Available online at www.sciencedirect.com
ScienceDirect
Mathematics and Computers in Simulation 192 (2022) 84–110
MATHEMATICS AND COMPUTERS IN SIMULATION
www.elsevier.com/locate/matcom

Original articles

Honey Badger Algorithm: New metaheuristic algorithm for solving optimization problems

Fatma A. Hashim[a], Essam H. Houssein[b], Kashif Hussain[c,*], Mai S. Mabrouk[d], Walid Al-Atabany[f,e]

[a] *Faculty of Engineering, Helwan University, Egypt*
[b] *Faculty of Computers and Information, Minia University, Egypt*
[c] *Department of Computer Science, Bahria University, Karachi Campus, Pakistan*
[d] *Faculty of Engineering, Misr University for Science and Technology, Egypt*
[e] *Information Technology and Computer Science School, Nile University, Egypt*
[f] *Biomedical Engineering Department, Helwan University, Egypt*

Received 10 January 2021; received in revised form 15 June 2021; accepted 16 August 2021
Available online 2 September 2021

Abstract

Recently, the numerical optimization field has attracted the research community to propose and develop various metaheuristic optimization algorithms. This paper presents a new metaheuristic optimization algorithm called Honey Badger Algorithm (HBA). The proposed algorithm is inspired from the intelligent foraging behavior of honey badger, to mathematically develop an efficient search strategy for solving optimization problems. The dynamic search behavior of honey badger with digging and honey finding approaches are formulated into exploration and exploitation phases in HBA. Moreover, with controlled randomization techniques, HBA maintains ample population diversity even towards the end of the search process. To assess the efficiency of HBA, 24 standard benchmark functions, CEC'17 test-suite, and four engineering design problems are solved. The solutions obtained using the HBA have been compared with ten well-known metaheuristic algorithms including Simulated annealing (SA), Particle Swarm Optimization (PSO), Covariance Matrix Adaptation Evolution Strategy (CMA-ES), Success-History based Adaptive Differential Evolution variants with linear population size reduction (L-SHADE), Moth-flame Optimization (MFO), Elephant Herding Optimization (EHO), Whale Optimization Algorithm (WOA), Grasshopper Optimization Algorithm (GOA), Thermal Exchange Optimization (TEO) and Harris hawks optimization (HHO). The experimental results, along with statistical analysis, reveal the effectiveness of HBA for solving optimization problems with complex search-space, as well as, its superiority in terms of convergence speed and exploration–exploitation balance, as compared to other methods used in this study. The source code of HBA is currently available for public at https://www.mathworks.com/matlabcentral/fileexchange/98204-honey-badger-algorithm.

Keywords: Exploration and exploitation; Honey Badger Algorithm; Meta-heuristic algorithms; Nature-inspired algorithms; Optimization problems; Swarm intelligence algorithms

* Corresponding author.
E-mail address: kashifhussain.bukc@bahria.edu.pk (K. Hussain).

https://doi.org/10.1016/j.matcom.2021.08.013

1. Introduction

Optimization refers to the process of finding best solutions for a given system from all the possible values to maximize or minimize the output. Over the last few decades, as the complexity of problems has increased, the need for new optimization techniques has become imperious [23,25]. Earlier, the conventional mathematical techniques that have been used for solving optimization problems are mostly deterministic that suffer from one major problem: local optima entrapment. This makes these techniques highly inefficient in solving real optimization problems, leading to a growing interest in stochastic optimization techniques over the last two decades [31,45]. Often, most of the real-world optimization problems, in the area of engineering [17], wireless sensor networks [1], image processing [22], feature selection [27,44], tuning of machine learning parameters [18], bio-informatics [13] etc., are highly non-linear and non-convex due to inherent complex constraints and many design variables. Therefore, solving these types of optimization problems is complex because of many inherent local minima. Moreover, there is no guarantee of finding a global solution [38].Thus, the difficulties associated with these types of real-life optimization problems motivate to develop alternative and effective techniques for better solutions.

In order to find better solution, many researchers have tried to propose new algorithms and/or improved the existing methods. Metaheuristic research community has implemented useful search strategies to obtain the global optimum. Because, in real-life optimization problems, search-space grows exponentially and makes the problem landscape highly multimodal, the conventional optimization methods often produce suboptimal solutions. This has, over the past few decades, led to the development of many new metaheuristic algorithms [54]. These methods have shown robust performances on a wider range of complex problems for obtaining the optimal solutions.

Recently, various search strategies have been effectively incorporated in metaheuristic algorithms; mostly inspired from nature, simulating principles of biology, physics, ethology or swarm intelligence [26,36]. Interestingly, some of them such as Genetic Algorithm [5], particle swarm optimization (PSO) [47], and Archimedes optimization algorithm [15] are fairly well-known among not only computer scientists but also scholars from other domains. This has resulted in extensive theoretical work and practical applications using metaheuristic techniques — mainly because of several major reasons including flexibility, gradient-free mechanism, and local optima avoidance. As these methods are gradient-free, there is no need to calculate derivative of the search-space; hence reducing computational cost, being highly flexible for solving a diverse range of problems. Due to these advantages, the application of metaheuristics can be found in different branches of science and industry [28,34].

The metaheuristics algorithms are categorized into two main classes: single solution-based and population-based approaches. The literature has evidenced population-based algorithms having better ability to explore the search space, and exploit the global optimum, as compared to single solution-based algorithms [6]. The population-based algorithms, according to the sources of inspiration, can be divided into three main categories: (1) Swarm Intelligence algorithms (SI), includes swarm-based techniques that mimic the social behavior of insect or animals groups. (2) Evolutionary Algorithms (EAs), which follow natural evolution process found in nature. And, (3) Natural Phenomenon algorithms (NP) that imitate the physical and chemistry principles; while, some include those inspired by human behavior, but are neither SI nor EA.

The SI metaheuristic algorithms mimic the self-organized and collective behaviors in nature. These algorithms take inspirations from the social behavior of animals, birds, plants, and human. Some of the well-known metaheuristic algorithms are: Grasshopper Optimization Algorithm (GOA) [49], Whale Optimization Algorithm (WOA) [43], Elephant Herding Optimization (EHO) [24,51], Harris Hawks optimization (HHO) [19] and Moth-flame Optimization (MFO) [42]. On the other hand, EAs are a type of stochastic global optimization methods inspired by natural evolution and genetic mechanisms [11], such as Genetic Algorithm (GAs) [21], Evolution Strategy (ES) [46], Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) [12], and History-based Adaptive Differential Evolution variants with linear population size reduction (L-SHADE) [50]. NP algorithms imitate the physical or chemical rules in the universe. Some of the popular and recent algorithms are Simulated Annealing (SA) [40], Thermal Exchange Optimization (TEO) [37], and Henry Gas Solubility Optimization [14].

Despite the need for more function evaluations, the literature shows that population-based algorithms are highly suitable for solving real challenging problems [16]. Logically, No-Free-Lunch (NFL) theorem [52] states that there are either no metaheuristic optimization algorithm able to solve all optimization problems or still problems not yet solved. These two reasons are the massive motivation to present a novel metaheuristic algorithm called Honey Badger Algorithm (HBA) which mimics foraging behavior of honey badger. Because the ability to maintain the trade-off balance between exploration and exploitation plays a significant role in effective search, HBA encapsulates

(a) (b)

Fig. 1. (a) Honey badger attacks lion, and (b) Honey badger climbs uppermost branches of trees [39].

dynamic search strategies. This characteristic enables HBA in solving hard optimization problems with many local regions, as it keeps ample population diversity throughout search process for investigating a large area the given landscape. Furthermore, 24 standard mathematical optimization problems, CEC'17 test-suite and, four real-world engineering design optimization problems are solved. The comparison with ten established metaheuristic algorithms, including SA, PSO, CMA-ES, L-SHADE, MFO, EHO, WOA, GOA, TEO, and HHO, validates efficacy of the proposed HBA algorithm.

Eventually, the main contributions of this research are as follows:

1. We propose a new swarm-based optimization algorithm, namely HBA, which mimics the behavior of Honey Badger.
2. The statistical significance, convergence speed, exploitation–exploration ratio, and diversity of HBA are evaluated against the state-of-the-art metaheuristic algorithms.
3. We perform a series of experiments to investigate impact of performance of the proposed algorithm over benchmark optimization problems, CEC'17 test-suite, and real-world engineering design problems, which is regarded as a challenging test-suit in the related literature.
4. HBA outperforms other competitor algorithms on hard optimization problems.

Rest of the paper is organized as follows: Section 2 provides the inspiration and mathematical model of the HBA algorithm. Experiments on standard benchmark and CEC'17 problems are detailed in Section 3, where the relative results are also reported. Section 4 highlights the experimental implementation on engineering design problems, along with presentation of the related results. Section 5 further presents in-depth analysis on the performance of HBA in comparison with several other metaheuristic algorithms. Finally, Section 6 concludes the paper and suggests potential directions for future studies.

2. Honey Badger algorithm

This section discusses the inspiration and mathematical model of Honey Badger Algorithm (HBA) which mimics the behavior of honey badger in nature.

2.1. Honey Badger general biology

Honey badger is a mammal with black and white fluffy fur often found in the semi-deserts and rainforests of Africa, Southwest Asia, and the Indian subcontinent — known for its fearless nature. This dog size (60 to 77 cms body length and 7 to 13 Kgs body weight) fearless forager preys sixty different species including the dangerous snakes. It is an intelligent animal able to use tools, and it loves honey. It prefers to stay solitary in self-dug holes, and meets the other badgers only to mate. There are 12 recognized honey badger subspecies. There is no specific breeding season for honey badgers as cubs are born throughout the year. Because of their courageous nature, it never hesitates attacking even much larger predators when it cannot escape (see Fig. 1a). This animal also can easily climb on trees, as shown in Fig. 1b, for reaching bird nests and beehives for food [3,4,20].

A honey badger locates its prey by walking slowly continuously using smelling mouse skills. It starts to determine the approximate location of prey through digging and ultimately catching it. In a day, it can dig as many as fifty holes in a radius of forty kilometers or more in foraging attempts. Honey badger likes honey, but it is not good in locating beehives. On the other hand, honey-guide (a bird) can locate the hives but cannot get honey. These phenomena lead a relationship between the two, where the bird leads the badger to beehives and helps it open hives using its long claws, then both enjoy the reward of teamwork [39].

2.2. Inspiration

Honey Badger Algorithm (HBA) imitates the foraging behavior of honey badger. For locating food source, the honey badger either smells and digs or follows honeyguide bird. We call the first case as digging mode while the second as honey mode. In the prior mode, it uses its smelling ability to approximate prey location; when reaching there, it moves around the prey to select the appropriate place for digging and catching the prey. In latter mode, honey badger takes the guide of honeyguide bird to directly locate beehive.

2.3. Mathematical model

As discussed earlier, HBA is divided into two phases which are "digging phase" and "honey phase", explained in detail as follows:

2.3.1. Algorithmic steps

This section introduces mathematical formulation of the proposed HBA algorithm. Theoretically, HBA is equipped with both exploration and exploitation phases, hence can be referred to as a global optimization algorithm. Pseudo-code of the proposed algorithm is presented in Algorithm 1; including population initialization, population evaluation, and updating parameters. Mathematically, steps of the proposed HBA are detailed as the following. Here, population of candidate solutions in HBA is represented as:

$$\text{Population of candidate solutions} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1D} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2D} \\ & & \dots\dots\dots\dots & & \\ x_{n1} & x_{n2} & x_{n3} & \dots & x_{nD} \end{bmatrix}$$

$$i\text{th position of honey badger } x_i = \left[x_i^1, x_i^2, \dots, x_i^D\right]$$

Step 1: Initialization phase. Initialize the number of honey badgers (population size N) and their respective positions based on Eq. (1):

$$x_i = lb_i + r_1 \times (ub_i - lb_i), r_1 \text{ is a random number between 0 and 1} \tag{1}$$

where x_i is ith honey badger position referring to a candidate solution in a population of N, while lb_i and ub_i are respectively lower and upper bounds of the search domain.

