Fig. 2.

Results from Fig. 2 show that ImBSA converges substantially more quickly than BSA in all functions. The convergences of both methods are somewhat comparable for several functions in the beginning stages, while the ImBSA convergence is superior for later stages.

Comparison with Some Self-Adaptive DE Variants

Computational results of thirty test functions with dimension 50, obtained by DE/best/1/bin [36], DE/rand/2/bin [37], DE/target-to-best/2/bin [37], DE/target-to-best/1/bin [38], jDE [39], CoDE [40], and ImBSA, are reported in Table 4. Results obtained by ImBSA are given in boldface if they are the best among the compared algorithms. From this table, several observations and conclusions can be made. The numbers of occasions where ImBSA is inferior to, equal to, and superior to other comparative algorithms are presented in Table 5. From the observation of Table 5, it is seen that ImBSA is superior to DE/rand/2/bin, DE/best/1/bin, DE/

target-to-best/1/bin, DE/target-to-best/2/bin/, jDE, CoDE on 28, 23, 24, 24, 28, and 26 test functions respectively; inferior to DE/rand/2/bin, DE/best/1/bin, DE/target-to-best/1/bin, DE/target-to-best/2/bin/, jDE, and CoDE on 1, 6, 5, 5, 1, and 3 test function respectively; and equal to one test function with each algorithms.

The Friedman rank test is also applied to thirty CEC 2014 special session and competition test functions (P) and seven algorithms (A) in this study. For each algorithm, the average ranks, the variance of ranks, and the average rank difference obtained by utilizing the average results of all test functions are presented in Table 6. At 95% confidence level, the critical value, i.e., F-statistic value, is 2.638. The calculated F-Score, i.e., Friedman statistics based on the average rank of all CEC2014 test functions, is shown in Table 6. It is observed that, at a 95% confidence level, the calculated F-Score value (3.543640898) is greater than the F-statistics value (2.638). Hence, the null hypothesis is rejected, and it is indicated that the mean ranks obtained by utilizing the average results of all the test functions are statistically different. Table 6 shows that, at a 95% confidence level, the computed critical difference value (1.4714061) is less than the mean rank difference. Hence, from the analysis of the result in Fig. 3, it is observed that ImBSA’s performance on thirty CEC 2014 special session and competition test functions is statistically significant at a 95% confidence level.

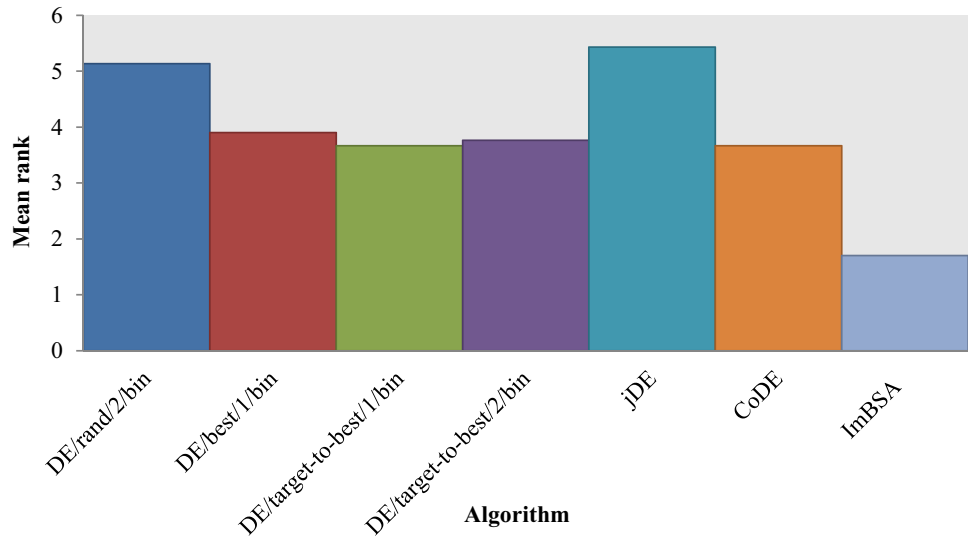
Table 6 Friedman test on CEC 2014 functions with 50 dimensions of DE/rand/2/bin, DE/best/1/bin, DE/target-to-best/1/bin, DE/target-to-best/2/bin/, jDE, CoDE, and ImBSA using mean performances

Algorithms	Average rank	Variance of rank	Average rank difference w.r.t. average rank
DE/rand/2/bin	5.133333333	2.050574713	3.433333333
DE/best/1/bin	3.9	1.886206897	2.2
DE/target-to-best/1/bin	3.666666667	2.643678161	1.966666667
DE/target-to-best/2/bin	3.766666667	3.357471264	2.066666667
jDE	5.433333333	5.288505747	3.733333333
CoDE	3.666666667	3.471264368	1.966666667
ImBSA	1.7	2.148275862	
F-Score			3.543640898
Critical difference (CD)			1.4714061
F-statistics			2.638

Table 7 Experimental results of PSO-cf, FI-PSO, CLPSO, DNLPSO, PSOGSA, DE-PSO, and ImBSA at 50 D after reaching D*1000 FEs on CEC 2014 test functions over 51 runs and 100 population size

F	PSO-cf	FI-PSO	CLPSO	DNLPSO	PSOGSA	DE-PSO	ImBSA
F1	1.48e+08±8.71e+07	1.80e+08±2.74e+07	4.24e+08±9.77e+07	6.31e+07±8.39e+07	7.68e+07±6.71e+07	1.19e+07±4.19e+06	8.74e+07±2.02e+07
F2	4.38e+09±3.01e+09	2.57e+09±3.24e+08	6.75e+09±9.24e+08	4.33e+09±1.07e+10	6.09e+09±6.68e+09	2.15e+07±2.58e+06	1.01e+08±4.01e+07
F3	5.78e+04±2.05e+04	1.92e+05±1.67e+04	1.20e+05±1.38e+04	8.30e+04±4.46e+04	9.96e+04±2.43e+04	6.53e+04±9.22e+03	1.87e+04±4.15e+03
F4	1.60e+03±4.55e+02	9.32e+02±5.18e+01	1.74e+03±1.92e+02	7.64e+02±1.13e+03	1.46e+03±8.24e+02	5.85e+02±2.74e+01	7.15e+02±4.69e+01
F5	5.31e+02±6.86e-02	5.27e+02±3.24e-02	5.26e+02±4.47e-02	5.28e+02±5.39e-02	5.22e+02±1.38e-01	5.25e+02±8.26e-02	5.21e+02±3.49e-02
F6	6.38e+02±6.09e+00	6.50e+02±2.56e+00	6.56e+02±1.74e+00	6.33e+02±1.20e+01	6.48e+02±5.16e+00	6.47e+02±1.03e+01	6.49e+02±1.48e+00
F7	7.45e+02±1.39e+01	7.25e+02±2.98e+00	7.61e+02±7.64e+00	7.29e+02±7.62e+01	7.55e+02±9.01e+01	7.01e+02±2.75e-02	7.02e+02±3.70e-01
F8	9.37e+02±2.95e+01	1.13e+03±1.49e+01	9.94e+02±1.57e+01	9.40e+02±5.10e+01	1.18e+03±6.57e+01	1.11e+03±9.60e+01	9.08e+02±9.45e+00
F9	1.11e+03±6.76e+01	1.31e+03±1.73e+01	1.34e+03±2.03e+01	1.07e+03±8.12e+01	1.38e+03±8.55e+01	1.28e+03±3.12e+01	1.21e+03±1.46e+01
F10	4.11e+03±8.52e+02	1.20e+04±3.49e+02	6.95e+03±4.58e+02	8.66e+03±3.62e+03	8.14e+03±1.05e+03	1.33e+04±7.50e+02	4.06e+03±3.30e+02
F11	8.07e+03±2.01e+03	1.48e+04±4.05e+02	1.12e+04±4.28e+02	1.18e+04±3.45e+03	8.62e+03±8.81e+02	1.40e+04±5.06e+02	1.07e+04±3.75e+02
F12	1.23e+03±8.03e-01	1.20e+03±4.22e-01	1.20e+03±2.13e-01	1.20e+03±1.42e+00	1.22e+03±2.18e-01	1.21e+03±5.12e-01	1.20e+03±1.32e-01
F13	1.36e+03±1.10e-01	1.31e+03±7.25e-02	1.33e+03±9.47e-02	1.34e+03±1.23e+00	1.35e+03±9.96e-01	1.36e+03±7.23e-02	1.30e+03±5.32e-02
F14	1.47e+03±1.64e+00	1.43+03±9.18e-01	1.44e+03±2.62e+00	1.41e+03±1.02e+01	1.46e+03±1.63e+01	1.45e+03±6.87e-02	1.40e+03±3.32e-02
F15	1.71e+03±3.42e+02	1.66e+03±4.62e+01	8.89e+03±3.75e+03	1.57e+03±1.07e+02	2.46e+04±4.61e+04	1.55e+03±4.07e+00	1.54e+03±5.35e+00
F16	1.66e+03±5.40e-01	1.65e+03±1.86e-01	1.64e+03±2.93e-01	1.67e+03±6.48e-01	1.63e+03±5.90e-01	1.72e+03±3.74e-01	1.62e+03±3.76e-01
F17	8.67e+06±7.83e+06	1.21e+07±3.02e+06	4.44e+07±1.51e+07	2.62e+06±1.79e+06	6.95e+06±5.92e+06	7.50e+05±4.09e+05	1.66e+07±5.47e+06
F18	2.06e+07±3.95e+07	7.50e+05±3.22e+05	5.67e+07±2.20e+07	2.79e+06±1.96e+07	1.45e+08±4.66e+08	4.04e+05±1.28e+05	2.56e+04±2.76e+04
F19	2.01e+03±8.3e+01	1.99e+03±1.22e+01	2.02e+03±1.02e+01	1.94e+03±2.82e+01	1.20e+03±7.19e+01	1.97e+03±2.14e+01	1.98e+03±1.08e+01
F20	2.47e+04±8.80e+03	6.03e+04±1.35e+04	2.33e+04±1.76e+04	4.39e+04±3.32e+04	6.22e+04±2.90e+04	7.87e+04±6.18e+03	2.07e+04±5.52e+03
F21	3.30e+06±3.71e+06	5.42e+06±1.78e+06	1.32e+07±4.13e+06	1.06e+06±8.17e+05	3.25e+06±2.11e+06	2.64e+05±1.25e+05	5.60e+06±1.61e+06
F22	3.53e+03±3.62e+02	3.89e+03±1.74e+02	3.87e+03±1.88e+02	3.97e+03±4.32e+02	3.83e+03±3.68e+02	3.92e+03±2.47e+02	3.84e+03±1.98e+02
F23	2.71e+03±1.49e+01	2.67e+03±1.73e+00	2.72e+03±1.16e+01	2.68e+03±4.78e+01	2.69e+03±5.50e+01	2.66e+03±1.95e+00	2.65e+03±2.63e+00
F24	2.69e+03±4.37e+00	2.75e+03±2.65e+00	2.72e+03±3.46e+00	2.70e+03±2.38e+01	2.71e+03±2.37e+01	2.88e+03±3.45e+00	2.68e+03±2.33e+00
F25	2.75e+03±1.10e+01	2.76e+03±4.85e+00	2.78e+03±8.61e+00	2.83e+03±2.28e+01	2.73e+03±1.10e+01	2.82e+03±3.78e+00	2.74e+03±3.41e+00
F26	2.79e+03±4.28e+01	2.72e+03±9.52e-02	2.71e+03±8.49e-01	2.78e+03±1.24e+02	2.80e+03±7.99e+01	2.75e+03±5.05e+01	2.70e+03±4.77e-02
F27	4.20e+03±1.63e+02	4.13e+03±5.61e+01	4.38e+03±1.75e+02	4.80e+03±2.42e+02	4.40e+03±2.17e+02	4.61e+03±1.11e+02	4.12e+03±8.99e+01
F28	7.54e+03±9.75e+02	5.55e+03±8.67e+01	5.99e+03±4.19e+02	6.37e+03±2.77e+02	6.10e+03±7.58e+02	6.83e+03±1.29e+03	5.17e+03±1.51e+02
F29	1.73e+07±4.34e+07	5.56e+05±1.61e+05	3.99e+07±7.05e+06	4.25e+07±4.57e+02	1.84e+07±3.20e+07	9.11e+05±1.95e+04	1.59e+05±6.09e+04
F30	1.44e+05±8.39e+04	1.02e+05±1.75e+04	3.10e+05±8.54e+04	5.50e+05±1.61e+03	7.55e+04±5.16e+04	5.70e+05±1.02e+04	5.40e+04±8.99e+03

Fig. 3 Ranks obtained by Friedman test on CEC 2014 test function at $D=50$



Comparison with some PSO variants

Table 7 shows the comparison of computational results obtained by PSO-cf [32], FI-PS, CLPSO [41], DNLPSO, PSO-GSA [41], DE-PSO, and ImBSA of thirty CEC2014 test functions with dimension 50. Results obtained by ImBSA are bolded if they are the best among the compared algorithms. From this table, the performance results of ImBSA, where it is inferior to, equal to, and superior to the above algorithms, are presented in Table 8. From Table 8, it is seen that the performance of ImBSA is superior to PSO-cf, FI-PS, CLPSO, DNLPSO, PSO-GSA, DE-PSO on 24, 27, 29, 24, 23, and 22 test functions; inferior on 6, 2, 0, 6, 7, 8, and 6 occasions; and similar to FI-PS, CLPSO, and DNLPSO on 1 case, respectively.

Next, the Friedman rank test has been conducted on thirty CEC2014 benchmark functions on ImBSA and seven above algorithms. For each algorithm, the average ranks and variance of positions obtained by utilizing all test functions’ average results are presented in Table 9. Here, at 95% confidence level, the critical value, i.e.,

Table 8 The number of occasions where ImBSA is superior to, inferior to, or similar to PSO-cf, FI-PSO, CLPSO, DNLPSO, PSO-GSA, and DE-PSO on CEC 2014 benchmark suite with dimension 50

Algorithms	Superior to (ImBSA > algorithms)	Inferior to (ImBSA < algorithms)	Equal to (ImBSA = algorithms)
PSO-cf	24	6	0
FI-PSO	27	2	1
CLPSO	29	0	1
DNLPSO	24	6	1
PSO-GSA	23	7	0
DE-PSO	22	8	0

F-statistic value, and the calculated F-Score, i.e., Friedman statistics, are reported in Table 9. From Table 9, it is seen that, at a 95% confidence level, the calculated F-Score value (4.243870263) is greater than the F-statistics value (2.638). Hence, the null hypothesis is rejected, and it indicates that the average ranks obtained by utilizing the average results of all the test functions are statistically different. Also, it is observed from Table 9 and Fig. 4 that the mean rank difference is superior to the calculated CD value (1.4714061). Moreover, the analysis summarized that ImBSA presents acceptable performance on thirty CEC 2014 special session and competition test functions.

Results and Discussion on CEC 2017 Test Function

In CEC 2017 test functions, there are unimodal functions, simple multimodal functions, hybrid functions, and composition functions. One can see the details of these functions in [40]. The obtained results of ImBSA are compared

Table 9 Friedman test on CEC 2014 functions with 50 dimensions of the algorithms PSO-cf, FI-PSO, CLPSO, DNLPSO, PSO-GSA, DE-PSO, and ImBSA using their mean performances

Algorithms	Mean rank	Variance of rank	A.V. difference
PSO-cf	4.2	3.682758621	2.233333333
FI-PSO	4.1	2.575862069	2.133333333
CLPSO	4.966666667	3.343678161	3
DNLPSO	4	3.862068966	2.033333333
PSO-GSA	4.566666667	3.21954023	2.6
DE-PSO	3.966666667	4.929885057	2
ImBSA	1.966666667	2.309195402	
F-Score			4.243870263
CD			1.4714061
F-statistics (critical value)			2.638

Table 10 Experimental results of ImBSA, ABSA, IBSA, OBSA, HBSA, SCA, and HSCA for 10-dimensional CEC 2017 benchmark functions

F	ImBSA	ABSA	IBSA	OBSA	HBSA	SCA	HSCA
F1	4.06450E-08 ± 1.51048E-07	7.07948E+05 ± 4.94628E+05	3.30206E+08 ± 3.34094E+08	7.25045E-06 ± 4.32158E-05	1.11458E-15 ± 3.85589E-15	3.32000E+09 ± 2.37000E+09	2.07000E+06 ± 7.80000E+06
F2	NA	NA	NA	NA	NA	NA	NA
F3	6.37722E-01 ± 2.54134E+00	8.04382E+03 ± 2.71762E+03	3.03498E+02 ± 3.68520E+02	3.06281E-01 ± 1.07817E+00	2.22915E-15 ± 1.14351E-14	5.02000E+03 ± 4.03000E+03	1.66000E+02 ± 1.06000E+02
F4	1.94320E+00 ± 9.58256E-01	6.61439E+00 ± 1.24643E+00	2.90127E+01 ± 1.97455E+01	6.41243E-01 ± 4.67706E-01	2.55878E+00 ± 9.93708E-01	1.80000E+02 ± 1.48000E+02	7.94000E+00 ± 6.56000E+00
F5	4.55921E+00 ± 1.96703E+00	1.14124E+01 ± 2.91119E+00	7.05064E+00 ± 2.39545E+00	6.33379E+00 ± 2.54348E+00	6.63367E+00 ± 2.54348E+00	6.57000E+01 ± 1.69000E+01	1.59000E+01 ± 7.05000E+00
F6	4.01248E-14 ± 5.48698E-14	2.30373E-01 ± 1.46815E-01	2.90350E-01 ± 3.81588E-01	1.22595E-11 ± 4.48235E-11	2.22209E-07 ± 1.58689E-06	3.10000E+01 ± 1.33000E+01	8.77000E-01 ± 8.65000E-01
F7	1.32304E+00 ± 1.85236E+00	2.77842E+01 ± 3.76909E+00	1.68171E+01 ± 1.73865E+00	1.83258E+01 ± 2.33556E+00	1.46251E+01 ± 2.00179E+00	1.02000E+02 ± 3.68000E+01	2.81000E+01 ± 7.49000E+00
F8	5.22380E+00 ± 1.83321E+00	1.23568E+01 ± 3.04022E+00	6.48019E+00 ± 1.75070E+00	7.38134E+00 ± 2.40202E+00	5.80116E+00 ± 2.25880E+00	5.20000E+01 ± 1.45000E+01	1.50000E+01 ± 4.86000E+00
F9	2.22915E-15 ± 1.59193E-14	5.48454E+00 ± 3.41014E+00	1.18372E+00 ± 2.54913E+00	0.00000E+00 ± 0.00000E+00	6.23582E-02 ± 1.57899E-01	2.98000E+02 ± 2.25000E+02	6.07000E+00 ± 1.20000E+01
F10	2.00454E+02 ± 1.05990E+02	3.24072E+02 ± 1.13252E+02	2.58206E+02 ± 1.19058E+02	3.49525E+02 ± 1.77023E+02	2.43753E+02 ± 1.46112E+02	1.39000E+03 ± 1.85000E+02	5.58000E+02 ± 2.46000E+02
F11	1.74827E+00 ± 9.32627E-01	2.26297E+01 ± 1.07151E+01	2.51691E+00 ± 1.52342E+02	2.64269E+01 ± 1.32556E+00	4.55542E+00 ± 3.88022E+00	3.19000E+02 ± 3.14000E+02	2.60000E+01 ± 1.20000E+01
F12	6.22038E+03 ± 6.21966E+03	6.05160E+05 ± 4.45048E+05	1.24582E+05 ± 2.99688E+05	2.50163E+03 ± 2.27373E+03	4.62871E+03 ± 4.48480E+03	6.20000E+07 ± 8.72000E+07	4.41000E+05 ± 6.21000E+05
F13	6.02678E+00 ± 3.02507E+00	9.88310E+03 ± 1.03839E+04	7.56871E+03 ± 5.30705E+03	1.00502E+01 ± 3.96874E+00	6.34249E+00 ± 3.52732E+00	2.09000E+05 ± 3.88000E+05	8.67000E+03 ± 7.23000E+03
F14	7.32681E-01 ± 7.21789E-01	2.63668E+02 ± 3.03758E+02	3.31824E+00 ± 3.03680E+03	2.17640E+03 ± 1.26462E+00	5.00763E+00 ± 3.62063E+00	5.22000E+02 ± 1.16000E+03	6.61000E+01 ± 1.79000E+01
F15	6.70991E-01 ± 6.84973E-01	6.76427E+02 ± 6.78677E+02	1.09214E+03 ± 2.14520E+03	8.60043E-01 ± 5.64822E-01	3.14392E+00 ± 3.24683E+00	2.16000E+03 ± 2.71000E+03	2.09000E+02 ± 4.43000E+02
F16	1.43771E+00 ± 6.48368E+00	1.80948E+01 ± 1.16273E+01	8.14264E+01 ± 7.68210E+01	2.42898E+00 ± 4.17197E+00	2.37561E+01 ± 5.111309E+01	2.29000E+02 ± 1.24000E+02	5.34000E+01 ± 6.28000E+01
F17	8.62933E-01 ± 5.86130E-01	1.11013E+01 ± 5.42658E+00	1.23049E+01 ± 1.41917E+01	7.17343E+00 ± 7.80053E+00	1.15749E+01 ± 1.01530E+01	1.12000E+02 ± 4.15000E+01	4.49000E+01 ± 1.05000E+01
F18	3.66309E-01 ± 3.69762E-01	1.81728E+04 ± 1.22786E+04	5.79293E+00 ± 1.55627E+03	7.99411E+00 ± 6.53437E+00	2.20613E+00 ± 1.72834E+00	3.29000E+05 ± 5.65000E+05	1.57000E+04 ± 1.07000E+04
F19	8.02427E-02 ± 2.37995E-01	1.34211E+03 ± 2.17297E+03	1.44889E+03 ± 1.68302E+03	5.17554E-01 ± 5.59608E-01	4.72309E-01 ± 7.71608E-01	6.40000E+03 ± 7.80000E+03	4.77000E+02 ± 2.20000E+03
F20	8.02935E-02 ± 2.57251E-01	5.96673E+00 ± 4.59777E+00	1.14000E+01 ± 2.87046E+01	2.77416E-01 ± 3.70850E-01	4.22042E-01 ± 4.72504E-01	1.46000E+02 ± 5.56000E+01	3.16000E+01 ± 1.07000E+01
F21	1.33567E+02 ± 4.89450E+01	1.15551E+01 ± 5.54783E+01	1.40670E+02 ± 3.15181E+01	1.53471E+02 ± 2.56221E+01	1.74386E+02 ± 5.08661E+01	1.38000E+02 ± 3.42000E+01	1.02000E+02 ± 3.05000E+00
F22	7.62982E+01 ± 3.49365E+01	9.29918E+01 ± 2.67701E+01	9.38909E+01 ± 2.17402E+01	9.39188E+01 ± 2.04327E+01	1.01436E+02 ± 1.24388E+00	3.34000E+02 ± 1.89000E+02	9.11000E+01 ± 3.06000E+01
F23	3.01655E+02 ± 4.12870E+01	3.15898E+02 ± 2.86088E+00	3.10103E+02 ± 1.63852E+01	3.11475E+02 ± 4.54940E+00	3.11372E+02 ± 5.51697E+00	3.75000E+02 ± 1.51000E+01	3.17000E+02 ± 4.64000E+01
F24	2.56602E+02 ± 1.10028E+02	3.38323E+02 ± 6.03892E+01	2.13810E+02 ± 6.34241E+01	2.14799E+02 ± 1.21421E+02	2.96583E+02 ± 9.20899E+01	3.50000E+02 ± 9.78000E+01	1.69000E+02 ± 1.11000E+02
F25	4.10139E+02 ± 2.00865E+01	4.04219E+02 ± 1.12893E+01	4.39821E+02 ± 1.34395E+01	3.99002E+02 ± 6.39319E+00	4.30831E+02 ± 2.15515E+01	5.72000E+02 ± 1.21000E+02	4.18000E+02 ± 2.17000E+01
F26	2.74142E+02 ± 6.95722E+01	3.11974E+02 ± 5.52662E+01	3.16450E+02 ± 1.26831E+02	2.74371E+02 ± 9.09839E+01	2.80785E+02 ± 7.66952E+01	6.90000E+02 ± 1.86000E+02	3.08000E+02 ± 6.83000E+01
F27	3.90809E+02 ± 2.43357E+00	3.91701E+02 ± 1.98929E+00	4.05688E+02 ± 8.12464E+00	3.91487E+02 ± 2.16770E+00	4.19356E+02 ± 5.23487E+01	4.18000E+02 ± 1.41000E+01	3.98000E+02 ± 5.15000E+00
F28	3.45350E+02 ± 1.31515E+02	3.91793E+02 ± 6.82571E+01	4.88280E+02 ± 1.36353E+02	4.87903E+02 ± 9.74021E+00	4.91550E+02 ± 1.20698E+01	5.84000E+02 ± 1.01000E+02	3.54000E+02 ± 3.90000E+01
F29	2.53236E+02 ± 8.32787E+00	2.57739E+02 ± 1.02854E+01	1.02854E+01 ± 1.52174E+01	2.63991E+02 ± 1.26315E+01	2.62634E+02 ± 1.96069E+01	3.84000E+02 ± 5.18000E+01	2.76000E+02 ± 1.79000E+01
F30	4.30792E+02 ± 6.59888E+03	7.23238E+04 ± 1.65358E+05	3.48831E+04 ± 3.36968E+04	2.76666E+04 ± 2.97105E+04	5.76240E+02 ± 6.69693E+02	2.15000E+06 ± 1.68000E+06	4.07000E+04 ± 5.58000E+04