Step 2: Defining intensity (I). Intensity is related to concentration strength of the prey and distance between it and ith honey badger. I_i is smell intensity of the prey; if the smell is high, the motion will be fast and vice versa, is given by Inverse Square Law [35] as shown in Fig. 2 and is defined by Eq. (2).

$$\begin{aligned} I_i &= r_2 \times \frac{S}{4\pi d_i^2}, r_2 \text{ is a random number between 0 and 1} \\ S &= (x_i - x_{i+1})^2 \\ d_i &= x_{prey} - x_i \end{aligned} \tag{2}$$

where S is source strength or concentration strength (location of prey as shown in Fig. 2). In Eq. (2), d_i denotes distance between prey and the ith badger.

Step 3: Update density factor. The density factor (α) controls time-varying randomization to ensure smooth transition from exploration to exploitation. Update decreasing factor α that decreases with iterations to decrease randomization with time, using Eq. (3):

$$\alpha = C \times \exp\left(\frac{-t}{t_{max}}\right), t_{max} = \text{maximum number of iterations} \tag{3}$$

where C is a constant ≥ 1 (default = 2).

Step 4: Escaping from local optimum. This step and the two next steps are used to escape from local optima regions. In this context, the proposed algorithm uses a flag F which alters search direction for availing high opportunities for agents to scan the search-space rigorously.

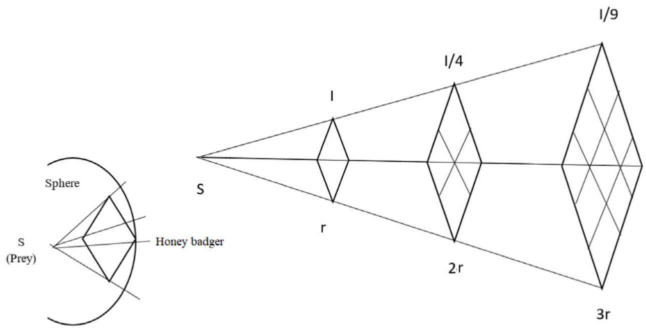


Fig. 2. Inverse square law. I is smell intensity, S is location of prey, and r is random number between 0 and 1.

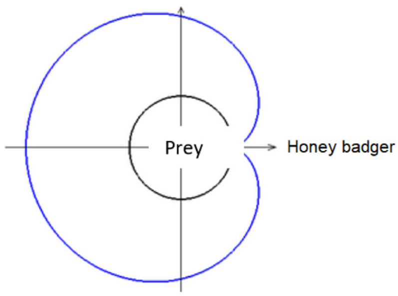


Fig. 3. Digging phase: blue outline is smell intensity, black circular line shows prey location. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Step 5: Updating the agents' positions. As discussed earlier, HBA position update process (x_{new}) is divided into two parts which are "digging phase" and "honey phase". Following is given better explanation:

Step 5-1: Digging phase. In digging phase, a honey badger performs action similar to Cardioid shape [2] as shown in Fig. 3. The Cardioid motion can be simulated by Eq. (4):

$$x_{new} = x_{prey} + F \times \beta \times I \times x_{prey} + F \times r_3 \times \alpha \times d_i \times |\cos(2\pi r_4) \times [1 - \cos(2\pi r_5)]| \quad (4)$$

where x_{prey} is position of the prey which is the best position found so far – global best position in other words. $\beta \geq 1$ (default $= 6$) is ability of the honey badger to get food. d_i is distance between prey and the ith honey badger, see Eq. (2). r_3, r_4, and r_5 are three different random numbers between 0 and 1. F works as the flag that alters search direction, it is determined using Eq. (5):

$$F = \begin{cases} 1 & \text{if } r_6 \leq 0.5 \\ -1 & \text{else,} \end{cases} \quad r_6 \text{ is a random number between 0 and 1} \quad (5)$$

In the digging phase, a honey badger heavily relies on smell intensity I of prey x_{prey}, distance between the badger and prey d_i, and time-varying search influence factor α. Moreover, during digging activity, a badger may receive any disturbance F which allows it to find even better prey location (see Fig. 3).

Step 5-2: Honey phase. The case when a honey badger follows honey guide bird to reach beehive can be simulated as Eq. (6):

$$x_{new} = x_{prey} + F \times r_7 \times \alpha \times d_i, \, r_7 \text{ is a random number between 0 and 1} \quad (6)$$

where x_{new} refer to the new position of honey badger, whereas x_{prey} is prey location, F and α are determined using Eqs. (5) and (3), respectively. From Eq. (6), it can be observed that a honey badger performs search close to prey location x_{prey} found so far, based on distance information d_i. At this stage, the search is influenced by search behavior varying by time (α). Moreover, a honey badger may find disturbance F.

Algorithm 1 Pseudo code of HBA.

Set parameters t_{max}, N, β, C.
Initialize population with random positions.
Evaluate the fitness of each honey badger position x_i using objective function and assign to f_i, $i \in [1, 2, ..., N]$.
Save best position x_{prey} and assign fitness to f_{prey}.
while $t \leq t_{max}$ **do**
  Update the decreasing factor α using (3).
  for $i = 1$ to N **do**
    Calculate the intensity I_i using Eq. (2).
    if $r < 0.5$ **then** ▷ r is random number between 0 and 1
      Update the position x_{new} using Eq. (4).
    else
      Update the position x_{new} using Eq. (6).
    end if
    Evaluate new position and assign to f_{new}.
    if $f_{new} \leq f_i$ **then**
      Set $x_i = x_{new}$ and $f_i = f_{new}$.
    end if
    if $f_{new} \leq f_{prey}$ **then**
      Set $x_{prey} = x_{new}$ and $f_{prey} = f_{new}$.
    end if
  end for
end while Stop criteria satisfied.
Return x_{prey}

Table 1
Sensitivity analysis for parameters β and C.

Parameters Scenarios	Scenario 1 $\beta = 0.5$	Scenario 2 $\beta = 2.0$	Scenario 3 $\beta = 4.0$	Scenario 4 $\beta = 6.0$	Scenario 5 $\beta = 8.0$
$C = 0.5$	1.556E+04	1.288E+04	1.256E+04	1.101E+04	**0.178E+04**
$C = 1.0$	1.071E+04	1.0825e+04	1.059E+04	8.667E+03	9.116E+03
$C = 1.5$	9.933E+03	9.342E+03	9.427E+03	1.025E+04	8.568E+03
$C = 2.0$	9.176E+03	8.838E+03	9.579E+03	8.071E+03	1.036E+04
$C = 2.5$	9.5479e+03	8.535E+03	8.689E+03	9.106E+03	1.118E+04

Theoretically, HBA is regarded as a global optimization algorithm due to exploration and exploitation phases. In order to make HBA easy to implement and understand, the number of operators to be adjusted are minimized. Note that the computational complexity of the proposed method is of $O(t_{max}ND)$ where t_{max} shows the maximum number of iterations, N is the number of solutions or population size, and D indicates the number of decision variable. Therefore, the overall complexity including the objective function defined in Eqs. (4) and (6) is calculated as $O(t_{max}ND)$.

2.4. Sensitivity analysis

Since BHA uses two user-defined parameters (β and C), it is important to select values for these parameters carefully because they affect HBA performance significantly. To analyze the effect of these parameters, sensitivity analysis is performed on CEC Composition Function 2 (N = 3). In this connection, five scenarios are drawn with the combination of different values for β and C. As it can be observed in Table 1 and Fig. 4 that for $\beta = 6$ the best fitness value was achieved at $c = 2$. This concludes that the best parameter values for the proposed algorithm are $\beta = 6$ and $c = 2$ (see Fig. 4).

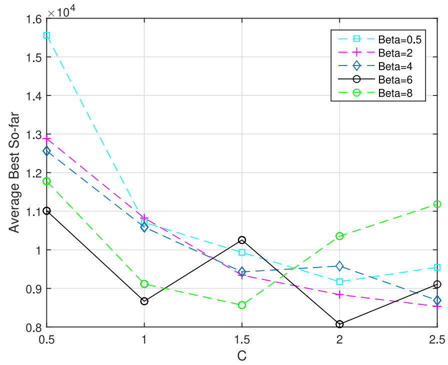


Fig. 4. Fitness achieved by BHA for parameters β and c.

2.5. Exploration and exploitation phases

Exploration and exploitation are the two major components of any search strategy [9]. Exploration is ensured by extending search to far-reached regions in the search space. On the other hand, by exploitation, the search agents converge towards already identified promising region, using local search strategy [7]. The important factor in maintaining balance between the two contradictory capabilities is to ensure right amount of randomness [29]. In this connection, the HBA algorithm has three main control parameters. (1) I: is the intensity and it relates two main parts. The first one is the distance between the honey badger and prey, this value is decreased or increased with time which results the intensity adjustment for enabling search agents to transfer from exploration to exploitation and vice versa according to the state of the given individual. The second part is the distance between each two neighboring search agents which carries interaction between search agents, hence escaping from local optimum. (2) α: is a randomization control factor which deceases with time to reduce population diversity over the course of iterations. This achieves the required trade-off balance between exploration and exploitation. And, (3) F: is the flag that changes search direction of agents and provides population diversity for rigorous exploration of the given search space.

Usually, exploration and exploitation abilities are theoretically analyzed in metaheuristic research, without providing substantial practical measures. In this paper, the exploration and exploitation is obtained using dimension-wise diversity measurement presented by Hussain et al. in [29]. According to this approach, during the course of search process, exploration can be measured by the increased mean value of distance within dimensions of population; on the other hand, reduced mean value can be considered as exploitation phase where search agents are located in a concentrated region. That said, if the reduced mean value of dimension-wise diversity remains unchanged constantly, then it can be suggested that the algorithm has achieved convergence. Here, it is important to notice that if the convergence is achieved early in iterations without finding global optimum, then an algorithm is supposed to suffer from premature convergence problem. This implies that an efficient search strategy does not allow search agents converge without enough exploration of the search-space. In search process, the dimension-wise diversity can be measured as Eq. (7):

$$
\begin{aligned}
Div_j &= \frac{1}{N}\sum_{i=1}^{N} median(x^j) - x_i^j, \\
Div^t &= \frac{1}{D}\sum_{j=1}^{D} Div_j,\ t = 1, 2, \ldots, t_{max}
\end{aligned}
\tag{7}
$$

where x_i^j is jth dimension of ith honey badger position and median(x^j) is the median value of jth dimension in population of N candidate solutions. The Div_j is mean diversity for dimension j. This dimension-wise diversity is then averaged in Div^t on all D dimensions for t iteration. Once the population diversity is computed for all iterations t_{max}, it is now possible to determine the percentage of exploration and exploitation using Eq. (8):

$$Exploration\% = \frac{Div^t}{\max(Div)} \times 100; \quad Exploitation\% = \frac{\left|Div^t - \max(Div)\right|}{\max(Div)} \times 100 \tag{8}$$

where $\max(Div)$ is the maximum diversity in t_{max} iterations.

3. Experimental results

The proposed HBA algorithm was implemented in MATLAB version R2016a. The numerical efficiency of HBA was evaluated by solving 24 standard benchmark functions, 29 functions of CEC'17, and 4 engineering design problems. For performance validation of HBA, the results are compared with ten state-of-the-art optimization algorithms: SA [40], PSO [47], CMA-ES [12], L-SHADE [50], MFO [42], EHO [51], WOA [43], GOA [49], TEO [37], and HHO [19]. From the selected competitive methods, SA, PSO, CMA-ES, and L-SHADE are well established algorithms in metaheuristic literature. On the other hand, MFO, WOA, GOA, TEO, and HHO are relatively new additions but have shown significant results while solving optimization problems employed in this research. The reason to make such selection is to include acknowledged, as well as, promising algorithms in our comparisons, in order to prove overall efficacy of the proposed method. To obtain a fair comparison, HBA and the competitor algorithms were executed for 30 independent runs, and the maximum number of iterations was 1000 for each optimization problem.