Fig. 4 Ranks obtained by Friedman test on CEC 2014 function at $D=50$

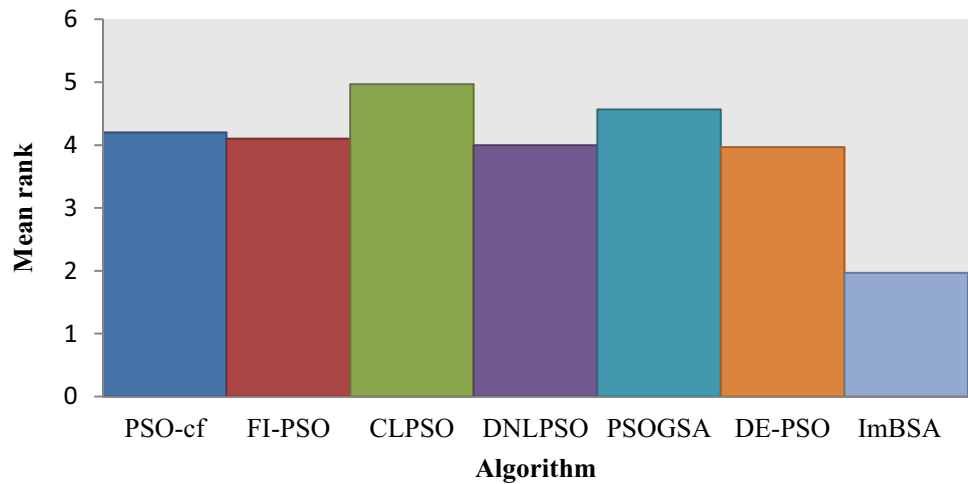


Fig.4. Ranks obtained by Friedman test on CEC2014 function at $D = 50$

with ABSA [13], IBSA [4], OBSA [42], HBSA [2], SCA [43], and HSCA [44] respectively. For fair comparison, the parameter setting for all the algorithms is kept the same. The performance results of these algorithms on the test functions are presented in Table 10. Table 11 presents the counting number that ImBSA is better than, worse than, and similar to the compared algorithms. This table shows that ImBSA is superior to ABSA, IBSA, OBSA, HBSA, SCA, and HSCA on 27, 28, 24, 26, 29, and 27 test functions.

For statistical analysis, the Friedman rank test has been conducted based on the mean performance on CEC 2017 for each algorithm. By utilizing the average results of all CEC2017 test functions, the obtained average ranks and variance of ranks are presented in Table 14. At 95% confidence level, the obtained critical value (F-statistic value) and Friedman statistics (F-Score) are reported in Table 12. Table 12 shows that, at a 95% confidence level, the calculated F-Score value (37.86853147) is greater than the F-statistics value (2.638), which indicates that the average ranks obtained by utilizing the average results of all the test functions are statistically different. It is also observed from

Table 12 and Fig. 5 that the mean rank difference is superior to the calculated CD value (1.496560163). Hence, it is summarized that ImBSA presents outstanding performance on the CEC2017 special session and competition test functions from the analysis.

Complexity Analysis

This section details the complexity of our ImBSA algorithms with dimensions 50 and 10 correspondingly for IEEE CEC 2014 and 2017 test functions. Analysis of an algorithm’s complexity is useful for comparing the suggested algorithm with other algorithms regarding the time consumed to solve a problem. The algorithm with lower complexity is considered efficient. The present work calculates the complexity according to the definition presented in IEEE CEC 2014 and 2017. Algorithms 2 and 3 represent the simple codes used

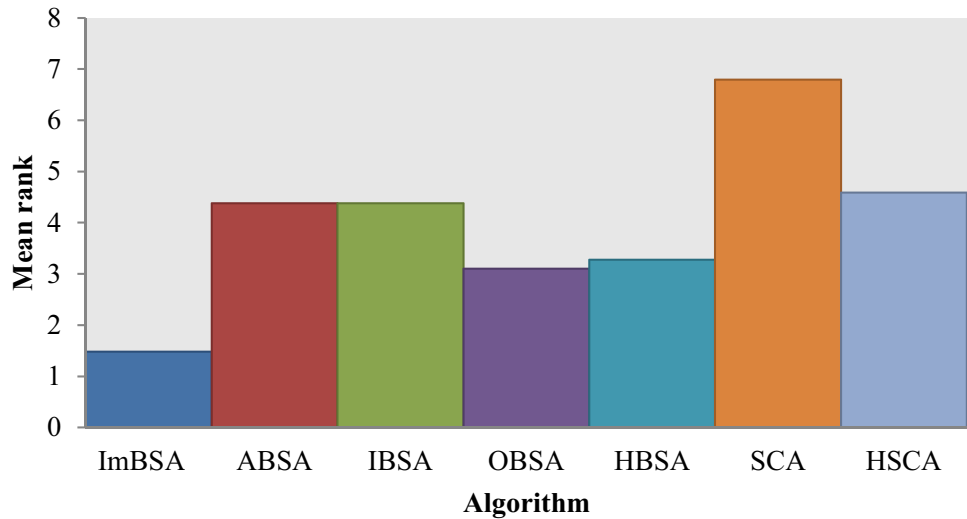
Table 11 The number of occasions where ImBSA is superior to, inferior to, and similar to the compared algorithms on 10-dimension CEC 2017 benchmark functions

Algorithms	Superior to (ImBSA > algorithms)	Inferior to (ImBSA < algorithms)	Equal to (ImBSA = algorithms)
ABSA	27	2	0
IBSA	28	1	0
OBSA	24	5	0
HBSA	26	3	0
SCA	29	0	0
HSCA	27	2	0

Table 12 Friedman test on CEC2017 real-world benchmark functions with 10 dimensions of ABSA, IBSA, OBSA, HBSA, SCA, HSCA, and ImBSA based on their mean performances

Algorithms	Average rank	Variance of rank	Average rank difference w.r.t. average rank
ImBSA	1.482758621	0.75862069	
ABSA	4.379310345	1.815270936	2.896551724
IBSA	4.379310345	1.958128079	2.896551724
OBSA	3.103448276	2.596059113	1.620689655
HBSA	3.275862069	2.63546798	1.793103448
SCA	6.793103448	0.384236453	5.310344828
HSCA	4.586206897	2.179802956	3.103448276
F-Score			37.86853147
Critical difference (CD)			1.496560163
F-statistics			2.638

Fig. 5 Ranks obtained by Friedman test on CEC 2017 test functions of 10D



to evaluate the value of T_0 in IEEE CEC 2014 and 2017 functions, respectively. These codes were recommended in the problem definition and evaluation criteria of IEEE CEC 2014 and 2017, respectively. The calculated algorithm complexity of the test function IEEE CEC 2014 [34] and 2017 [35] is shown in Tables 13 and 14.

Algorithm 2

```

for i = 1:1,000,000
    x = 0.55 + (double)i;
    x = x + x; x = x/2; x = x*x;
    x = sqrt(x); x = log(x); x = exp(x); x = x/(x + 2);
end
    
```

Algorithm 3

```

x = 0.55;
for i = 1:1,000,000
    x = x + x; x = x/2; x = x*x; x = sqrt(x); x = log(x); x = exp(x); x = x/(x + 2);
end
    
```

- T_1 is the time to execute 200,000 evaluations of benchmark function F18 by itself with D dimensions.
- T_2 is the time to execute ImBSA with 200,000 evaluations of F18 in D dimensions.
- $\widehat{T_2}$ is the average T_2 value of 5 runs.

Table 13 Runtime complexity of IEEE CEC 2014 test function with $D=50$ (in seconds)

Algorithm	T_0	T_1	$\widehat{T_2}$	$(\widehat{T_2} - T_1)/T_0$
BSA	0.1134	5.0699	16.5214	100.9832
ImBSA	0.1134	3.1012	3.6397	4.748677

According to Tables 13 and 14, both T_1 and $\widehat{T_2}$ scaled linearly higher of BSA than ImBSA for the respective dimensions, as shown by the paired comparison of the value of $(\widehat{T_2} - T_1)/T_0$.

Figures 6 and 7 have been drawn to depict the intricacy better. From these two figures, it can be observed that the height of the cone of ImBSA is lower than the height of the cone of BSA. This indicates that BSA is more complex than ImBSA. Thus, in terms of runtime complexity, ImBSA has a superior algorithm than BSA.

State-of-the-Art Recent BSA Variants

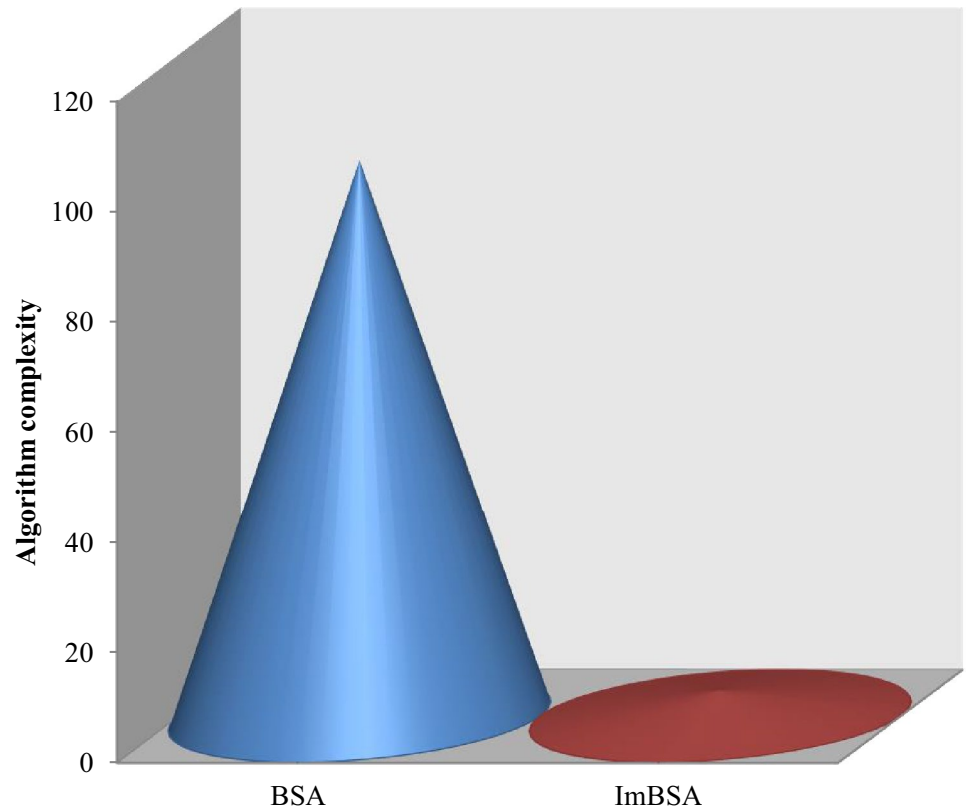
In this section, the proposed ImBSA is compared with some recently proposed variants of BSA, namely GWO-BSA [45], e-SOSBSA [46], BS-DE [47], LFBSA [11], IBS [10], and hDEBSA [48]. For this comparison, the CEC 2017 test functions with ten dimensions have been considered.

The details of these functions can be seen in [31]. For fare comparison, the parameter setting of all the algorithms is kept the same. The performance results of these algorithms on the test functions are presented in Table 15. Table 16 presents the number of functions where the proposed ImBSA is better than, worse than, and similar to the compared algorithms. This table shows that ImBSA is superior to GWO-BSA, BS-DE, LFBSA, IBS, and hDEBSA on all the twenty-nine test functions, whereas ImBSA is superior to e-SOSBSA on 20 test functions.