3.1. Parameter settings

Apart from algorithm specific parameter settings mentioned in Table 2, some common settings among the selected algorithms include, 50 as population size (N), 1000 maximum iterations (t_{max}), and 30 independent runs for each optimization problem.

3.2. Standard benchmark functions analysis

3.2.1. Benchmark functions description

The performance of HBA was investigated on 24 standard benchmark functions [32] and compared with ten metaheuristic algorithms. This test-bed is divided to three categories: unimodal functions ($f_1 - f_8$) have only one global optimum, hence used for assessing the exploitation ability of optimization methods. In contrast, multimodal functions ($f_9 - f_{16}$), with several local optimal regions, test optimization methods with ability to void local optima and find global optimum location. Fixed-dimension multimodal functions ($f_{17} - f_{24}$) have a large number of local optima, so they are used to evaluate exploration and exploitation balance in metaheuristic algorithms. Detailed descriptions are given in Tables 3–5.Notably, D, R and $f(x^*)$ indicate dimension of the function, boundary of search-space, and objective function value of the optimum location, respectively. Overall, this test-bed with 24 optimization problems of variety of difficulties poses significant challenge for any optimization problem, hence we preferred these problems to effectively validate the performance HBA as well as the competitive methods.

3.2.2. Statistical results

Tables 6, 7, and 8 report mean and standard deviation (STD) of the optimal fitness values achieved by the selected algorithms over 30 independent runs. Moreover, to determine statistical significance of the proposed HBA against the counterparts, Table 9 presents p-value obtained by Friedman Test with $\alpha = 0.05$. According to mean and standard deviation values presented in the mentioned tables, it can be suggested that HBA achieved absolute global optimum on 11 out of 16 30-D unimodal and multimodal functions and half of the 50-D functions; considering $f(x^*) \leq$1.0E−300 and -8.800E−16 as zero. Also, on fixed-dimensional functions, HBA successfully found global optimum on half of the problems. Other than HBA, HHO was the second best algorithm in our experiments, by achieving global optimum on 6 out of 16 unimodal and multimodal functions each for 30-D and 50-D. In comparison with the selected 8 counterparts, HBA outperformed all by achieving best mean values on overall benchmark test suit. Specifically, HBA produced smaller mean and standard deviation than SA, PSO, L-SHADE, MFO, EHO,

Table 2
Parameters settings of HBA and selected algorithms.

Algorithms	Parameters
SA	Materials number = 50 Cooling rate $\alpha = 0.8$ Initial temperature $T_0 = 1$
PSO	Swarm size $S = 50$ Inertia weight decreases linearly from 0.9 to 0.4 (Default) C_1 (individual-best acceleration factor) increases linearly from 0.5 to 2.5 (Default) C_2 (global-best acceleration factor) decreases linearly from 2.5 to 0.5 (Default)
CMA-ES	Population size N = 50 Number of parents $\mu = \lfloor N/2 \rfloor$ Parent weights $w = \log(mu + 0.5) - \log(1 : mu)$ Step size $\sigma = 0.3 \times 200$
L-SHADE	Population size = 50 Crossover rate $M_{CR} = 0.5$ Scaling factor $M_F = 0.5$
MFO	Moth-flame number = 50 a (The convergence constant) $[-2 : -1]$ b (Spiral factor) = 1
EHO	Elephants number = 50 Clans number = 5 Kept elephants number = 2 The scale factor $\alpha = 0.5$ The scale factor $\beta = 0.1$
WOA	Whales number = 50 a variable decreases linearly from 2 to 0 (Default) $a2$ linearly decreases from -1 to -2 (Default)
GOA	Grasshoppers number = 50 Intensity of attraction = 0.5 Attractive length scale = 1.5 C_{min} = 0.00004 C_{max} = 1
TEO	Objects number = 50 TMs (Thermal memory) = 3 c_1, c_2 (controlling variables) = 1
HHO	Harris Hawk number = 50 E_0 variable changes from -1 to 1 (Default)
HBA	Honey Badger number = 50 β (the ability of a honey badger to get food) = 6 C = 2

and GOA on all unimodal and multimodal 30-D and 50-D functions, except for CMA-ES which achieved global optimum on 30 dimensional f_{13}. The proposed algorithm produced superior results among the competitor methods on all fixed-dimensional suite. Compared with WOA, HBA performed better on 14 out of 16 30-D functions, while both found global optimum on two problems (f_{13} and f_{14}). Similarly, HBA outperformed WOA on 15 50-D functions and both generated global optimum on f_7, f_{13}, and f_{14}. HBA also generated smaller mean and standard deviation values as compared to TEO and HHO on 12 out of 16 30-D and 50-D unimodal and multimodal functions, whereas the three methods found global optimum (zero) on f_7, f_{12}, f_{13}, and f_{14}.

The earlier discussion is further validated by Table 9 presenting statistical significance of the results generated by HBA and rest of the methods selected in this study. The p-value suggests that the performance of HBA on 30-D unimodal and multimodal suite is significantly different from SA, PSO, CMA-ES, L-SHADE, MFO, WOA, and GOA with 95% certainty, while it is insignificantly different from TEO and HHO. Same applies to 50-D test-suite. While on fixed-dimensional benchmark functions, HBA generated significantly different results than the counterparts

Table 3
Unimodal functions description.

Expression	Name	D	R	$f(x^*)$
$f_1(x) = \left(\sum_{i=1}^{D} x_i^2\right)^2$	Chung Reynolds	30	[−100,100]	0
$f_2(x) = \sum_{i=1}^{D} x_i^2$	De Jong's (sphere)	30	[−100, 100]	0
$f_3(x) = \sum_{j=2}^{D-2} [(x_{i-1} - 10x_i)^2$ $+5(x_{i+1} - x_{i+2})^2$ $+(x_i - x_{i+1})^4$ $+10(x_{i-1} - x_{i+2})^4]$	Powell Singular-2	30	[−4,5]	0
$f_4(x) = \sum_{i=1}^{D} \lvert x_i\rvert^{i+1}$	Powell Sum	30	[−1,1]	0
$f_5(x) = \sum_{i=1}^{D} \lvert x_i\rvert$	Schwefel 2.20	30	[−100,100]	0
$f_6(x) = \sum_{i=1}^{D} \lvert x_i\rvert + \prod_{i=1}^{D} \lvert x_i\rvert$	Schwefel 2.22	30	[−100,100]	0
$f_7(x) = \sum_{i=1}^{D} x_i^{10}$	Schwefel 2.23	30	[−10,10]	0
$f_8(x) = \sum_{i=1}^{D} i x_i^2$	Sum Squares	30	[−10,10]	0

except for PSO. When performing pair-wise comparison of complete benchmark suite performances, the p-values suggest that the overall performance of HBA is significantly different from rest of the methods used in experiments.

3.2.3. Convergence analysis

The statistical results discussed earlier show robustness of the proposed HBA algorithm. How efficiently the method progressed during the course of search process is illustrated via convergence graphs provided in Figs. 5–7 for unimodal and multimodal functions with dimensions 30, 50 and fixed-dimensions, respectively. The graphs show best values found by the selected algorithms in each of 1000 iterations, the values are averaged over 30 independent runs. Note that convergence graphs for all standard benchmark functions are not presented; instead, some selected graphs are reported in this paper in order to limit paper length. Figs. 5 and 6 present convergence graphs for functions f_1, f_4, f_7, f_{10}, f_{13}, and f_{16}. While, 7 shows convergence of the selected algorithms on f_{17}, f_{19}, f_{20}, f_{22}, f_{23}, and f_{24}. According to the convergence curves for 30-D and 50-D unimodal and multimodal functions, HBA clearly converged faster than the counterparts like SA, PSO, CMA-ES, L-SHADE, MFO, EHO, WOA, GOA, TEO, and HHO. Same can be inferred for fixed-dimensional functions when observing Fig. 7. From the graphs, it is also apparent that SA, PSO, CMA-ES, L-SHADE, GOA, and EHO suffered from premature convergence by trapping in local optimal locations, hence did not show any improvement in most part of search process — apart from fixed-dimensional functions where mostly PSO displayed relatively better convergence ability. Overall, the convergence analysis performed in this section, it can be inferred that the proposed HBA algorithm remained the best method among others with ability to find global optimum location faster than other selected methods. Though, HHO and TEO also maintained better convergence ability on most of the benchmark functions in this experimental study.

3.2.4. Exploration–exploitation analysis

Exploration and exploitation are two major corner-stones of any metaheuristic performance. An efficient search strategy is devised with heuristic approach that helps avoid trapping in local regions. Hence, it is important to

Table 4
Multimodal optimization problems.

Mathematical expression	Name	D	R	$f(x^*)$
$f_9(x) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right) + 20 + e$	Ackley	30	[−32,32]	0
$f_{10}(x) = \sum_{i=1}^{D-1} \left[(x_i^2)^{x_{i+1}^2+1} + (x_{i+1}^2)^{x_i^2+1}\right]$	Brown	30	[−1,4]	0
$f_{11}(x) = x_1^2 + 10^6 \sum_{i=2}^{D} x_i^2$	Cigar	30	[−10,10]	0
$f_{12}(x) = \sum_{i=1}^{D} x_i^6 \left[2 + \sin\left(\frac{1}{x_i}\right)\right]$	Csendes	30	[−1,1]	0
$f_{13}(x) = 1 + \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right)$	Griewank	30	[−600,600]	0
$f_{14}(x) = 10D + \sum_{i=1}^{D} \left[x_i^2 - 10\cos(2\pi x_i)\right]$	Rastrigin	30	[−5.12, 5.12]	0
$f_{15}(x) = \sum_{i=2}^{D} \left[(x_i - 1)^2 + (x_1 - x_i^2)^2\right]$	Schwefel 2.25	30	[0,10]	0
$f_{16}(x) = \sum_{i=1}^{D} x_i^2 + \left(\sum_{i=1}^{D} 0.5 i x_i\right)^2 + \left(\sum_{i=1}^{D} 0.5 i x_i\right)^4$	Zakharov	30	[−5,10]	0

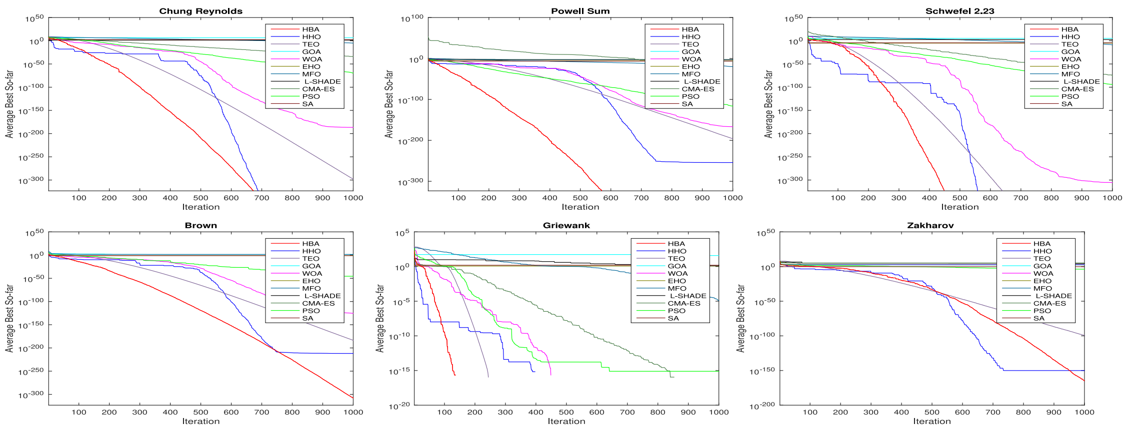


Fig. 5. Convergence curves of competitor algorithms on unimodal and multimodal functions with 30 dimensions.

measure exploration and exploitation in a metaheuristic so that efficiency of the in-built search strategy can be evaluated. According to [29,30], merely investigating convergence graphs and end results will not present insightful information about how exactly the algorithm behaved during the course of search process. It is therefore, this study measured exploration and exploitation percentage ratios of the proposed HBA while solving optimization

Table 5
Fixed-dimensional optimization problems.