Table 14 Runtime complexity of IEEE CEC 2017 test function with $D=10$ (in seconds)

Algorithm	T_0	T_1	$\widehat{T_2}$	$(\widehat{T_2} - T_1)/T_0$
BSA	0.0439	3.183	10.6769	170.7039
ImBSA	0.0439	2.0938	7.4529	122.0752

Fig. 6 Algorithm complexity on CEC 2014 test functions with 50D



For statistical analysis, the Friedman rank test has been conducted based on the mean performance on CEC 2017 for

each algorithm. By utilizing the average results of all CEC 2017 test functions, the obtained average ranks and variance

Fig. 7 Algorithm complexity of CEC 2017 test function with 10D

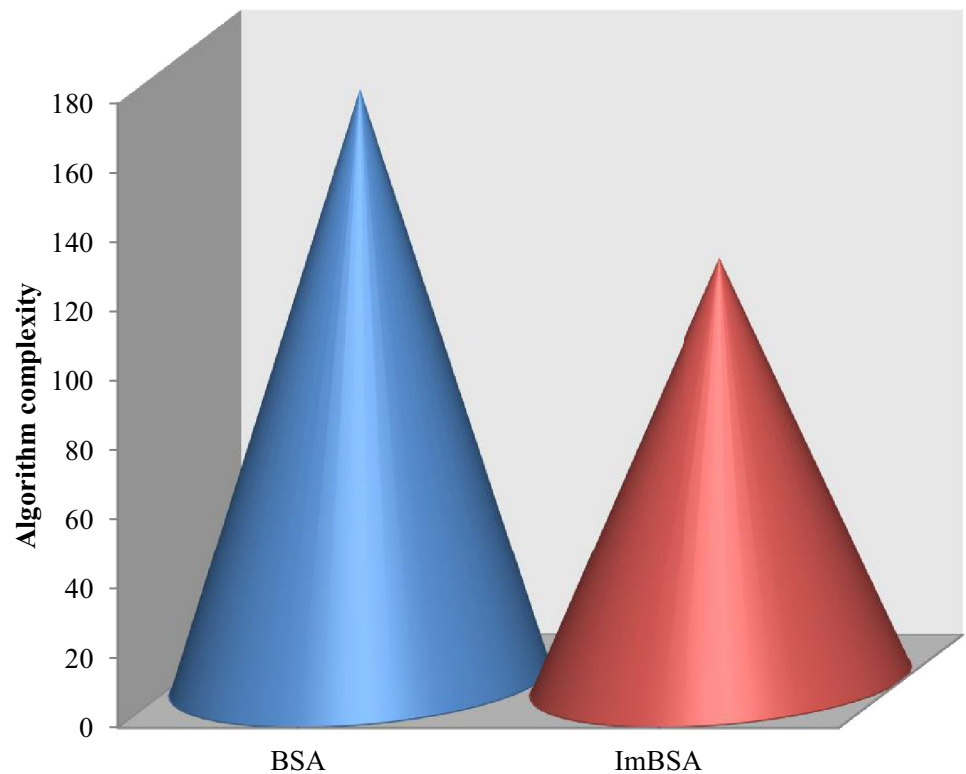
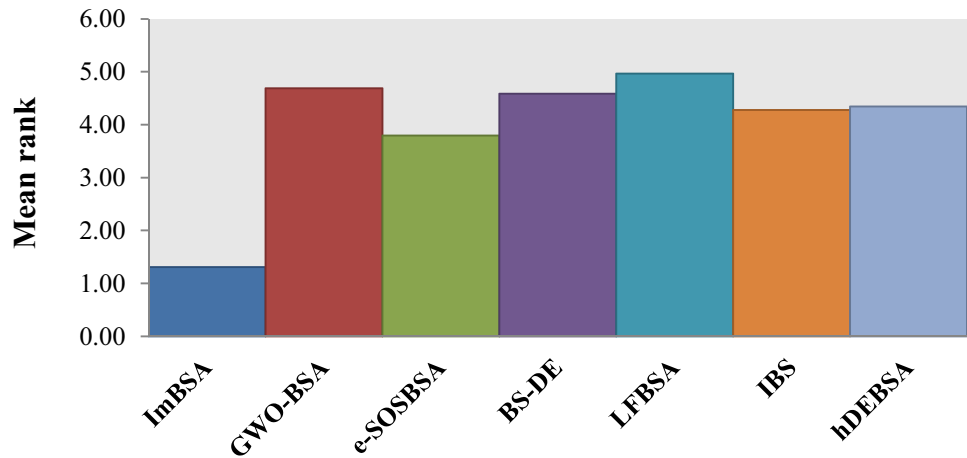


Fig.7.Algorithm complexity of cec17 test function with 10D

Table 15 Experimental results of ImBSA, GWO-BSA, e-SOSBSA, BS-DE, LFBSA, IBS, and hDEBSA at for 10-dimensional CEC 2017 benchmark functions

F	ImBSA	GWO-BSA	e-SOSBSA	BS-DE	LFBSA	IBS	hDEBSA
F1	4.06450E-08 ± 1.51048E-07	4.03647E+07 ± 2.94746E+05	2.85E+02 ± 9.72E+02	1.29471E+05 ± 1.92756E+05	1.29476E+02 ± 4.98473E+02	7.1.1986E+03 ± 5.19285E+09	6.54891E+01 ± 1.62456E+02
F3	6.37722E-01 ± 2.54134E+00	5.98574E+03 ± 1.37465E+03	1.11E-15 ± 7.96E-15	2.61902E+02 ± 1.89234E+02	1.92847E+02 ± 1.92834E+02	2.18576E+03 ± 5.01931E+03	1.84128E+00 ± 1.26756E+01
F4	1.94320E+00 ± 9.38256E-01	7.38475E+00 ± 1.35465E+00	9.25E-02 ± 8.95E-02	5.97164E+00 ± 6.19402E+00	7.18371E+00 ± 7.02947E+00	1.76564E+02 ± 1.62535E+01	4.61973E+00 ± 1.02565E+00
F5	4.55921E+00 ± 1.96703E+00	1.83756E+01 ± 3.93485E+00	4.81E+00 ± 2.19E+00	1.38902E+01 ± 104967E+00	1.70918E+01 ± 3.02947E+00	3.91847E+01 ± 1.72645E+01	4.80952E+00 ± 1.40234E+00
F6	4.01248E-14 ± 5.48698E-14	7.38374E-01 ± 5.63546E-01	5.57E-08 ± 2.79E-07	7.109175E-01 ± 5.10948E-01	9.01836E-01 ± 7.01846E-01	4.19386E-01 ± 1.26355E+01	2.22915E-12 ± 1.59645E-12
F7	1.32304E+01 ± 1.85236E+00	2.27476E+01 ± 3.28375E+00	1.59E+01 ± 2.48E+00	3.71902E+01 ± 4.10759E+00	3.82856E+01 ± 5.71493E+00	1.64552E+02 ± 3.45246E+01	1.56259E+01 ± 2.41142E+00
F8	5.22380E+00 ± 1.833321E+00	1.82746E+01 ± 3.83746E+00	4.19E+00 ± 1.81E+00	1.01947E+01 ± 3.19476E+00	2.73658E+01 ± 5.82756E+00	1.18386E+01 ± 1.19387E+01	5.72939E+00 ± 1.81455E+00
F9	2.22915E-15 ± 1.59193E-14	5.83756E-03 ± 3.83746E+00	1.42E-02 ± 6.64E-02	5.02345E-03 ± 4.14867E+01	7.19386E-03 ± 1.39576E+01	2.92837E-03 ± 2.19287E-01	1.52847E-13 ± 1.38275E-12
F10	2.00454E+02 ± 1.05990E+02	3.82746E+02 ± 1.92847E+02	1.56E+02 ± 1.20E+02	7.17245E+02 ± 1.97836E+02	5.79184E+02 ± 3.18375E+02	1.23874E+03 ± 1.19298E+02	2.20564E+02 ± 1.24396E+02
F11	1.74827E+00 ± 9.32627E-01	2.92847E+01 ± 1.2847E+01	1.28E+00 ± 1.56E+00	3.00137E+01 ± 1.87356E+01	3.13562E+01 ± 1.28357E+01	3.10129E+02 ± 3.18182E+02	1.85319E+00 ± 1.09356E+00
F12	6.22038E+03 ± 6.21966E+03	7.27465E+03 ± 3.92847E+03	1.05E+04 ± 1.00E+04	9.73465E+03 ± 3.19486E+03	7.11849E+03 ± 3.01846E+03	6.99286E+03 ± 8.19283E+03	7.65847E+03 ± 4.19685E+03
F13	6.02678E+00 ± 3.02507E+00	1.03747E+02 ± 1.38576E+02	1.05E+02 ± 2.81E+02	1.62143E+01 ± 2.23961E+01	3.19385E+01 ± 3.19385E+01	2.92827E+01 ± 3.19183E+01	8.87484E+01 ± 4.12763E+02
F14	7.32681E-01 ± 7.21789E-01	9.92847E+02 ± 2.92857E+02	7.05E+00 ± 1.48E+01	2.19485E+01 ± 1.20185E+01	2.19386E+01 ± 1.13295E+01	5.73646E+02 ± 1.74654E+03	9.43174E-01 ± 7.83851E-01
F15	6.70991E-01 ± 6.84973E-01	6.91834E+02 ± 5.92948E+02	8.59E-01 ± 6.45E-01	1.93857E+02 ± 1.12482E+02	2.16457E+02 ± 1.17562E+02	7.18273E+01 ± 2.73654E+01	4.56182E-01 ± 4.58465E-01
F16	1.43771E+00 ± 6.48368E+00	2.29193E+01 ± 2.19385E+01	2.56E+01 ± 5.44E+01	1.31937E+01 ± 1.02386E+01	2.19386E+01 ± 5.13057E+01	2.32837E+01 ± 1.54673E+01	5.47730E+01 ± 2.33786E+01
F17	8.62933E-01 ± 5.86130E-01	1.19183E+01 ± 1.19384E+00	3.41E+00 ± 5.90E+00	2.19385E+01 ± 1.27465E+01	3.82762E+01 ± 1.19285E+01	1.64647E+01 ± 4.54648E+01	1.56191E+00 ± 3.98456E+00
F18	3.66309E-01 ± 3.69762E-01	1.19185E+02 ± 1.82831E+02	4.86E+02 ± 1.26E+03	1.18274E+02 ± 1.12746E+02	1.93871E+02 ± 1.17532E+02	3.54643E+01 ± 5.74553E+01	6.97014E+02 ± 5.26565E+02
F19	8.02427E-02 ± 2.37995E-01	1.31829E+00 ± 1.11892E+00	1.57E+00 ± 4.18E+00	3.19373E+02 ± 1.19287E+02	4.19386E+02 ± 2.21029E+02	7.43874E+00 ± 5.94848E+00	1.53107E+01 ± 3.46865E+01
F20	8.02935E-02 ± 2.57251E-01	7.19385E+00 ± 4.91837E+00	8.78E-01 ± 2.38E+00	1.11075E+01 ± 1.10275E+01	3.11837E+01 ± 1.05729E+01	1.74657E+00 ± 5.83746E+00	1.41538E-01 ± 3.32796E-01
F21	1.33567E+02 ± 4.89450E+01	1.51836E+02 ± 4.59185E+01	1.13E+02 ± 4.05E+01	1.79183E+02 ± 3.05000E+00	1.92193E+02 ± 3.19385E+01	1.54648E+02 ± 3.54637E+01	1.97274E+02 ± 5.02865E+01
F22	7.62982E+01 ± 3.49365E+01	9.19375E+01 ± 2.91821E+01	9.47E+01 ± 2.39E+01	7.91023E+01 ± 4.10983E+01	9.37182E+01 ± 3.18375E+01	8.54728E+01 ± 1.72382E+01	9.61525E+01 ± 1.40896E+01
F23	3.01655E+02 ± 4.12870E+01	3.03017E+02 ± 2.10365E+00	3.06E+02 ± 3.24E+00	3.01927E+02 ± 4.19375E+01	3.04516E+02 ± 1.97234E+01	3.02776E+02 ± 1.18287E+01	3.06760E+02 ± 2.58596E+00
F24	2.56602E+02 ± 1.10028E+02	2.59063E+02 ± 1.01759E+01	2.49E+02 ± 1.17E+02	2.91745E+02 ± 2.01481E+02	2.71834E+02 ± 1.79175E+02	2.84737E+02 ± 9.19928E+01	2.97862E+02 ± 1.05868E+02
F25	4.10139E+02 ± 2.00865E+01	4.16422E+02 ± 1.38567E+01	4.20E+02 ± 2.28E+01	4.17024E+02 ± 2.38576E+01	4.19167E+02 ± 2.11938E+01	4.15287E+02 ± 1.29387E+02	4.19522E+02 ± 2.27856E+01
F26	2.74142E+02 ± 6.95722E+01	2.92195E+02 ± 2.92752E+01	2.96E+02 ± 3.10E+01	2.90294E+02 ± 3.08124E+01	2.91763E+02 ± 1.92857E+01	2.82736E+02 ± 1.29837E+02	2.93798E+02 ± 1.42235E+02
F27	3.90809E+02 ± 2.43357E+00	3.91543E+02 ± 3.91756E+00	3.93E+02 ± 2.82E+00	3.91947E+02 ± 2.19857E+00	3.91275E+02 ± 5.18756E+01	3.91538E+02 ± 1.39836E+01	3.93704E+02 ± 2.17832E+00
F28	3.45350E+02 ± 1.31515E+02	3.47213E+02 ± 5.18476E+01	3.30E+02 ± 9.97E+01	3.92847E+02 ± 3.19285E+01	3.71029E+02 ± 3.71345E+01	3.84476E+02 ± 1.20938E+02	3.86240E+02 ± 1.47596E+02
F29	2.53236E+02 ± 8.32787E+00	2.58309E+02 ± 1.71044E+01	2.50E+02 ± 1.09E+01	2.59173E+02 ± 1.91837E+01	2.58235E+02 ± 1.12783E+01	2.57937E+02 ± 5.19287E+01	2.59285E+02 ± 9.02896E+00
F30	4.30792E+02 ± 6.59888E+03	3.27469E+03 ± 1.24783E+03	2.01E+04 ± 1.14E+05	9.14790E+02 ± 2.24725E+02	5.96303E+02 ± 3.57869E+02	7.12948E+02 ± 1.56802E+02	3.55165E+03 ± 4.72872E+03

Fig. 8 Results of Friedman rank test for CEC 2017 test functions with 10D



of ranks are presented in Table 17. At 95% confidence level, the obtained critical value (F-statistic value) and Friedman statistics.