Mathematical expression	Name	D	R	$f(x^*)$
$f_{17}(x) = -\frac{0.001}{\lfloor 0.001^2 + (x_1 - 0.4x_2 - 0.1)^2 \rfloor} - \frac{0.001}{\lfloor 0.001^2 + (2x_1 + x_2 - 1.5)^2 \rfloor}$	Chen Bird	2	[−500,500]	−2000
$f_{18}(x) = 100(x_1 - x_2^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1(x_2 - 1)^2 + (x_4 - 1)^2 + 19.8(x_2 - 1)(x_4 - 1)$	Colville	4	[−10,10]	0
$f_{19}(x) = \left[1 - \left\lvert \frac{\sin[\pi(x_1 - 2)] \sin[\pi(x_2 - 2)]}{\pi^2 (x_1 - 2)(x_2 - 2)} \right\rvert^5 \right] \times \left[2 + (x_1 - 7)^2 + 2(x_2 - 7)^2\right]$	Damavandi	2	[0,14]	0
$f_{20}(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$	Matyas	2	[−10,10]	0
$f_{21}(x) = (2x_1^3x_2 - x_2^3)^2 + (6x_1 - x_2^2 + x_2)^2$	Price	2	[−500,500]	0
$f_{22}(x) = g(r).h(t)$ where $g(r) = \left[\sin(r) - \frac{\sin(2r)}{2} + \frac{\sin(3r)}{3} + \frac{\sin(4r)}{4} + 4\right] \left(\frac{r^2}{r+1}\right)$ $h(t) = 0.5\cos(2t - 0.5) + \cos(t) + 2$ $r = \sqrt{x_1^2 + x_2^2}$ $t = atan2(x_1 + x_2)$	Sawtoothxy	2	[−20,20]	0
$f_{23}(x) = 4 - 4x_1^3 + 4x_1 + x_2^2$	Trecanni	2	[−5,5]	0
$f_{24}(x) = 0.5x_1^2 + 0.5[1 - \cos(2x_1)] + x_2^2$	Zirilli	2	[−500,500]	0

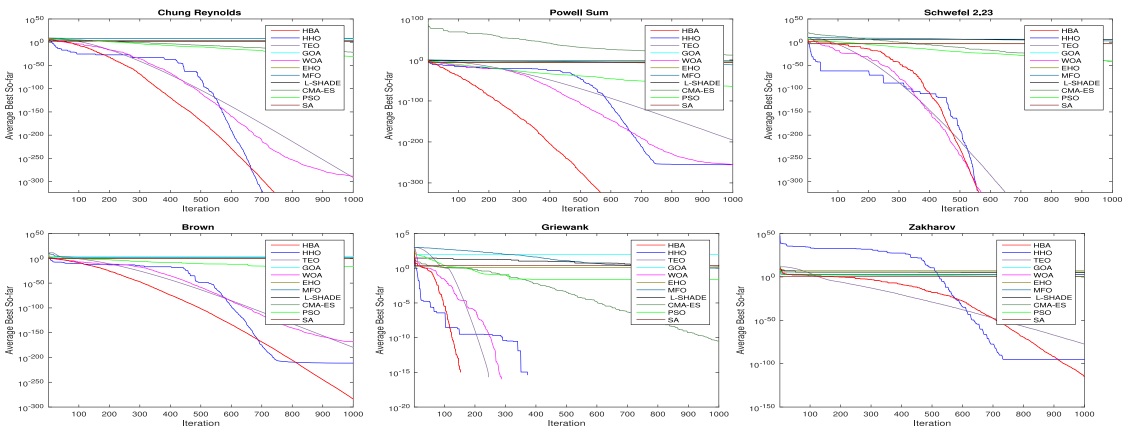


Fig. 6. Convergence curves of competitor algorithms on unimodal and multimodal functions with 50 dimensions.

problems with 50 dimensions. The ratios are depicted via Fig. 8 for functions f_1, f_4, f_7, f_{10}, f_{13}, and f_{16}, as an example of general behavior of HBA. Fig. 8 which shows that mostly HBA started with high exploration and low exploitation; and as the iterations progress, HBA maintains exploitation high. However, it can also be stated that HBA showed dynamic behavior and maintained exploration relatively high on some functions, like Powell Sum (f_4) and Griewank (f_{13}), to avoid trapping in local optima. As far as the fixed-dimensional functions are concerned, Fig. 9 shows exploration–exploitation behavior of HBA for functions f_{17}, f_{19}, f_{20}, f_{22}, f_{23}, and f_{24}. On these

Table 6
Statistical results obtained for the unimodal and multimodal functions with Dim = 30.

Fun.	Meas.	SA	PSO	CMA-ES	L-SHADE	MFO	EHO	WOA	GOA	TEO	HHO	HBA
f_1	Mean	1.21E+02	1.58E−75	7.74E−35	9.57E+02	5.34E−06	9.86E−12	1.12E−169	2.48E+06	4.98E−297	**0.00E+00**	**0.00E+00**
	STD	2.12E+01	8.19E−75	9.58E−35	1.35E+03	3.70E−06	2.37E−12	0.00E+00	1.65E+06	0.00E+00	0.00E+00	0.00E+00
f_2	Mean	9.73E−02	3.71E−28	2.94E−18	1.21E+01	2.0515E−08	2.52E−06	2.66E−122	1.43E+01	6.46E−184	7.64E−202	**6.74E−302**
	STD	7.69E−03	2.03E−27	2.08E−18	1.60E+01	9.01E−06	4.77E−07	1.24E−121	9.67E+00	0.00E+00	0.00E+00	0.00E+00
f_3	Mean	6.34E+00	1.0734e−09	2.65E−08	2.33E+07	7.51E−05	1.20E−03	4.25E−114	2.57E+03	9.41E−183	8.28E−196	**8.42E−289**
	STD	8.58E−01	5.01E−10	9.90E−09	7.17E+06	1.13E−04	7.67E−05	2.27E−113	2.46E+02	0.00E+00	0.00E+00	0.00E+00
f_4	Mean	2.68E−08	5.39E−91	2.07E−07	1.87E−01	5.35E−22	2.28E−09	7.96E−153	7.32E−04	3.39E−195	2.67E−251	**0.00E+00**
	STD	4.74E−09	1.74E−90	2.04E−07	4.87E−01	7.13E−22	4.15E−09	3.95E−152	6.54E−04	0.00E+00	0.00E+00	0.00E+00
f_5	Mean	2.48E+01	3.50E−03	2.06E−08	2.69E+01	1.00E+02	1.49E−02	5.39E−84	3.18E+02	5.12E−105	3.78E−101	**8.90E−157**
	STD	8.54E−01	8.6E−03	3.83E−09	8.51E+01	1.73E+02	8.59E−04	2.91E−83	7.17E+01	2.04E−105	3.74E−101	4.21E−156
f_6	Mean	2.21E+39	7.5E−03	8.53E−08	6.91E+10	3.33E+02	1.49E−02	1.35E−83	1.68E+31	8.20E−104	2.52E−105	**1.50E−156**
	STD	2.96E+39	1.59E−02	1.32E−08	3.83E+10	2.00E+02	1.00E−03	5.63E−83	2.91E+31	2.32E−104	4.33E−105	3.70E−156
f_7	Mean	1.74E−06	2.87E−89	4.48E−73	7.92E+01	5.86E−10	4.79E−34	5.22E−287	2.05E+05	**0.00E+00**	**0.00E+00**	**0.00E+00**
	STD	1.28E−06	1.57E−88	6.34E−73	3.93E+01	9.34E−10	5.10E−34	0.00E+00	1.55E+05	0.00E+00	0.00E+00	0.00E+00
f_8	Mean	4.96E+00	1.15E−42	3.36E−17	1.99E+04	5.51E+02	1.48E−05	2.24E−120	4.59E+02	2.95E−182	1.29E−196	**2.00E−300**
	STD	5.46E−01	1.94E−42	7.39E−18	6.39E+03	5.67E+02	2.22E−06	6.14E−120	2.26E+02	0.00E+00	0.00E+00	0.00E+00
f_9	Mean	6.84E+00	8.54E−01	1.25E−01	1.74E+00	2.41E+00	1.54E+00	2.96E−16	4.04E+00	**−8.88E−16**	**−8.88E−16**	**−8.88E−16**
	STD	5.19E−01	9.32E−02	3.86E−14	3.87E+00	6.37E−01	1.49E−01	2.05E−15	9.29E−01	0.00E+00	0.00E+00	0.00E+00
f_{10}	Mean	4.46E−02	3.61E−41	1.65E−65	4.99E−05	8.00E+00	6.33E−01	1.68E−120	2.09E+02	5.52E−184	3.64E−203	**1.96E−300**
	STD	3.47E−03	6.25E−41	7.54E−65	9.54E−5	2.00E+00	3.25E−01	2.91E−120	1.47E+02	0.00E+00	0.00E+00	0.00E+00
f_{11}	Mean	3.65E+07	1.81E−31	2.38E−12	4.10E+06	2.93E+01	2.65E+07	8.61E−113	1.05E+10	1.57E−175	2.05E−191	**5.78E−295**
	STD	3.82E+06	9.91E−31	1.70E−13	4.35E+06	1.73E+01	5.26E+06	4.55E−112	8.96E+09	0.00E+00	0.00E+00	0.00E+00
f_{12}	Mean	3.24E−09	1.09E−11	1.50E−48	5.80E+02	2.12E−09	2.24E−09	9.91E−183	6.73E−04	**0.00E+00**	**0.00E+00**	**0.00E+00**
	STD	7.29E−10	1.33E−11	5.75E−50	8.18E+02	3.06E−09	6.29E−10	0.00E+00	1.00E−03	0.00E+00	0.00E+00	0.00E+00
f_{13}	Mean	1.41E+00	1.72E−02	**0.00E+00**	1.38E+00	2.50E−03	1.19E+00	**0.00E+00**	9.08E+01	**0.00E+00**	**0.00E+00**	**0.00E+00**
	STD	2.81E−02	7.36E−03	0.00E+00	1.41E−01	4.25E−03	1.43E−01	0.00E+00	6.38E+01	0.00E+00	0.00E+00	0.00E+00
f_{14}	Mean	2.94E+02	4.68E+01	9.70E+01	1.54E+03	1.83E+02	2.72E+01	**0.00E+00**	2.39E+02	**0.00E+00**	**0.00E+00**	**0.00E+00**
	STD	1.36E+01	1.21E+01	1.20E+02	2.01E−01	3.00E+01	2.58E+01	0.00E+00	3.23E+01	0.00E+00	0.00E+00	0.00E+00
f_{15}	Mean	1.47E+03	3.05E−20	2.43E−08	2.91E+08	4.53E−01	2.51E+02	2.65E−02	2.95E+09	7.82E−189	1.73E−13	**0.00E+00**
	STD	1.96E+02	5.29E−20	1.53E−09	3.65E+08	5.97E−01	3.35E+02	2.55E−02	3.10E+09	0.00E+00	1.39E−13	0.00E+00
f_{16}	Mean	3.38E−01	5.53E−04	1.25E+05	4.22E+04	3.18E+02	3.06E−04	452.66E+00	1.17E+03	7.67E−97	2.58E−136	**6.21E−157**
	STD	5.24E−02	8.88E−04	2.56E+04	3.76E+04	8.19E+01	1.39E−04	102.47E+00	4.79E+02	1.33E−96	4.47E−136	3.23E−156