(F-Score) are reported in Table 17. Table 17 shows that, at a 95% confidence level, the calculated F-Score value (13.149847095) is greater than the F-statistics value (2.638), which indicates that the average ranks obtained from the mean results of all the test functions are statistically different. It is also observed from Table 17 and Fig. 8 that the mean rank difference is superior to the calculated CD value (1.496560163). Hence, it is summarized that ImBSA presents outstanding performance on CEC 2017 special session and competition test functions than its competitors considered in this study.

Application of ImBSA to Real-World Problem

In this part, six recognized engineering test problems [49] are tested with the proposed updated version of BSA, namely, (i) frequency modulation sounds parameter identification problem, (ii) Lennard–Jones potential problem, (iii) Tersoff potential function minimization problem, (iv)

speed reducer design problem, (v) I-beam design problem, (vi) cantilever beam design problem, and (vii) speed reducer design engineering problem. The following can be summarized in the numerical engineering test problem:

Frequency Modulation Sounds Parameter Identification Problem [49]

The sound wave is a widespread optimization issue and plays a significant part in numerous current music systems. Frequency-modulation (FM) fusion is skillful in producing unique sounds but is not always properly regulated. A small number of factors can play a crucial role in delivering an extensive range of sound tones. Noises that are comparable to the goal sounds can be produced; therefore, using ImBSA for FM synthesis, the FM matching FM approach is described to identify optimum parameter values. Initially,

Table 16 The number of occasions where ImBSA is superior to, inferior to, and similar to the compared algorithms on 10-dimension CEC 2017 benchmark functions

Algorithms	Superior to (ImBSA > algorithms)	Inferior to (ImBSA < algorithms)	Equal to (ImBSA = algorithms)
GWO-BSA	29	0	0
e-SOSBSA	20	9	0
BS-DE	29	0	0
LFBSA	29	0	0
IBS	29	0	0
hDEBSA	29	0	0

Table 17 Friedman test on CEC 2017 real-world benchmark functions with 10 dimensions of GWO-BSA, e-SOSBSA, BS-DE, LFBSA, IBS, hDEBSA, and ImBSA based on their mean performances

Algorithms	Average rank	Variance of rank	Average rank difference w.r.t. average rank
ImBSA	1.310344828	0.221674877	
GWO-BSA	4.689655172	1.793103448	3.379310345
e-SOSBSA	3.793103448	5.955665025	2.482758621
BS-DE	4.586206897	2.036945813	3.275862069
LFBSA	4.965517241	2.320197044	3.655172414
IBS	4.275862069	2.635467980	2.965517241
hDEBSA	4.344827586	4.305418719	3.034482759
F-Score			13.149847095
Critical difference (CD)			1.496560163
F-statistics			2.638

a set of parameters is established by ImBSA, and the FM synthesizer generates the sound.

The distances between the target and synthesized sounds are treated as the fitness value in this task (or objective function value). This engineering benchmark problem is a tough multimodal challenge. Six of the sound waves are the dimension of this problem. The objective function has the following definition:

$$\text{Minimize } f(y) = \sum_{t=0}^{100} (y(t) - y_0(t))^2 \tag{17}$$

where Eqs. (18) and (19) give the primary sound wave and target sound.

$$y(t) = a_1 \sin(\omega_1 t \theta + a_2 \sin(\omega_2 t \theta + a_3 \sin(\omega_3 t \theta))) \tag{18}$$

$$y_0(t) = 1.0 * \sin(5.0 * t \theta + 1.5 * \sin(4.8 * t \theta + 2.0 * \sin(4.9 * t \theta))) \tag{19}$$

Six variables are $y = \{a_1, \omega_1, a_2, \omega_2, a_3, \omega_3\}$, and the domain of these variables is $[-6.4, 6.35]$. Here $\theta = 2\pi/100$.

Lennard–Jones Potential Problem [49]

The potential challenge of Lennard–Jones (LJ) is to minimize the molecular power potential of the pure LJ cluster. An icosahedral core and a combination of surface grid points are present in the grid structure of the LJ cluster. The algorithm ImBSA is implemented to make the particle less energy-efficient to adapt its molecular structure to construct the atoms. This hypothetical challenge of multimodal optimization energy reduction involves an endless number of local minimum levels. The Cartesian coordinates provide the cluster setup of “n” atoms in LJ problems.

$$\vec{p}_i = \{\vec{x}_i, \vec{y}_i, \vec{z}_i\}, i = 1, 2, 3, \dots, N \tag{20}$$

The following are given for the mathematical creation of Lennard–Jones’ atom “N” pair potential:

$$V_N(p) = \sum_{i=1}^{N-1} \sum_{j=i+1}^N (r_{ij}^{-12} - 2.r_{ij}^{-6}) \tag{21}$$

Where, $r_{ij} = \|\vec{p}_j - \vec{p}_i\|$ (22)

And the gradient is

$$\nabla_j V_N(p) = -12 \sum_{i=1, i \neq j}^{N-1} (r_{ij}^{-14} - 2.r_{ij}^{-8}) (\vec{p}_j - \vec{p}_i), j = 1, 2, 3, \dots, N \tag{23}$$

Ten atomic issues were taken into account, and a total of 30 variables were added. Any allowing and translating the $\{p_j\}$ order lead to the identical functional value, as $V_N(p)$ only depends on the pair distances. The search space of first three variables are $(0, 4)$, $(0, 4)$, and $(0, \pi)$, and for other variables are: $\left[-4 - \frac{1}{4} \left\lceil \frac{i-4}{3} \right\rceil, 4 + \frac{1}{4} \left\lceil \frac{i-4}{3} \right\rceil\right]$ where $\lceil r \rceil$ is the nearest integer of r.

Tersoff Potential Function Minimization Problem for Model Si(B) [49]

The total Tersoff potential energy function of the system is the sum of individual potentials of atoms and is defined as

$$f(\vec{X}_1, \vec{X}_2, \vec{X}_3, \dots, \vec{X}_N) = E_1(\vec{X}_1, \vec{X}_2, \vec{X}_3, \dots, \vec{X}_N) + \dots + E_N(\vec{X}_1, \vec{X}_2, \vec{X}_3, \dots, \vec{X}_N) \tag{24}$$

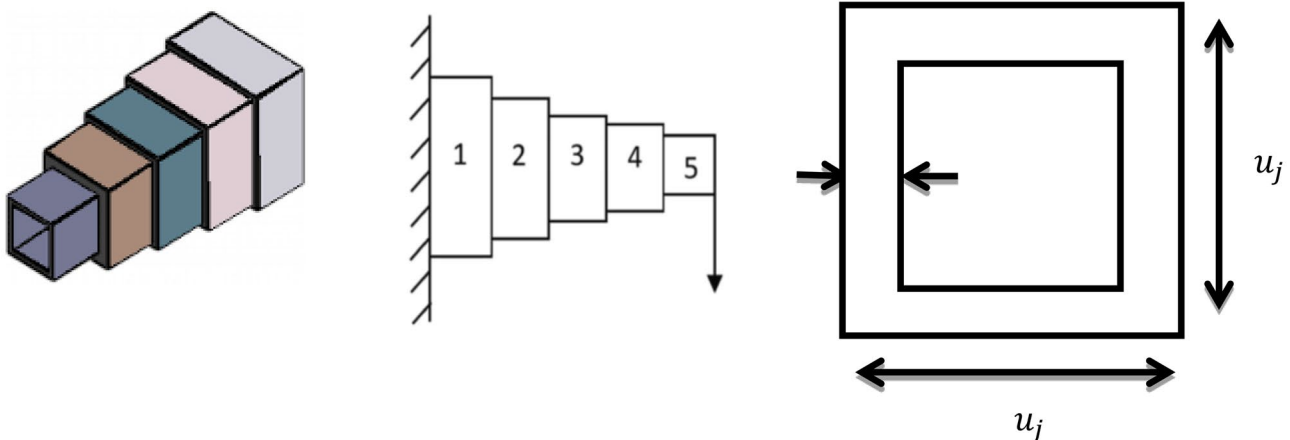


Fig. 9 Cantilever beam design problem

The Tersoff potential of individual atoms can be formally defined as follows:

$$E_i = \frac{1}{2} \sum_{i \neq j} f_c(r_{ij})(V_R(r_{ij}) - B_{ij}V_A(r_{ij})), \forall i \tag{25}$$

Here r_{ij} is the distance between atoms i and j , V_R is a repulsive term, V_A is an attractive term, $f_c(r_{ij})$ is a switching function, and B_{ij} is a many-body term that depends on the positions of atoms i and j and the neighbors of atom i . The term B_{ij} is given by

$$B_{ij} = \left(1 + \gamma^{n_1} \xi_{ij}^{n_1}\right)^{-1/2n_1} \tag{26}$$

In Eq. (26), n_1 and γ are known fitted parameters, and the term ξ_{ij} for atoms, i and j (i.e., for bond), are given by:

$$\xi_{ij} = \sum_{k \neq i} f_c(r_{ik})g(\theta_{ijk})\exp\left(\lambda_3^3 - (r_{ij} - r_{ik})^3\right) \tag{27}$$

In Eq. (27), the term θ_{ijk} is the bond angle between bonds ij and ik , and the function g is given by:

$$g(\theta_{ijk}) = 1 + c^2/d^2 - c^2/\left(d^2 + (h - \cos(\theta_{ijk}))^2\right) \tag{28}$$

The quantities λ_3 and c, d, h which appear in Eqs. (27) and (28) are also known as fitted parameters. The terms $V_R(r_{ij})$, $V_A(r_{ij})$ and the switching function $f_c(r_{ij})$, appear in Eq. (25) are given by:

$$V_R(r_{ij}) = Ae^{-\lambda_1 r_{ij}} \tag{29}$$

$$V_A(r_{ij}) = Be^{-\lambda_2 r_{ij}} \tag{30}$$

$$f_c(r_{ij}) = \begin{cases} 1 & , r_{ij} \leq R - D \\ \frac{1}{2} - \frac{1}{2} \sin\left(\frac{\pi(r_{ij}-R)}{D}\right) & , R - D < r_{ij} < R + D \\ 0 & , r_{ij} \geq R + D \end{cases} \tag{31}$$

where A, B, λ_1, R, D , and λ_2 are given fitted parameters.

Thus, the cost function now can be redefined as:

$$f(x) = E_1(x) + E_2(x) + \dots + E_N(x), x \in \Omega \tag{32}$$

In general, for N atoms, the no. of unknown variables is $n = 3 \times N - 6$. Now the cluster X of N atoms can be redefined as

$$x = \{x_1, x_2, x_3, \dots, x_n\}, x \in IR^{3N-6} \tag{33}$$

The search region for Si(B) model of Tersoff potential is

$$\Omega = \{ \{x_1, x_2, x_3, \dots, x_n\} : -4.25 \leq x_1, x_i \leq 4.25; i = 4, 5, \dots, n; 0 \leq x_2 \leq 4; 0 \leq x_3 \leq \pi \}.$$

The number of atoms is therefore regarded to be 10, with the total no variables being 30. Reference [49] provides a complete overview of this problem.

Spread Spectrum Radar Polyphase Code Design Problem[49]

The spectrum radar polyphase code design challenge is one of the most frequent design optimization issues for global optimization techniques. The choice of proper waveform must be given close attention when a radar system is designed using pulse compression. There are several modulation methods of radar pulse that allow pulse compression. Polyphase codes are appealing because they give reduced side lobes in the compressed signals and make digital processing techniques easier to apply. This synthesis is based upon the features of the aperiodic autocorrelation function, assuming that the receiver is being consistently radar pulse processed. The problem has constant variables and a min–max nonlinear non-convex optimization problem with several local solutions. This problem is mathematically formed as follows:

$$\text{Minimize } f(X) = \max\{\phi_1(x), \phi_2(x), \phi_3(x), \dots, \phi_{2m}(x)\} \tag{34}$$

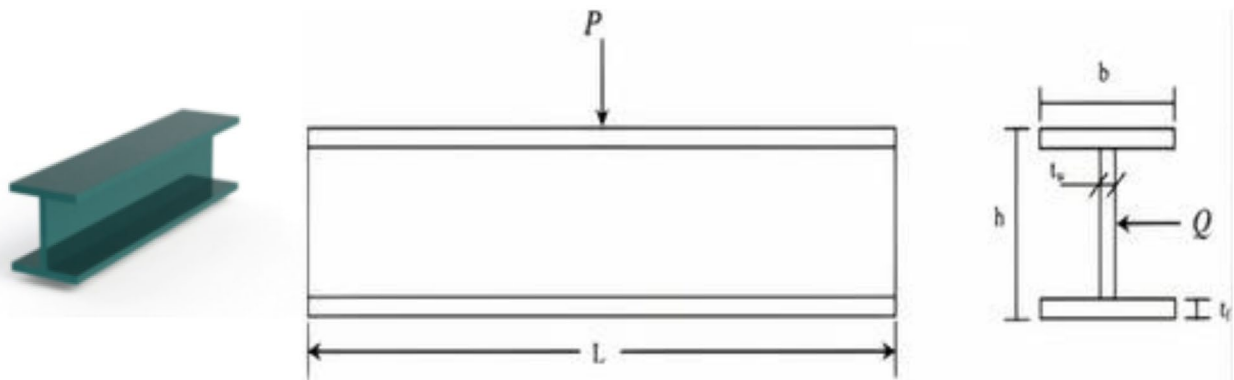


Fig. 10 I-beam design problem

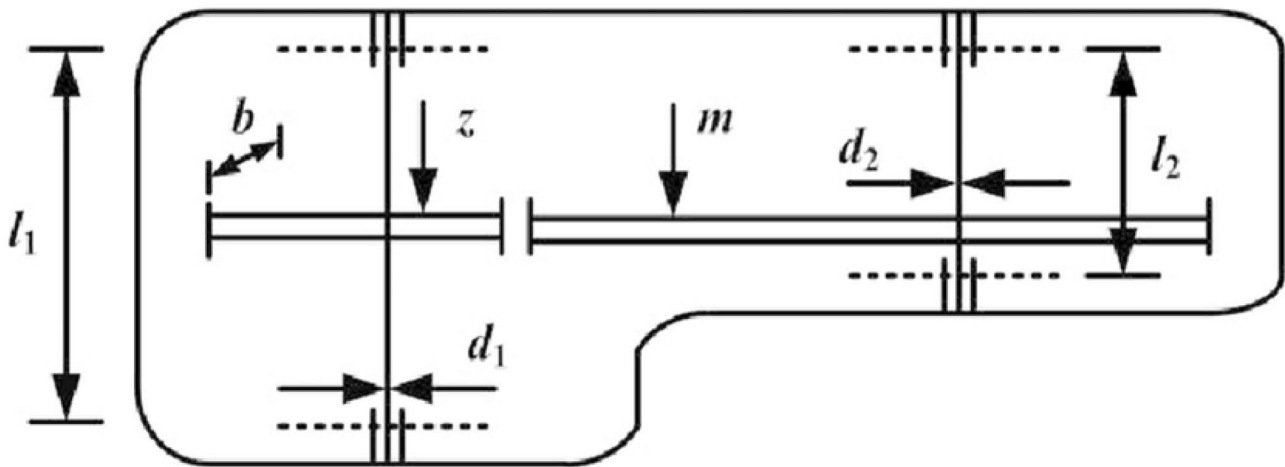


Fig. 11 Speed reducer design problem

where $X = \{(x_1, x_2, x_3, \dots, x_4) \in R^D \mid 0 \leq x_j \leq 2\pi, j = 1, 2, 3, \dots, D\}$, and $m = 2D - 1$, with

$$\phi_{2i-1}(X) = \sum_{j=i}^D \cos\left(\sum_{k=|2i-j-1|+1}^j X_k\right), i = 1, 2, 3, \dots, D \tag{35}$$

$$\phi_{2i}(X) = 0.5 + \sum_{j=i+1}^D \cos\left(\sum_{k=|2i-j|+1}^j X_k\right), i = 1, 2, 3, \dots, D - 1 \tag{36}$$

$$\phi_{m+i}(X) = -\phi_i(X), i = 1, 2, 3, \dots, m; \tag{37}$$

In this context, the objective is to optimize (minimize) the most extensive module among the autocorrelation function linked to the compressed radar pulse at optimum recipient output. At the same time, the variables indicate symmetry phase differences. This is a problem of NP-hard optimization. Due to its objective function, the problem may be characterized as a class of continuous minimal global optimization problems. The problem includes 20 variables, and the variable domain is $[0, 2\pi]$.

Cantilever Beam Design Engineering Problem [49]

The topic is the optimization of the weight of a square cross-sectional cantilever beam depicted in Fig. 9. At node 1, the beam is firmly held, while at node 5, a vertical force exists. The five design variables include the heights (or widths) of the beam and the thickness of the fixed elements of the beams. The analysis of the problem is as follows:

$$\text{Minimize } f_5(\vec{u}) = 0.0624(u_1 + u_2 + u_3 + u_4 + u_5) \tag{38}$$

Subject to,

$$g_1(\vec{u}) = \frac{61}{u_1^3} + \frac{37}{u_2^3} + \frac{19}{u_3^3} + \frac{7}{u_4^3} + \frac{1}{u_5^3} - 1 \leq 0 \tag{39}$$

$$\text{Where } 0.01 \leq u_j \leq 100 (j = 1, 2, 3, 4, 5), \vec{u} = (u_1, u_2, u_3, u_4, u_5) \tag{40}$$

Table 18 Parameter estimation for frequency-modulated (FM) sound waves

Algorithms	Best	Medians	Worst	Mean	SD
ImBSA	1.83e+001	2.17e+001	2.81e+001	2.13e+001	2.01e+000
mISOS	1.97e+001	2.60e+001	2.90e+001	2.58e+001	2.06e+000
ISOS	2.41e+001	2.72e+001	2.92e+001	2.71e+001	1.28e+000
SOS	2.44e+001	2.62e+001	2.79e+001	2.64e+001	9.89e-001
ASOS_BF1	2.39e+001	2.64e+001	2.78e+001	2.63e+001	1.11e+000
ASOS_BF2	1.95e+001	2.64e+001	2.82e+001	2.61e+001	1.93e+000
ASOS_BF1&2	2.15e+001	2.61e+001	2.85e+001	2.61e+001	1.41e+000
BSA	2.37e+001	2.82e+001	2.99e+001	2.79e+001	1.45e+000
ABSA	2.56e+001	2.83e+001	3.05e+001	2.82e+001	1.42e+000
DE	2.54e+001	2.65e+001	2.87e+001	2.65e+001	8.92e-001

Table 19 Lennard–Jones potential problem

Algorithms	Best	Median	Worst	Mean	SD
ImBSA	−6.17e + 000	−4.35e + 000	−2.98e + 000	−4.85e + 000	7.19e − 001
mISOS	−6.04e + 000	−4.23e + 000	−2.93e + 000	−4.37e + 000	7.76e − 001
ISOS	−6.28e + 000	−3.40e + 000	−2.11e + 000	−3.48e + 000	9.39e − 001
SOS	−6.42e + 000	−4.03e + 000	−2.47e + 000	−4.12e + 000	8.74e − 001
ASOS_BF1	−4.41e + 000	−3.29e + 000	−2.43e + 000	−3.29e + 000	5.40e − 001
ASOS_BF2	−5.42e + 000	−3.31e + 000	−2.30e + 000	−3.42e + 000	7.56e − 001
ASOS_BF1&2	−3.46e + 000	−2.56e + 000	−2.04e + 000	−2.59e + 000	3.53e − 001
BSA	−4.37e + 000	−2.51e + 000	−2.03e + 000	−2.71e + 000	5.81e − 001
ABSA	−4.37e + 000	−2.51e + 000	−2.03e + 000	−2.71e + 000	5.81e − 001
DE	−3.91e + 000	−2.78e + 000	−2.14e + 000	−2.84e + 000	4.28e − 001

I-beam Design Engineering Problem [46]

The ImBSA algorithm tested a new design problem, i.e., the problem of I-beam design, incorporating four variables, in addressing actual engineering difficulties. The goal of I-beam design optimization is to decrease vertical I-beam deflection, as seen in Fig. 10. With the I-beam design optimization issue, cross-sectional areas and stress limits are concurrently fulfilled with specified loads. The following are the formulation of the I-beam optimization problem:

$$\text{Minimize } f_6(\vec{u}) = \frac{5000}{\frac{u_3(u_1 - 2u_4)}{12} + \frac{u_2 u_4^3}{6} + 2u_2 u_4 \left(\frac{u_1 - u_4}{2}\right)^2} \quad (41)$$

Subject to,

$$g_1(\vec{u}) = 2u_2 u_3 + u_3(u_1 - 2u_4) - 300 \leq 0 \quad (42)$$

$$g_2(\vec{u}) = \frac{18u_1 \times 10^4}{u_3(u_1 - 2u_4)^3 + 2u_2 u_3(4u_4^2 + 3u_1(u_1 - 2u_4))} + \frac{15u_2 \times 10^3}{(u_1 - 2u_4)u_3^3 + 2u_3 u_2^3} - 56 \leq 0 \quad (43)$$

Where, $10 \leq h \leq 80, 10 \leq u_2 \leq 50, 0.9 \leq u_3 \leq 5, \text{ and } 0.9 \leq u_4 \leq 5,$ (44)

$$\vec{u} = (u_1, u_2, u_3, u_4) = (h, b, t_w, t_f)$$

Speed Reducer Design Engineering Problem

The proposed ImBSA is employed to determine the speed reducer design problem [46]. Various components of this problem are the number of teeth on pinion (z), the module of teeth (m), face width (b), the diameter of shaft 1 (d_1), and diameter of shaft 2 (d_2), length of shaft 1 between bearings (l_1), and length of shaft 2 between bearings (l_2) as shown seen in Fig. 11. The objective of this engineering benchmark design optimization problem is to minimize the total weight of the speed reducer. The constraints included in this problem are bending stress, surface stress, and transverse deflections. The analytical expressions of this problem are as follows:

$$\begin{aligned} \text{Minimize } f_7(\vec{u}) = & 0.7854u_1 u_2^2 (3.3333u_3^2 + 14.933u_3 - 43.0934) \\ & - 1.508u_1 (u_6^2 + u_7^2) + 7.4777(u_6^3 + u_7^3) \\ & + 0.7854(u_4 u_6^2 + u_5 u_7^2), \end{aligned} \quad (45)$$

Subject to

$$g_1(\vec{u}) = \frac{27}{u_1 u_2^2 u_3} - 1 \leq 0 \quad (46)$$

Table 20 Tersoff potential function minimization problem

Algorithms	Best	Median	Worst	Mean	SD
ImBSA	−1.99e + 001	−1.79e + 001	−1.73e + 001	−1.68e + 001	2.52e + 000
mISOS	−1.93e + 001	−1.36e + 001	−1.02e + 001	−1.40e + 001	2.22e + 000
ISOS	−1.59e + 001	−1.21e + 001	−8.77e + 000	−1.28e + 001	1.70e + 000
SOS	−1.48e + 001	−1.19e + 001	−8.56e + 000	−1.18e + 001	1.69e + 000
ASOS_BF1	−1.77e + 001	−1.16e + 001	−4.66e + 000	−1.15e + 001	2.48e + 000
ASOS_BF2	−1.71e + 001	−1.12e + 001	−8.39e + 000	−1.15e + 001	1.92e + 000
ASOS_BF1&2	−1.60e + 001	−1.11e + 001	−8.35e + 000	−1.14e + 001	1.89e + 000
BSA	−2.03e + 001	−1.32e + 001	−1.01e + 001	−1.33e + 001	2.09e + 000
ABSA	−1.85e + 001	−1.35e + 001	−1.10e + 001	−1.37e + 001	1.73e + 000