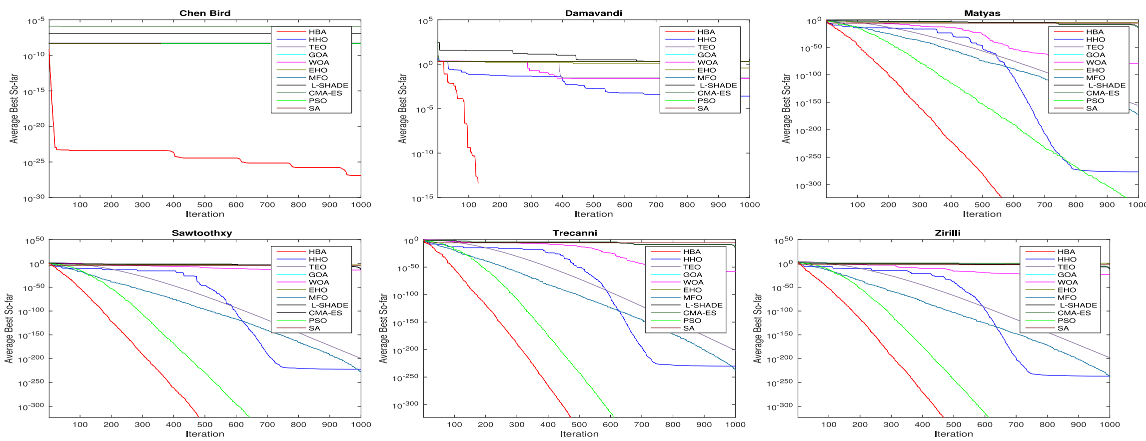


Fig. 7. Comparison of convergence characteristic of competitor algorithms obtained in fixed-multimodal dimension.

complex and multimodal functions, HBA showed highly dynamic behavior. It can be seen in Fig. 9 that HBA started search with high exploration and low exploitation, and later, it remained more exploitative. That said, it can also be seen that HBA varied exploration–exploitation ratios during search process for searching best optimal location. The exploration–exploitation percentage measurements for all the selected algorithms are shown in Fig. 10 for unimodal and multimodal 50-D functions. The stacked bar charts show the exploration and exploitation percentage measured in 1000 iterations. According to the graphs, HBA maintained exploration more than 50% on 10 out of 16 unimodal and multimodal functions; however, the exploitation percentage ratio was higher around 60% on rest of the functions. Same is the case with TEO which performed third best in our experiments. Contrarily, HHO,

Table 7
Statistical results obtained for the unimodal and multimodal functions with Dim = 50.

Fun.	Meas.	SA	PSO	CMA-ES	L-SHADE	MFO	EHO	WOA	GOA	TEO	HHO	HBA
f_1	Mean	1.75E+03	2.16E−30	3.96E−22	1.83E−01	6.68E+02	1.69E+02	2.51E−286	1.13E+07	7.69E−291	**0.00E+00**	**0.00E+00**
	STD	6.86E+02	2.30E−30	1.90E−22	2.57E−01	4.54E+02	7.15E+01	0.00E+00	1.44E+07	0.00E+00	0.00E+00	0.00E+00
f_2	Mean	4.75E−01	5.57E−14	1.83E−11	1.14E−01	2.62E+01	2.06E−01	5.15E−179	1.65E+01	1.02E−180	5.48E−207	**6.73E−286**
	STD	3.87E−02	7.68E−14	1.74E−12	3.80E−03	1.20E−03	1.21E−01	0.00E+00	6.82E+00	0.00E+00	0.00E+00	0.00E+00
f_3	Mean	3.95E+01	2.14E−05	8.00E+03	5.63E+06	7.69E+03	4.88E+01	3.94E−177	4.48E+03	7.22E−179	5.82E−189	**2.41E−280**
	STD	2.47E+00	3.40E−05	7.09E+03	7.91E+05	6.72E+03	4.69E+00	0.00E+00	2.59E+03	0.00E+00	0.00E+00	0.00E+00
f_4	Mean	2.30E−07	7.31E−61	1.40E+14	6.76E−03	5.05E−12	1.83E−05	6.37E−270	7.67E−04	3.86E−196	2.31E−260	**0.00E+00**
	STD	1.61E−07	1.23E−60	1.98E+14	5.02E−03	7.15E−12	1.82E−05	0.00E+00	3.79E−04	0.00E+00	0.00E+00	0.00E+00
f_5	Mean	1.12E+02	2.70E+00	4.53E−05	8.29E+01	1.01E+02	3.99E+01	9.02E−110	5.57E+02	2.70E−103	3.06E−101	**4.96E−149**
	STD	9.52E+00	1.51E+00	8.64E−06	3.98E+01	1.00E+02	4.04E+00	1.46E−109	1.65E+02	3.23E−104	5.21E−101	1.58E−148
f_6	Mean	2.03E+56	3.15E+02	1.52E−03	3.98E+15	8.00E+02	3.30E+01	6.60E−107	2.87E+45	4.20E−98	2.06E−101	**8.82E−148**
	STD	1.79E+56	5.07E+02	1.48E−03	5.87E+15	4.00E+02	6.91E+00	1.13E−106	4.98E+45	7.26E−98	3.56E−101	3.69E−147
f_7	Mean	1.60E−03	2.35E−37	4.84E−42	5.12E+07	2.71E+01	1.80E−05	**0.00E+00**	1.56E+06	**0.00E+00**	**0.00E+00**	**0.00E+00**
	STD	7.41E−04	4.05E−37	6.07E−42	3.02E+07	2.25E+01	2.19E−05	0.00E+00	1.89E+06	0.00E+00	0.00E+00	0.00E+00
f_8	Mean	5.28E+01	7.65E−14	4.82E−09	1.47E+05	2.60E+03	2.68E+01	5.35E−178	1.92E+03	7.79E−179	2.79E−200	**7.26E−284**
	STD	7.80E+00	9.82E−14	3.15E−09	6.55E+03	1.61E+03	2.13E+01	0.00E+00	8.86E+02	0.00E+00	0.00E+00	0.00E+00
f_9	Mean	6.54E+00	1.04E+00	1.98E−01	2.78E−01	3.69E+00	1.84E+00	2.66E−15	3.67E+00	**−8.88E−16**	**−8.88E−16**	**−8.88E−16**
	STD	5.19E−01	1.21E−01	1.68E−02	6.30E−01	4.29E−01	3.59E−02	3.55E−15	8.43E−01	0.00E+00	0.00E+00	0.00E+00
f_{10}	Mean	2.09E−01	4.60E−15	NaN	1.54E+05	1.64E+02	1.72E+00	1.01E−174	2.21E+03	1.46E−180	8.01E−202	**4.15E−285**
	STD	2.18E−02	7.97E−15	NaN	2.76E+05	1.89E+02	3.64E−01	0.00E+00	1.06E+03	0.00E+00	0.00E+00	0.00E+00
f_{11}	Mean	1.71E+08	1.03E+01	2.10E−05	1.73E+07	3.34E+09	4.61E+07	3.26E−172	1.53E+10	3.01E−172	3.93E−195	**1.80E−275**
	STD	1.26E+07	1.78E+01	5.70E−06	1.53E+07	5.77E+09	1.46E+07	0.00E+00	9.41E+09	0.00E+00	0.00E+00	0.00E+00
f_{12}	Mean	1.30E−07	6.15E−07	4.16E−28	6.26E+02	5.33E−07	2.30E−07	4.94E−324	1.68E−03	**0.00E+00**	**0.00E+00**	**0.00E+00**
	STD	6.44E−08	5.46E−07	8.72E−29	8.85E+02	4.02E−07	7.44E−08	0.00E+00	8.82E−04	0.00E+00	0.00E+00	0.00E+00
f_{13}	Mean	2.70E+00	1.34E−01	2.51E−11	3.13E+00	3.09E+01	1.84E+00	**0.00E+00**	1.59E+02	**0.00E+00**	**0.00E+00**	**0.00E+00**
	STD	1.49E−01	1.35E−01	4.18E−12	6.29E−01	5.20E+01	5.00E−02	0.00E+00	5.61E+01	0.00E+00	0.00E+00	0.00E+00
f_{14}	Mean	3.71E+02	8.56E+01	3.25E+01	7.63E+03	2.34E+02	3.91E+01	**0.00E+00**	4.38E+02	**0.00E+00**	**0.00E+00**	**0.00E+00**
	STD	5.01E+01	1.54E+01	3.35E+00	4.25E+01	4.47E+00	2.89E+01	0.00E+00	5.22E+01	0.00E+00	0.00E+00	0.00E+00
f_{15}	Mean	2.41E+04	4.04E−04	9.26E−05	4.88E+09	7.91E+05	1.52E+03	2.26E−167	6.28E+08	2.21E−188	4.10E−11	**0.00E+00**
	STD	1.07E+04	5.93E−04	9.72E−05	4.55E+09	4.83E+05	1.97E+03	0.00E+00	3.16E+08	0.00E+00	7.05E−11	0.00E+00
f_{16}	Mean	2.67E+00	4.27E+00	3.90E+05	5.85E+04	6.66E+02	4.88E+06	9.15E+02	1.06E+03	1.40E−76	4.95E−89	**3.54E−101**
	STD	6.02E−01	1.43E+00	7.42E+04	2.79E+04	2.67E+02	5.56E+06	1.27E+01	4.26E+02	1.96E−76	8.58E−89	1.85E−100

Table 8
Statistical results obtained for the fixed-dimension multimodal functions over different dimensions.

Fun.	Meas.	SA	PSO	CMA-ES	L-SHADE	MFO	EHO	WOA	GOA	TEO	HHO	HBA
f_{17}	Mean	6.17E−10	4.54E−09	3.87E−09	1.13E−07	4.53E−09	4.64E−09	4.53E−09	4.53E−09	4.72E−09	4.54E−09	**1.46E−17**
	STD	1.16E−10	8.14E−12	4.12E−09	1.25E−09	6.45E−2	5.75E−11	3.61E−22	8.14E−12	2.22E−10	5.85E−25	4.90E−17
f_{18}	Mean	2.12E+01	1.24E−04	2.59E−03	1.35E+01	4.16E−02	1.00E+01	2.53E+01	2.13E+01	3.76E+01	1.07E−05	**5.89E−13**
	STD	1.80E+01	3.19E−05	2.08E−03	1.25E+01	3.90E−02	5.89E+00	4.31E+01	1.07E+01	3.60E+00	1.59E−05	2.75E−12
f_{19}	Mean	4.27E+00	2.00E+00	2.00E+00	2.00E+00	2.00E+00	1.10E+00	7.60E−01	1.33E+00	1.03E−02	4.30E−05	**−9.97E−14**
	STD	5.74E+00	0.00E+00	4.12E−08	5.56E−06	0.00E+00	6.09E−01	1.09E+00	1.15E+00	1.36E−02	7.13E−05	3.24E−14
f_{20}	Mean	1.94E−07	**0.00E+00**	3.87E−05	5.90E−09	1.71E−120	8.54E−05	1.58E−70	9.86E−16	1.06E−156	5.60E−272	**0.00E+00**
	STD	1.59E−07	0.00E+00	7.65E−05	1.88E−09	2.96E−120	9.05E−05	2.73E−70	7.35E−16	9.25E−157	0.00E+00	0.00E+00
f_{21}	Mean	1.73E+00	3.46E−11	5.88E−06	8.01E−07	4.65E−08	9.73E+01	1.14E+04	3.21E−06	1.61E−07	4.63E−15	**2.40E−15**
	STD	2.05E+00	2.45E−11	2.63E−06	5.71E−07	5.70E−08	1.28E+02	1.98E+04	5.50E−06	2.77E−07	8.02E−15	6.25E−15
f_{22}	Mean	7.32E+01	**0.00E+00**	3.12E+00	3.66E−01	2.19E−219	3.29E−03	2.52E+00	4.73E−13	3.53E−198	1.83E−226	**0.00E+00**
	STD	7.83E+01	0.00E+00	6.91E+00	8.10E−01	0.00E+00	1.99E−03	4.37E+00	5.07E−13	0.00E+00	0.00E+00	0.00E+00
f_{23}	Mean	5.77E−07	**0.00E+00**	4.89E−150	4.33E−10	3.55E−15	2.32E−05	1.15E−08	1.04E−14	6.34E−202	5.27E−14	**0.00E+00**
	STD	4.91E−07	0.00E+00	5.89E−150	6.22E−10	0.00E+00	4.02E−05	2.00E−08	1.49E−14	0.00E+00	9.12E−14	0.00E+00
f_{24}	Mean	8.28E+04	**0.00E+00**	6.77E−100	4.88E−100	8.17E−229	3.86E−01	6.23E−44	2.09E−10	7.74E−198	2.40E−226	**0.00E+00**
	STD	1.43E+05	0.00E+00	4.87E−100	3.86E−100	0.00E+00	6.30E−01	1.08E−43	7.32E−11	0.00E+00	0.00E+00	0.00E+00

Table 9
P-values at $\alpha = 0.05$ by Friedman test for standard benchmark functions.

Funs.	vs. SA	vs. PSO	vs. CMA-ES	vs. L-SHADE	vs. MFO	vs. EHO	vs. WOA	vs. GOA	vs. TEO	vs. HHO
$D = 30$	1.04E−05	1.04E−05	2.50E−05	1.04E−05	1.04E−05	9.10E−06	4.15E−05	1.04E−05	1.10E−02	1.59E−02
$D = 50$	1.04E−05	1.04E−05	1.04E−05	1.04E−05	1.04E−05	1.04E−05	5.09E−03	1.04E−05	2.48E−02	1.59E−02
Fixed-Dim.	1.70E−03	6.41E−02	1.70E−03	1.70E−03	6.04E−03	1.70E−03	1.70E−03	1.70E−03	6.04E−03	6.04E−03
Overall	2.94E−12	1.01E−10	6.95E−12	2.94E−12	1.00E−11	2.59E−12	7.32E−09	2.94E−12	1.84E−05	1.65E−05

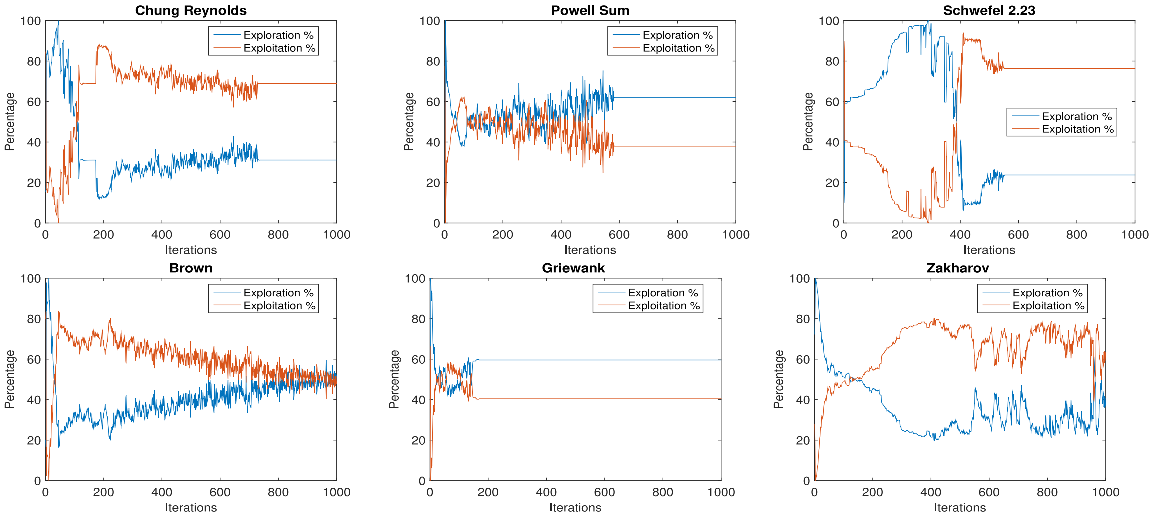


Fig. 8. Exploration and exploitation phases in HBA on the standard unimodal and multimodal functions with 50 dimensions.

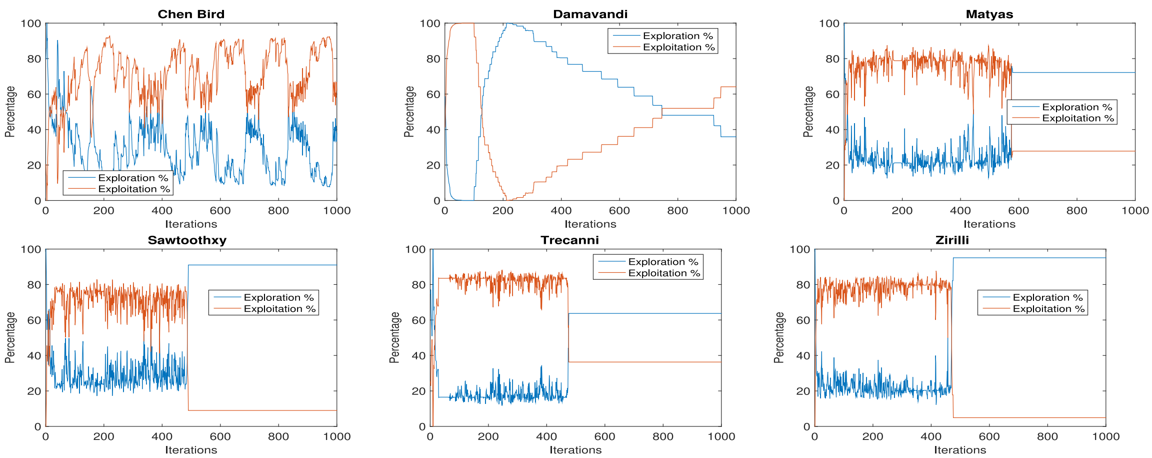


Fig. 9. Exploration and exploitation phases in HBA on the standard fixed-dimensional functions.

which remained second best after HBA maintained exploration lesser than exploitation on almost all the standard benchmark functions. Interestingly on the standard benchmark functions, GOA and CMA-ES maintained exploration extravagantly about 90% or more. Contrarily, L-SHADE maintained exploration–exploitation percentage 2%:98% on all standard benchmark functions. Among the least performers, the ratio of exploration–exploitation maintained by SA was around 70%:30%, PSO with mostly around 35%:65%; while MFO, EHO, WOA maintained relatively varying percentages of exploration and exploitation throughout the specified experiments. The fixed-dimensional results shown in Fig. 11, it is yet affirmed that HBA mostly maintained exploration reasonably high compared to exploitation.

3.3. CEC 2017 test suite analysis

CEC'17 is considered as tough test-bed due to high complexity, hence employed for assessing the quality of HBA. The proposed HBA algorithm was tested on CEC'17 functions with 30 and 50 dimensions, giving promising results comparable to other selected algorithms. The CEC'17 test-suite consists of 29 benchmark functions, as f_2 is excluded from the suite [53]. It comprises of four sets of functions: unimodal functions (f_1 and f_3), multimodal

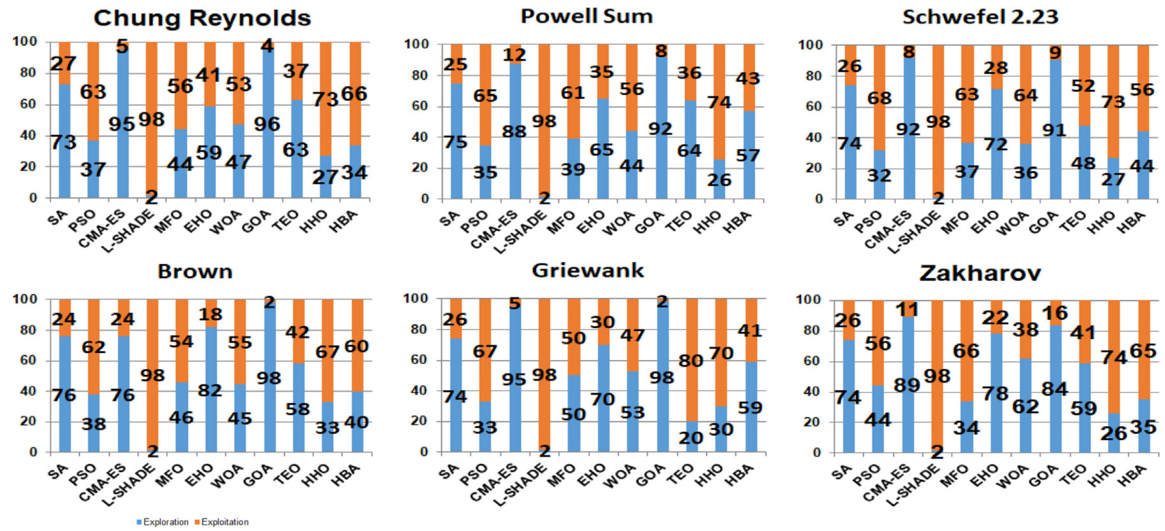


Fig. 10. Exploration and exploitation percentage maintained by metaheuristic algorithms on the standard unimodal and multimodal functions with 50 dimensions.

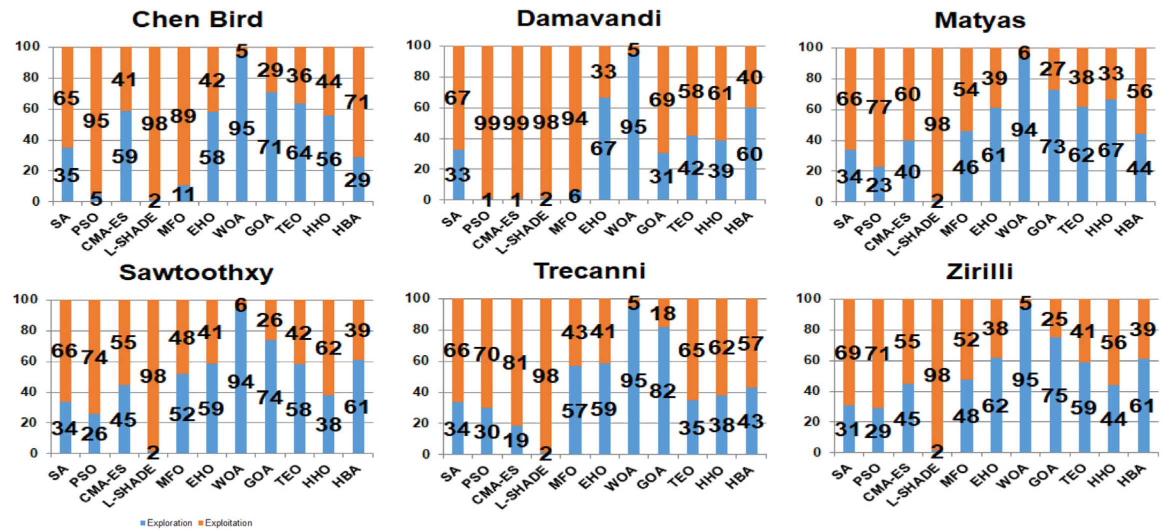


Fig. 11. Exploration and exploitation percentage maintained by metaheuristic algorithms on the standard fixed-dimensional functions.

functions (f_4-f_{10}), hybrid functions (f_{11}-f_{20}), and composition functions (f_{21}-f_{30}). The multimodal functions are used to evaluate the ability to avoid local regions. While, the composite functions resembling the real world optimization problems with highly dynamic search-spaces are employed to investigate the trade-off balance between exploitation and exploration capabilities of the search algorithms.

3.3.1. Statistical results

Tables 10 and 11 report the mean and standard deviation (STD) of the optimal fitness values achieved by the selected algorithms on optimization problems with 30 and 50 dimensions, respectively. The P-values at $\alpha = 0.05$ by Friedman test are shown in Table 12. As results interpret, HBA outperformed the other metaheuristic algorithms on 23 and 19 out of 29 30-D and 50-D problems in CEC'17 test-suite, respectively. The best results are highlighted in boldface. These results suggest that the proposed HBA algorithm can solve a wide variety of optimization problems.