Table 21 Spread spectrum radar polyphase problem

Algorithms	Best	Medians	Worst	Mean	SD
ImBSA	1.59e+000	2.13e+000	2.65e+000	2.17e+000	2.19e−001
mISOS	1.76e+000	2.24e+000	2.59e+000	2.25e+000	2.27e−001
ISOS	2.17e+000	2.39e+000	2.59e+000	2.39e+000	1.19e−001
SOS	2.07e+000	2.36e+000	2.65e+000	2.36e+000	1.63e−001
ASOS_BF1	2.04e+000	2.33e+000	2.59e+000	2.31e+000	1.44e−001
ASOS_BF2	2.02e+000	2.34e+000	2.54e+000	2.32e+000	1.49e−001
ASOS_BF1&2	1.90e+000	2.33e+000	2.66e+000	2.35e+000	1.70e−001
BSA	1.81e+000	2.33e+000	2.58e+000	2.27e+000	2.16e−001
ABSA	1.90e+000	2.32e+000	2.60e+000	2.30e+000	1.64e−001
DE	2.01e+000	2.33e+000	2.59e+000	2.31e+000	1.63e−001

$$g_2(\vec{u}) = \frac{397.5}{u_1 u_2^2 u_3^2} - 1 \leq 0 \tag{47}$$

$$g_8(\vec{u}) = \frac{5u_2}{u_1} - 1 \leq 0 \tag{53}$$

$$g_3(\vec{u}) = \frac{1.93u_4^3}{u_2 u_6^4 u_3} - 1 \leq 0 \tag{48}$$

$$g_9(\vec{u}) = \frac{u_1}{12u_2} - 1 \leq 0 \tag{54}$$

$$g_4(\vec{u}) = \frac{1.93u_5^3}{u_2 u_7^4 u_3} - 1 \leq 0 \tag{49}$$

$$g_{10}(\vec{u}) = \frac{1.5u_6 + 1.9}{u_4} - 1 \leq 0 \tag{55}$$

$$g_5(\vec{u}) = \frac{\left[\left(\frac{745u_4}{u_2 u_3} \right)^2 + 16.9 \times 10^6 \right]^{\frac{1}{2}}}{110. u_6^3} - 1 \leq 0 \tag{50}$$

$$g_{11}(\vec{u}) = \frac{1.1u_7 + 1.9}{u_5} - 1 \leq 0 \tag{56}$$

$$g_6(\vec{u}) = \frac{\left[\left(\frac{745u_5}{u_2 u_3} \right)^2 + 157.5 \times 10^6 \right]^{\frac{1}{2}}}{85.0u_7^3} - 1 \leq 0 \tag{51}$$

where $2.6 \leq u_1 \leq 3.6$, $0.7 \leq u_2 \leq 0.8$, $17 \leq u_3 \leq 28$, $7.3 \leq u_4 \leq 8.3$, $7.3 \leq u_5 \leq 8.3$, $2.9 \leq u_6 \leq 3.9$, and $5.0 \leq u_7 \leq 5.5$, $\vec{u} = (u_1, u_2, u_3, u_4, u_5, u_6, u_7) = (z, m, b, l_1, l_2, d_1, d_2)$.

$$g_7(\vec{u}) = \frac{u_2 u_3}{40} - 1 \leq 0 \tag{52}$$

Results and Discussions on Real-World Problems

The ImBSA has calculated 1000 function evaluations of 50 population sizes and 25 runs for the first four tasks. Tables 18, 19, 20, and 21 describe the result of four issues,

Table 22 Comparison of performance result of the cantilever beam design problem

Algorithms	Optimum decision variables					Objective function value
	u_1	u_2	u_3	u_4	u_5	
ImBSA	6.0157385	5.3090857	4.4927465	3.5019894	2.1534867	1.33991812032
e-SOSBSA	6.0160377	5.3091474	4.4943187	3.5014931	2.1526627	1.3399564
m-SCA	6.0089	5.3049	4.5023	3.5077	2.1504	1.33999
SCA	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
CS	6.0089	5.3049	4.5023	3.5077	2.1504	1.33999
GCA (I)	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
GCA (II)	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
MMA	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
CONLIN	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400

Table 23 Comparison of performance result of the I-beam design problem

Algorithms	Optimum decision variables				Objective function value
	u_1	u_2	u_3	u_4	f_{min}
ImBSA	80	50	1.7646138	5.0000000	0.007100487920316
e-SOSBSA	80	50	1.7647059	5.0000000	0.0071005
Cauchy-GWO	80	50	0.9	2.2599	0.013075
ARSM	80	37.05	1.71	2.31	0.01570
Improved ARSM	79.99	48.42	0.9	2.40	0.1310
CS	80	50	0.9	2.3216715	0.0130747
GWO	80	42.8154	0.9	2.7179	0.013202

i.e., the optimal solution. The findings of the problems mentioned above are derived from the reference [49].

The result achieved is compared to ISOS, SOS, ASOS_BF1, ASOS_BF2, ASOS_BF1&2, BSA, ABSA, DE, and ImBSA for the frequency-modulated (FM) parameter estimation problem. Table 18 contains the best, median, worst, median, and SD obtained by ImBSA and some other algorithms on this problem. Table 18 shows that ImBSA receives the best solution. Therefore, the findings of Table 18 suggest that the performance of the ImBSA for this problem is best compared to other methods.

The outcomes achieved by the proposed algorithm are compared with ISOS, SOS, ASOS BF1, ASOS BF2, ASOS BF1&2, BSA, ABSA, DE, and ImBSA for Lennard–Jones potential problem. Table 19 presents the best, median, worst, average, and SD showing the performance outcomes of this

problem by ImBSA and aforesaid algorithms. Table 19 displays that ImBSA obtains the best, median, and mean results. Thus, one can readily sum up from the findings that the performance of the ImBSA is the best compared to other techniques.

The findings obtained are compared with other algorithms like ISOS, SOS and ASOS BF1, ASOS BF2, ASOS BF1&2, BSA and ABSA, and ImBSA for Tersoff’s Si(B) potential. Table 20 contains the best, median, worst, mean, and SD performance outcomes of this optimization task. Table 20 shows that the BSA obtains the best answer, although ImBSA obtains the best median and mean outcomes. The results suggest that the ImBSA version demonstrates a superior way than some algorithms compared here.

The comparison table of the results obtained by ISOS, SOS, ASOS BF1, ASOS BF2, ASOS BF1&2, BSA,

Table 24 Comparison of performance result of the speed reducer problem

Algorithm	u_1	u_2	u_3	u_4	u_5	u_6	u_7	f_{min}
ImBSA	3.5008989	0.7000000	17.0000000	7.3007000	7.7159679	3.3538547	5.2846039	2994.4710501
e-SOSBSA	3.5000000	0.7000000	17.0000000	7.3000000	7.7153199	3.3502147	5.2866545	2994.4710661
SC-GWO	3.50064	0.7	17	7.30643	7.80617	3.35034	5.28694	2996.9859
PSO	3.58147	0.7	17.8282	7.98445	7.82083	3.15398	5.1873	3005.3248
GWO	3.6	0.8	28	7.3	8.3	2.9	5.0	3020.2331
SCA	3.51889	0.7	17	7.3	8.3	3.35899	5.30519	3028.8657
wPSO	3.50662	0.7	17	7.44735	7.88468	3.2725	5.38998	3003.7983
CS	3.50150	0.7	17	7.6050	7.8181	3.3520	5.2875	3000.9810
Ray and Saini	3.51418	0.700005	17	7.497343	7.8346	2.9018	5.0022	2732.9006 (infeasible)
Akhtar et al	3.50612	0.700006	17	7.549126	7.85933	3.36558	5.289773	3008.08
Ku et al	3.6	0.7	17	7.3	7.8	3.4	5	2876.1176 (infeasible)
Montes and Coello	3.50616	0.700831	17	7.46018	7.962143	3.3629	5.3090	3025.005
mGWO	3.50128	0.7	17	7.34965	7.80177	3.35087	5.28712	2997.7748
wGWO	3.50008	0.7	17	7.3193	7.81168	3.35072	5.28692	2997.085
m-SCA	3.52394	0.7	17	7.3	7.8	3.36280	5.32467	3033.2845
OBSCA	3.00576	0.72755	21.8423	7.30835	8.15455	3.36452	5.25164	3027.5130
MFO	3.59093	0.70554	19.7972	8.08267	7.84181	3.70621	5.48167	3836.2164
WOA	3.52111	0.7	17	7.3	7.8	3.35021	5.29533	3010.1480
SSA	3.50031	0.7	17	7.80001	7.85001	3.35247	5.2867	3002.5678
ISCA	3.50081	0.7	17	7.3	7.8	3.35129	5.28698	2997.1295
Chaotic SSA	3.50031	0.7	17	7.80001	7.85001	3.35247	5.2867	3002.5678

ABSA, DE, and ImBSA for the radar polyphase code design problem is presented in Table 21. Table 21 shows that ImBSA obtains the best solution, while ASOS BF2 gets the worst outcome. So, by observing Table 21, it can be said that the ImBSA works better than any other approach. This suggests ImBSA's overall superiority to solve this problem. The explanation above indicates that the performance of the ImBSA is acceptable to solve this problem.

The proposed ImBSA has been employed to solve the cantilever beam design with the same configuration as the reference [46]. The optimal output attained is shown in Table 22, and results other than ImBSA are derived from [46]. In comparison with m-SCA, SCA, CS, GCA(I), GCA(II), MMA, CONLIN, and e-SOSBS, the optimal performance of the suggested ImBSA is better. Compared to the other optimizer offered in this study, the comparative research shown in Table 22 shows ImBSA's efficient performance capability.

Cauchy-GWO, ARSM, improved ARSM, CS, GWO, and e-SOSBSA were solved I-beam design engineering optimization problems previously. The suggested ImBSA research is also employed to solve this significant optimization issue with the same parameter configuration used in reference [46]. The best outcome achieved by ImBSA is presented in Table 23, and the results other than ImBSA are taken from [46]. The optimized values of the four design variables are 80, 50, 1.7646138, and 5.0000000, and an ideal I-beam vertical deflection is 0.0071004879203201. Table 23 demonstrates that ImBSA handles the problem with improved performance in the design of beams compared to other algorithms given in this table.

The speed reducer design problem has previously been optimized with a different algorithm which can be seen in [46]. Table 24 provides a summary of the best solution for the recorded methods. The results were achieved with the same parameter setting as in reference [46] to compare the performance of ImBSA. The optimum output is compared with e-SOSBSA, SC-GWO, CS, SCA, PSO, wPSO, GWO, Ray and Saini, mGWO, wGWO, m-SCA, OBSCA, SSA, MFO, WOA, ISCA, Chaotic SSA, Akhtar et al., Ku et al., Montes, and Coello. The results of these methods are taken from [46]. The optimum values of all decision variables are 3.5008989, 0.7000000, 17.0000000, 7.3007000, 7.7159679, 3.3538547, and 5.2846039, and the optimum weight of the speed reducer is 2994.4710501. The results' comparison with other algorithms is also shown in the same table, demonstrating the proposed ImBSA over different algorithms.

It is evident from the preceding discussion, findings, and reviews that the quality, efficiency, and resilience of ImBSA have improved to address real-life challenges. In addition, ImBSA may also be used for practical applications as an alternative optimizer.

Conclusion

In this study, a new mutation strategy of the BSA has been proposed, which exploits the searchability of the algorithm in a better way. A new modified parameter scheme has been considered in ImBSA. An ensemble of mutation strategies with a multi-population concept has also been incorporated into this study. Incorporating the concepts above, a novel BSA variant, called multi-population-based self-adaptive BSA (ImBSA, in short), has been proposed in this study. The extensive performance of ImBSA on CEC 2014 and CEC 2017 test functions has been tested by comparing with several algorithms found in the literature. The comparison results show that ImBSA outperforms some other efficient algorithms. For statistical analysis, a nonparametric Friedman test in which a pairwise post hoc Bonferroni-Dunn test with a 95% confidence level is utilized to highlight the significant difference of performance of ImBSA among other compared algorithms. The statistical analysis also confirms that ImBSA is statistically more effective than the compared algorithms.

The study addresses the efficiency of the proposed algorithm employing six problems of engineering design optimization. Results on technical challenges also approve that ImBSA has been improved significantly compared to different optimizers in the literature.