Table 10
Statistical results obtained for the CEC'17 functions with Dim = 30.

Fun.	Meas.	SA	PSO	CMA-ES	L-SHADE	MFO	EHO	WOA	GOA	TEO	HHO	HBA
f_1	Mean	9.79E+10	2.10E+04	5.21E+04	3.52E+09	9.33E+09	5.30E+10	5.77E+08	6.85E+10	5.99E+10	1.98E+07	**1.22E+04**
	STD	2.28E+07	3.27E+04	1.80E+04	1.54E+09	2.75E+09	1.70E+09	1.96E+08	9.67E+09	1.41E+10	4.06E+06	2.58E+04
f_3	Mean	4.28E+05	5.19E+04	3.40E+05	1.34E+05	1.07E+05	8.64E+04	2.73E+05	1.89E+05	8.50E+04	2.80E+04	**1.29E+04**
	STD	2.00E+04	2.36E+03	8.91E+04	3.80E+04	3.67E+04	3.33E+04	2.44E+04	6.76E+04	9.28E+04	8.84E+03	4.81E+03
f_4	Mean	5.58E+02	4.99E+02	**4.12E+02**	9.24E+02	7.75E+02	1.56E+04	8.24E+02	1.27E+04	2.05E+04	5.48E+02	4.78E+02
	STD	3.44E+02	2.33E+01	4.80E−01	2.01E+02	1.02E+01	2.65E+03	6.24E+01	6.66E+03	2.50E+03	1.52E+01	1.95E+01
f_5	Mean	1.21E+03	6.45E+02	6.42E+02	7.63E+02	6.96E+02	9.48E+02	8.17E+02	9.78E+02	9.15E+02	7.42E+02	**6.32E+02**
	STD	1.63E+02	7.24E+01	7.62E+01	3.19E+01	8.77E+00	3.02E+01	4.10E+01	5.34E+01	1.11E+01	3.53E+01	3.50E+01
f_6	Mean	7.32E+02	6.25E+02	**6.00E+02**	6.34E+02	6.31E+02	6.97E+02	6.95E+02	7.09E+02	6.90E+02	6.65E+02	6.43E+02
	STD	1.39E+01	1.31E+01	1.73E−06	6.93E+00	2.13E+01	5.32E+00	8.90E+00	1.58E+01	3.38E+01	4.20E+00	1.65E+01
f_7	Mean	1.29E+03	8.92E+02	8.97E+02	1.09E+03	1.04E+03	1.46E+03	1.26E+03	1.74E+03	1.33E+03	1.28E+03	**8.78E+02**
	STD	3.62E+01	5.62E+01	1.12E+01	3.99E+01	1.62E+02	1.26E+01	8.62E+01	1.31E+02	5.27E+01	4.22E+01	4.09E+01
f_8	Mean	1.24E+03	9.27E+02	9.25E+02	1.06E+03	1.01E+03	1.16E+03	9.91E+02	1.22E+03	1.16E+03	9.52E+02	**9.17E+02**
	STD	4.34E+01	3.31E+01	7.48E+01	2.44E+01	3.22E+01	6.11E+01	1.53E+01	3.77E+01	3.27E+01	1.90E+01	2.48E+01
f_9	Mean	2.49E+04	2.87E+03	**9.00E+02**	4.77E+03	9.14E+03	1.18E+04	1.25E+04	2.52E+04	9.97E+03	6.80E+03	2.80E+03
	STD	4.56E+03	1.27E+03	0.00E+00	1.32E+03	1.36E+03	1.49E+03	1.76E+03	4.34E+03	1.61E+03	1.30E+03	8.38E+02
f_{10}	Mean	1.26E+04	4.89E+03	8.69E+03	9.15E+03	6.17E+03	9.30E+03	7.24E+03	8.21E+03	8.57E+03	5.29E+03	**4.63E+03**
	STD	7.23E+02	1.04E+03	2.53E+02	5.58E+02	3.37E+02	3.29E+02	7.33E+02	1.02E+03	2.76E+02	6.45E+02	6.16E+02
f_{11}	Mean	2.67E+09	**1.20E+03**	6.56E+03	3.27E+03	1.68E+03	8.45E+03	5.35E+03	2.59E+04	1.14E+04	1.27E+03	**1.20E+03**
	STD	7.19E+01	4.05E+00	4.49E+03	1.79E+03	2.41E+02	1.64E+03	1.86E+03	9.03E+03	2.87E+03	3.98E+01	3.07E+01
f_{12}	Mean	2.68E+10	**6.91E+04**	7.97E+06	2.08E+08	2.61E+08	1.36E+10	2.86E+08	1.11E+10	1.77E+10	2.18E+07	6.20E+05
	STD	4.56E+02	8.27E+04	5.29E+06	1.10E+08	2.65E+08	2.29E+09	1.21E+08	4.63E+09	3.99E+09	2.10E+07	7.56E+05
f_{13}	Mean	3.80E+10	1.77E+04	4.41E+06	2.37E+07	2.39E+07	9.48E+09	1.89E+06	9.32E+09	1.66E+10	4.15E+05	**1.29E+04**
	STD	3.12E+00	7.23E+03	2.95E+06	1.66E+07	4.14E+07	2.76E+09	9.80E+05	9.34E+09	2.83E+09	1.29E+05	9.49E+03
f_{14}	Mean	2.07E+09	**2.51E+04**	3.48E+05	1.58E+05	3.32E+05	4.29E+06	5.58E+06	7.05E+06	3.07E+07	4.12E+05	**2.51E+04**
	STD	5.12E+00	1.02E+04	2.92E+05	2.44E+05	3.82E+05	2.64E+06	1.64E+06	7.40E+06	3.22E+07	2.28E+05	2.98E+04
f_{15}	Mean	4.56E+09	1.47E+04	3.08E+06	4.13E+06	6.36E+04	4.42E+08	1.56E+05	5.67E+08	1.26E+09	4.75E+04	**4.49E+03**
	STD	4.23E+01	1.21E+04	1.17E+06	4.49E+06	4.99E+04	2.86E+08	8.27E+04	5.88E+08	7.95E+08	1.50E+04	2.99E+03
f_{16}	Mean	4.00E+04	2.63E+03	2.61E+03	3.97E+03	3.55E+03	5.74E+03	3.90E+03	4.84E+03	6.64E+03	3.50E+03	**2.60E+03**
	STD	3.12E+01	2.73E+02	3.89E+02	2.46E+02	2.15E+02	7.96E+02	5.62E+02	6.23E+02	1.80E+03	2.29E+02	2.93E+02
f_{17}	Mean	2.48E+05	2.23E+03	**2.17E+03**	2.74E+03	2.25E+03	4.22E+03	2.56E+03	4.98E+03	6.86E+03	2.63E+03	**2.17E+03**
	STD	6.12E+00	4.16E+02	1.18E+02	2.33E+02	3.53E+02	5.83E+02	8.85E+01	2.37E+03	3.14E+03	1.90E+02	2.11E+02
f_{18}	Mean	5.86E+09	**5.53E+04**	4.11E+06	2.19E+06	5.68E+06	6.26E+07	1.86E+07	5.09E+07	1.08E+08	7.49E+05	1.94E+05
	STD	8.12e+03	2.44E+04	3.10E+06	2.70E+06	6.16E+06	4.33E+07	1.70E+07	5.69E+07	1.10E+08	3.69E+05	1.91E+05
f_{19}	Mean	3.76E+09	1.39E+04	2.41E+06	5.79E+06	3.06E+07	6.26E+08	2.00E+07	1.94E+09	1.83E+09	6.14E+05	**5.65E+03**
	STD	6.12E+02	9.55E+03	1.57E+06	7.16E+06	3.87E+07	1.86E+08	1.46E+07	1.49E+09	1.35E+09	5.48E+05	3.13E+03
f_{20}	Mean	4.58E+03	2.63E+03	2.56E+03	3.06E+03	2.47E+03	3.12E+03	2.70E+03	3.16E+03	3.27E+03	2.87E+03	**2.44E+03**
	STD	8.12E+00	1.02E+02	1.51E+02	1.79E+02	2.46E+02	2.31E+02	2.38E+02	1.08E+02	1.85E+02	2.04E+02	1.88E+02
f_{21}	Mean	3.18E+03	2.41E+03	**2.40E+03**	2.56E+03	2.53E+03	2.78E+03	2.63E+03	2.72E+03	2.81E+03	2.58E+03	**2.40E+03**
	STD	6.12E+00	2.99E+01	6.76E+01	2.01E+01	3.30E+01	7.91E+01	1.04E+02	3.21E+01	8.42E+01	6.17E+01	6.29E+01
f_{22}	Mean	1.23E+04	3.40E+03	9.85E+03	5.24E+03	6.96E+03	9.61E+03	5.87E+03	1.02E+04	9.52E+03	7.18E+03	**2.45E+03**
	STD	7.15E+01	1.91E+03	4.03E+02	2.61E+03	5.42E+02	1.12E+02	2.82E+03	8.49E+02	1.97E+02	1.28E+03	8.28E+02
f_{23}	Mean	7.62E+03	2.75E+03	**2.71E+03**	2.93E+03	2.78E+03	3.63E+03	3.16E+03	3.27E+03	3.86E+03	3.12E+03	2.82E+03
	STD	1.11E−12	4.42E+01	7.09E+01	2.95E+01	1.30E+01	1.30E+02	1.03E+02	7.40E+01	2.93E+02	8.31E+01	5.09E+01
f_{24}	Mean	5.31E+03	2.97E+03	**2.86E+03**	3.09E+03	3.01E+03	3.87E+03	3.19E+03	3.25E+03	4.57E+03	3.33E+03	3.01E+03
	STD	4.5E+00	7.64E+01	5.53E+01	2.87E+01	1.25E+01	1.00E+02	4.31E+01	1.51E+02	2.37E+02	1.62E+02	4.27E+01
f_{25}	Mean	7.71E+03	2.99E+03	**2.89E+03**	3.16E+03	3.10E+03	5.38E+03	3.07E+03	1.07E+03	6.12E+03	2.91E+03	**2.89E+03**
	STD	6.78E+01	1.19E+02	1.10E−01	7.36E+01	1.66E+02	1.63E+02	8.27E+01	8.37E+03	4.43E+02	1.76E+01	1.39E+01
f_{26}	Mean	1.77E+04	5.08E+03	4.15E+03	6.68E+03	5.70E+03	1.14E+04	8.57E+03	9.54E+03	1.11E+04	7.87E+03	**4.01E+03**
	STD	9.67E+01	1.94E+02	6.16E+02	2.92E+02	3.67E+02	3.36E+02	1.95E+02	1.29E+03	7.62E+02	9.10E+02	9.25E+02
f_{27}	Mean	1.11E+04	3.28E+03	3.20E+03	3.31E+03	3.26E+03	5.00E+03	3.84E+03	3.87E+03	5.68E+03	3.41E+03	**3.20E+03**
	STD	6.34E+01	6.17E+01	7.45E−05	2.98E+01	3.62E+01	4.35E+02	9.98E+01	4.50E+02	6.51E+02	8.21E+01	3.09E−04
f_{28}	Mean	9.55E+03	**3.26E+03**	3.30E+03	3.68E+03	4.42E+03	7.32E+03	3.53E+03	7.02E+03	8.41E+03	3.40E+03	**3.26E+03**
	STD	7.39E+01	4.38E+01	9.17E−05	1.68E+02	7.11E+02	3.78E+02	1.30E+02	9.42E+02	6.39E+02	1.19E+02	4.26E+01
f_{29}	Mean	5.50E+05	3.72E+03	4.12E+03	4.80E+03	3.99E+03	7.72E+03	5.57E+03	6.18E+03	1.97E+04	4.56E+03	**3.62E+03**
	STD	8.50E+01	2.67E+02	2.32E+02	2.72E+02	1.63E+02	1.49E+03	8.30E+02	9.30E+02	9.27E+03	4.94E+02	2.02E+02
f_{30}	Mean	1.10E+10	6.91E+03	1.86E+06	1.24E+07	8.14E+05	1.61E+09	5.80E+07	9.49E+08	5.38E+09	4.03E+06	**6.49E+03**
	STD	4.44E+06	4.13E+02	1.01E+06	1.04E+07	8.44E+05	5.95E+08	2.50E+07	6.02E+08	2.19E+09	1.60E+06	8.04E+03

3.3.2. Convergence analysis

To further understand the results provided in Tables 10 and 11, Figs. 12 and 13 provide convergence abilities of the selected algorithms while solving 30-D and 50-D CEC'17 optimization problems, respectively. Note that Figs. 12 and 13 illustrate convergence graphs for selected functions, like Shifted and Rotated Bent Cigar Function (f_1), Shifted and Rotated Expanded Scaffer's F6 Function (f_6), Hybrid Function 1 (N = 3) (f_{11}), Hybrid Function 6 (N = 4) (f_{16}), Composition Function 1 (N = 3) (f_{21}), and Composition Function 6 (N = 5) (f_{26}). Similar to efficient convergence ability shown by HBA on standard benchmark functions, on complex functions of CEC'17 also the proposed method converged to better optimal locations as compared to the counterparts selected in this study.