Based on the findings of experimental results, statistical testing, and real-world applications, some concluding remarks are given below:

- Compared with conventional BSA, the improved stability of BSA's exploitation and exploration has been established and improved. With auto-adaptive mutation rate and mixrate parameters, in the suggested ImBSA, the stability and convergence speed have been improved.
- The numerical findings of the CEC 2014 and CEC 2017 benchmark suites demonstrate that the proposed ImBSA has increased search capacity compared to some of the competitors.
- The convergence curve drawn between the fitness values at each generation determines the stability within the suggested methodology for exploitation and identifies the optimal value throughout the process of searching.
- A convergence and statistical validation study of the data confirms the improved performance by ImBSA when addressing issues with optimization compared to traditional BSA and other comparative optimization methods.
- To determine the numerical findings, the challenge of technical test problems, including diverse levels of complexity and space of research, also promotes the efficiency of the suggested ImBSA.

Limitation and Future Direction

There is hardly any study without limitations. Although we have tried to enhance the efficiency of the BSA, this study has several limitations. Some of these limitations and future directions are given below:

- The adaptive parameter setting based control parameters value has only been suggested here. But this can be done in many other ways such as the fuzzy adaptive, neighborhood-based, concept of the topology of the population, and fitness distance-based. These are some future directions of the current work.
- The algorithm's stochastic analysis, such as convergence analysis and stability analysis, may also be analyzed so that the proposed algorithm may have a strong mathematical base.
- The best-guided search-direction matrix is used in the proposed method, but this can be further improved in different ways.
- This method is applied only on single-objective constrained and unconstrained optimization problems which may be extended to the multi-objective scenario.
- This approach may be used to determine numerous optimization problems from academia and industry.

Acknowledgements The authors would like to thank the editor and anonymous reviewers for valuable feedback towards the improvement of the manuscript.

Declarations

Conflict of Interest The authors declare no competing interests.

References

1. Civicioglu P. Backtracking search optimization algorithm for numerical optimization problems. *Appl Math Comput.* 2013;219:8121–44. <https://doi.org/10.1016/j.amc.2013.02.017>.
2. Nama S, Saha AK. A novel hybrid backtracking search optimization algorithm for continuous function optimization. *Decis Sci Lett.* 2019;8:163–74. <https://doi.org/10.5267/j.dsl.2018.7.002>.
3. Nama S, Saha AK, Ghosh S. A new ensemble algorithm of differential evolution and backtracking s algorithm with adaptive control parameter for function optimization. *Int J Ind Eng Comput.* 2016;7:323–38. <https://doi.org/10.5267/j.ijiec.2015.9.003>.
4. Nama S, Saha AK, Ghosh S. Improved backtracking search algorithm for pseudo dynamic active earth pressure on retaining wall supporting c- Φ backfill. *Appl Soft Comput J.* 2017;52:885–97. <https://doi.org/10.1016/j.asoc.2016.09.037>.
5. Chen D, Zou F, Lu R, Li S. Backtracking search optimization algorithm based on knowledge learning. *Inf Sci (Ny).* 2019;473:202–26. <https://doi.org/10.1016/j.ins.2018.09.039>.
6. Tian Z. Backtracking search optimization algorithm-based least square support vector machine and its applications. *Eng Appl Artif Intell.* 2020;94: 103801. <https://doi.org/10.1016/j.engappai.2020.103801>.
7. Zhang Y, Ma M, Jin Z. Backtracking search algorithm with competitive learning for identification of unknown parameters of photovoltaic systems. *Expert Syst Appl.* 2020;160: 113750. <https://doi.org/10.1016/j.eswa.2020.113750>.
8. Zhang Y, Huang C, Jin Z. Backtracking search algorithm with reusing differential vectors for parameter identification of photovoltaic models. *Energy Convers Manag.* 2020;223: 113266. <https://doi.org/10.1016/j.enconman.2020.113266>.
9. Caldeira RH, Gnanavelbabu A, Vaidyanathan T. An effective backtracking search algorithm for multi-objective flexible job shop scheduling considering new job arrivals and energy consumption. *Comput Ind Eng.* 2020;149: 106863. <https://doi.org/10.1016/j.cie.2020.106863>.
10. Emami H, Sharifi AA. An improved backtracking search optimization algorithm for cubic metric reduction of OFDM signals. *ICT Express.* 2020;6:258–61. <https://doi.org/10.1016/j.icte.2020.03.001>.
11. Zhang Y, Jin Z, Zhao X, Yang Q. Backtracking search algorithm with Lévy flight for estimating parameters of photovoltaic models. *Energy Convers Manag.* 2020;208: 112615. <https://doi.org/10.1016/j.enconman.2020.112615>.
12. Zhang Y. Backtracking search algorithm with specular reflection learning for global optimization. *Knowl Based Syst.* 2021;212: 106546. <https://doi.org/10.1016/j.knosys.2020.106546>.
13. Duan H, Luo Q. Adaptive backtracking search algorithm for induction magnetometer optimization. *IEEE Trans Magn.* 2014;50:1–6. <https://doi.org/10.1109/TMAG.2014.2342192>.
14. Guha D, Roy P, Banerjee S. Quasi-oppositional backtracking search algorithm to solve load frequency control problem of interconnected power system. *Iran J Sci Technol Trans Electr Eng.* 2020;44:781–804. <https://doi.org/10.1007/s40998-019-00260-0>.
15. Jin YF, Yin ZY. Enhancement of backtracking search algorithm for identifying soil parameters. *Int J Numer Anal Methods Geomech.* 2020;44:1239–61. <https://doi.org/10.1002/nag.3059>.
16. Yan X, Han Y, Gu X. An improved discrete backtracking searching algorithm for fuzzy multiproduct multistage scheduling problem. *Neurocomputing.* 2020;398:153–65. <https://doi.org/10.1016/j.neucom.2020.02.066>.
17. Saha A, Nama S, Ghosh S. Application of HSOS algorithm on pseudo-dynamic bearing capacity of shallow strip footing along with numerical analysis. *Int J Geotech Eng.* 2019. <https://doi.org/10.1080/19386362.2019.1598015>.
18. Lohar G, Sharma S, Saha AK, Ghosh S. Optimization of geotechnical parameters used in slope stability analysis by metaheuristic algorithms. In: Mandal J, Mukhopadhyay S, Roy A, editors. *Lecture Notes in Networks and Systems.* Singapore: Springer Science and Business Media; 2021. p. 223–31.
19. Sharma S, Saha AK, Lohar G. Optimization of weight and cost of cantilever retaining wall by a hybrid metaheuristic algorithm. *Eng Comput.* 2021;1–27. <https://doi.org/10.1007/s00366-021-01294-x>.
20. Sharma S, Kumar A. MPBOA - a novel hybrid butterfly optimization algorithm with symbiosis organisms search for global optimization and image segmentation. *Multimed Tools Appl.* 2020. <https://doi.org/10.1007/s11042-020-10053-x>.
21. Chakraborty S, Saha AK, Nama S, Debnath S. COVID-19 X-ray image segmentation by modified whale optimization algorithm with population reduction. *Comput Biol Med.* 2021;139: 104984. <https://doi.org/10.1016/j.combiomed.2021.104984>.
22. Mohammadzadeh A, Masdari M, Gharehchopogh FS, Jafarian A. Improved chaotic binary grey wolf optimization algorithm for workflow scheduling in green cloud computing. *Evol Intell.* 2021;14:1997–2025. <https://doi.org/10.1007/s12065-020-00479-5>.

23. Trinh C, Huynh B, Bidaki M, et al. Optimized fuzzy clustering using moth-flame optimization algorithm in wireless sensor networks. Netherlands: Springer; 2021.
24. Sharma S, Kumar A. m-MBOA : a novel butterfly optimization algorithm enhanced with mutualism scheme. *Soft Comput.* 2019. <https://doi.org/10.1007/s00500-019-04234-6>.
25. Chakraborty S, Kumar Saha A, Sharma S, et al. A novel enhanced whale optimization algorithm for global optimization. *Comput Ind Eng.* 2020;153: 107086. <https://doi.org/10.1016/j.cie.2020.107086>.
26. Chakraborty S, Saha AK, Sharma S, et al. A hybrid whale optimization algorithm for global optimization. Heidelberg: Springer, Berlin; 2021.
27. Wolpert DH, Macready WG. No free lunch theorems for optimization. *IEEE Trans Evol Comput.* 1997;1:67–82. <https://doi.org/10.1109/4235.585893>.
28. Zhu L, Ma Y, Bai Y. A self-adaptive multi-population differential evolution algorithm. *Nat Comput.* 2020;19:211–35. <https://doi.org/10.1007/s11047-019-09757-3>.
29. Li X, Wang L, Jiang Q, Li N. Differential evolution algorithm with multi-population cooperation and multi-strategy integration. *Neurocomputing.* 2021;421:285–302. <https://doi.org/10.1016/j.neucom.2020.09.007>.
30. Liang JJ, Qu BY, Suganthan PN. Problem definitions and evaluation criteria for the CEC 2014 special session on single objective real-parameter numerical optimization. Singapore: Nanyang Technological University; 2013. p. 1–21. Tech Rep 201311.
31. Awad NH, Ali MZ, Liang JJ, Qu BY, Suganthan PN. Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective bound constrained real-parameter numerical optimization. Singapore: Nanyang Technological University; 2016. Tech Report.
32. Clerc M, Kennedy J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans Evol Comput.* 2002;6:58–73. <https://doi.org/10.1109/4235.985692>.
33. Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *J Glob Optim.* 2007;39:459–71. <https://doi.org/10.1007/s10898-007-9149-x>.
34. Storn R, Price K. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim.* 1997;11:341–59. <https://doi.org/10.1023/A:1008202821328>.
35. Demšar J. Statistical comparisons of classifiers over multiple data sets. *J Mach Learn Res.* 1993;7:1–30. <https://doi.org/10.5555/1248547.1248548>.
36. Pant M, Thangaraj R, Abraham A. DE-PSO: a new hybrid meta-heuristic for solving global optimization problems. *New Math Nat Comput.* 2011;07:363–81. <https://doi.org/10.1142/s1793005711001986>.
37. Liang JJ, Suganthan PN. Dynamic multi-swarm particle swarm optimizer. In: *Proceedings - 2005 IEEE Swarm Intelligence Symposium. SIS; 2005.* p. 127–32.
38. Storn R. On the usage of differential evolution for function optimization. In: *Biennial Conference of the North American Fuzzy Information Processing Society - NAFIPS.* IEEE; 1996. p. 519–23.
39. Brest J, Greiner S, Bošković B, et al. Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. *IEEE Trans Evol Comput.* 2006;10:646–57. <https://doi.org/10.1109/TEVC.2006.872133>.
40. Wang Y, Cai Z, Zhang Q. Differential evolution with composite trial vector generation strategies and control parameters. *IEEE Trans Evol Comput.* 2011;15:55–66. <https://doi.org/10.1109/TEVC.2010.2087271>.
41. Liang JJ, Qin AK, Suganthan PN, Baskar S. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans Evol Comput.* 2006;10:281–95. <https://doi.org/10.1109/TEVC.2005.857610>.
42. Lin J. Oppositional backtracking search optimization algorithm for parameter identification of hyperchaotic systems. *Nonlinear Dyn.* 2015;80:209–19. <https://doi.org/10.1007/s11071-014-1861-8>.
43. Mirjalili S. SCA: a sine cosine algorithm for solving optimization problems. *Knowledge-Based Syst.* 2016;96:120–33. <https://doi.org/10.1016/j.knosys.2015.12.022>.
44. Gupta S, Deep K. A novel hybrid sine cosine algorithm for global optimization and its application to train multilayer perceptrons. *Appl Intell.* 2020;50:993–1026. <https://doi.org/10.1007/s10489-019-01570-w>.
45. Lakshmi Priya J, Jaya Christa ST. An effective hybridized GWO-BSA for resolving optimal power flow problem with the inclusion of unified power flow controller. *IETE J Res.* 2021. <https://doi.org/10.1080/03772063.2021.1942245>.
46. Nama S, Saha AK, Sharma S. Performance up-gradation of symbiotic organisms search by backtracking search algorithm. *J Ambient Intell Humaniz Comput.* 2021. <https://doi.org/10.1007/s12652-021-03183-z>.
47. Kuyu YÇ, Onieva E, Lopez-Garcia P. A hybrid optimizer based on backtracking search and differential evolution for continuous optimization. *J Exp Theor Artif Intell.* 2021. <https://doi.org/10.1080/0952813X.2021.1872109>.
48. Nama S, Saha AK, Saha A. The hDEBSA global optimization method: a comparative study on CEC2014 test function and application to geotechnical problem. In: Bhoi A, Mallick P, Liu CM, Balas V, editors. *Studies in Computational Intelligence.* Singapore: Springer; 2021. p. 225–58.
49. Das S, Suganthan PN. Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems. Singapore: Nanyang Technological University; 2011. p. 1–42.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.