Table 11
Statistical results obtained for the CEC’17 functions with Dim = 50.

Fun.	Meas.	SA	PSO	CMA-ES	L-SHADE	MFO	EHO	WOA	GOA	TEO	HHO	HBA
f_1	Mean	1.47E+11	9.39E+05	**3.16E+04**	2.01E+10	2.77E+10	1.19E+11	4.59E+09	1.25E+11	1.15E+11	1.17E+08	9.19E+05
	STD	0.00E+00	6.08E+03	3.72E+04	5.15E+09	2.32E+09	4.94E+09	1.43E+09	2.34E+10	7.85E+09	1.69E+07	2.13E+06
f_3	Mean	3.05E+14	7.99E+04	7.08E+05	2.89E+05	3.28E+05	4.00E+05	2.42E+05	5.44E+05	1.95E+05	9.59E+04	**7.97E+04**
	STD	0.00E+00	1.57E+04	2.15E+05	7.77E+04	7.85E+04	3.71E+05	1.00E+05	1.51E+04	5.63E+03	1.93E+04	1.02E+04
f_4	Mean	1.59E+04	5.79E+02	5.77E+02	3.16E+03	4.07E+03	3.37E+04	1.07E+03	3.73E+04	3.84E+04	7.14E+02	**5.73E+02**
	STD	6.34E+02	4.21E+01	6.23E+01	8.25E+02	1.59E+03	4.13E+03	1.88E+02	7.63E+03	3.10E+03	7.42E+01	4.94E+01
f_5	Mean	1.54E+03	7.89E+02	**5.31E+02**	1.02E+03	9.35E+02	1.19E+03	9.78E+02	1.42E+03	1.17E+03	8.85E+02	7.84E+02
	STD	3.45E+02	3.23E+01	8.17E+00	4.15E+01	3.27E+01	4.09E+01	5.31E+01	1.68E+02	6.09E+01	4.52E+01	3.94E+01
f_6	Mean	7.12E+02	6.61E+02	6.00E+02	6.56E+02	6.6E+02	7.09E+02	6.96E+02	7.22E+02	7.08E+02	6.74E+02	**6.58E+02**
	STD	1.08E+01	8.05E+00	2.25E−05	7.45E+00	7.54E+00	1.62E+00	7.84E+00	1.05E+01	5.46E+00	4.27E+00	4.79E+00
f_7	Mean	1.39E+03	**9.14E+02**	9.22E+02	1.58E+03	1.62E+03	2.08E+03	1.90E+03	2.42E+03	2.04E+03	1.88E+03	1.11E+03
	STD	1.81E+01	3.42E+01	1.47E+02	9.51E+01	4.02E+02	1.06E+01	4.63E+01	8.49E+01	6.20E+01	1.46E+02	1.06E+02
f_8	Mean	1.93E+03	1.12E+03	8.43E+02	1.32E+03	1.29E+03	1.52E+03	1.28E+03	1.63E+03	1.53E+03	1.19E+03	**1.06E+03**
	STD	4.14E+02	6.41E+01	5.99E+01	3.84E+01	5.09E+01	4.21E+01	5.88E+01	7.26E+01	7.43E+01	1.79E+01	4.75E+01
f_9	Mean	8.80E+04	9.73E+03	**9.00E+02**	1.79E+04	2.07E+04	4.19E+04	3.30E+04	6.32E+04	3.57E+04	2.80E+04	1.14E+04
	STD	1.80E+04	2.17E+03	7.45E−01	6.32E+03	6.97E+03	7.70E+02	1.22E+03	4.76E+03	1.80E+02	2.08E+03	4.53E+03
f_{10}	Mean	9.44E+03	7.83E+03	1.53E+04	1.58E+04	9.05E+03	1.57E+04	1.23E+04	1.50E+04	1.55E+04	9.24E+03	**7.35E+03**
	STD	3.48E+02	1.34E+03	5.22E+02	3.28E+02	8.77E+02	3.64E+02	1.21E+03	2.14E+02	2.04E+02	7.47E+02	9.27E+02
f_{11}	Mean	1.54E+03	1.39E+03	9.52E+04	1.44E+04	2.10E+04	2.26E+04	3.78E+03	3.47E+04	2.81E+04	1.65E+03	**1.38E+03**
	STD	5.20E+01	5.77E+01	3.12E+04	2.49E+03	2.00E+04	3.41E+03	9.27E+01	1.22E+04	3.31E+03	1.88E+02	5.57E+01
f_{12}	Mean	1.27E+08	7.44E+06	7.34E+06	2.61E+09	4.36E+09	7.74E+10	9.13E+08	6.08E+10	1.04E+11	1.80E+08	**7.08E+06**
	STD	3.54E+07	1.76E+05	2.36E+06	1.08E+09	3.91E+09	6.08E+09	2.27E+08	4.44E+10	5.08E+09	1.06E+08	1.30E+07
f_{13}	Mean	1.21E+07	**4.11E+03**	5.59E+06	3.41E+08	8.52E+08	4.79E+10	3.07E+07	1.64E+10	5.65E+10	1.67E+07	2.03E+04
	STD	1.54E+06	1.18E+03	3.55E+06	1.85E+08	9.08E+08	1.49E+09	2.28E+07	1.09E+10	2.32E+09	2.32E+07	1.37E+04
f_{14}	Mean	**6.20E+04**	7.33E+04	1.01E+06	2.09E+06	3.47E+06	8.19E+07	4.48E+06	7.56E+07	3.69E+08	3.44E+06	3.21E+05
	STD	4.37E+04	4.72E+04	4.06E+05	1.71E+06	6.61E+05	3.52E+07	3.49E+06	9.33E+07	2.34E+08	3.80E+06	3.98E+05
f_{15}	Mean	2.26E+06	**3.00E+04**	2.73E+06	3.73E+07	7.01E+04	8.27E+09	6.52E+06	9.81E+09	1.41E+10	6.88E+05	**3.00E+04**
	STD	3.58E+05	9.38E+03	1.35E+06	2.75E+07	3.06E+04	4.20E+08	3.81E+06	5.51E+09	3.06E+09	1.23E+05	2.82E+04
f_{16}	Mean	4.16E+03	3.96E+03	**2.02E+03**	5.83E+03	4.35E+03	1.06E+04	5.68E+03	8.80E+03	1.06E+04	4.03E+03	3.34E+03
	STD	1.66E+02	3.91E+02	1.82E+02	3.75E+02	2.41E+02	2.23E+03	2.47E+02	6.30E+02	1.25E+03	3.74E+02	3.00E+02
f_{17}	Mean	4.16E+03	**2.85E+03**	3.16E+03	4.73E+03	3.83E+03	9.69E+03	3.90E+03	1.37E+05	4.13E+04	3.88E+03	3.11E+03
	STD	3.94E+02	9.47E+01	2.97E+02	4.89E+02	7.09E+02	3.05E+03	5.20E+02	2.09E+05	1.61E+04	1.54E+02	2.75E+02
f_{18}	Mean	6.82E+05	**5.57E+05**	1.01E+07	6.84E+06	8.78E+06	1.51E+08	2.52E+07	4.22E+08	2.48E+08	1.58E+06	1.10E+06
	STD	1.96E+05	1.50E+05	2.32E+06	4.47E+06	9.22E+05	3.90E+07	2.00E+07	5.86E+07	1.41E+08	5.41E+05	7.21E+05
f_{19}	Mean	5.83E+06	1.59E+04	8.90E+05	2.27E+07	3.87E+05	2.98E+09	4.84E+06	3.70E+09	5.05E+09	7.17E+05	**1.33E+04**
	STD	1.16E+06	1.04E+04	4.01E+05	1.26E+07	4.22E+05	1.27E+09	5.03E+06	1.39E+09	2.71E+09	9.33E+05	8.77E+03
f_{20}	Mean	4.30E+03	3.42E+03	3.53E+03	4.62E+03	3.58E+03	4.37E+03	3.85E+03	4.80E+03	4.34E+03	3.81E+03	**3.13E+03**
	STD	5.73E+02	5.25E+02	3.85E+02	1.52E+02	4.97E+02	7.88E+01	6.46E+02	1.26E+02	2.23E+02	4.40E+02	2.97E+02
f_{21}	Mean	3.45E+03	2.76E+03	2.73E+03	2.83E+03	2.74E+03	3.32E+03	2.89E+03	3.23E+03	3.36E+03	2.84E+03	**2.55E+03**
	STD	2.13E+02	6.03E+01	7.44E+01	4.03E+01	5.86E+01	2.24E+01	7.11E+01	7.60E+01	5.11E+01	3.34E+00	5.50E+01
f_{22}	Mean	1.19E+04	1.02E+04	1.64E+04	1.77E+04	1.05E+04	1.73E+04	1.38E+04	1.74E+04	1.70E+04	1.11E+04	**8.37E+03**
	STD	9.23E+02	7.20E+02	3.18E+03	1.69E+03	9.70E+02	2.83E+02	5.22E+02	3.99E+02	7.12E+02	1.92E+02	2.94E+03
f_{23}	Mean	5.29E+03	3.02E+03	3.02E+03	3.39E+03	3.10E+03	4.56E+03	3.73E+03	3.69E+03	4.87E+03	3.86E+03	**3.00E+03**
	STD	2.71E+02	1.29E+02	1.20E+02	6.21E+01	2.92E+01	2.93E+01	1.09E+02	7.22E+01	2.19E+02	2.55E+01	9.49E+01
f_{24}	Mean	3.92E+03	3.22E+03	**2.93E+03**	3.51E+03	3.18E+03	5.20E+03	3.75E+03	3.90E+03	5.51E+03	4.42E+03	3.35E+03
	STD	1.09E+02	6.17E+01	5.02E+00	5.32E+01	5.81E+01	1.38E+02	2.66E+02	3.19E+02	1.90E+02	3.29E+02	9.50E+01
f_{25}	Mean	**3.01E+03**	3.07E+03	3.05E+03	5.26E+03	5.03E+03	1.56E+04	3.73E+03	2.27E+04	1.60E+04	3.21E+03	3.15E+03
	STD	3.30E+01	3.90E+01	3.21E+02	4.81E+02	2.68E+03	1.57E+03	1.72E+02	4.36E+03	1.12E+03	5.63E+01	4.25E+01
f_{26}	Mean	1.45E+04	**3.95E+03**	4.44E+03	9.69E+03	9.38E+03	1.83E+04	1.48E+04	2.16E+04	1.79E+04	1.08E+04	6.35E+03
	STD	2.76E+03	1.82E+03	7.87E+01	3.21E+02	5.52E+02	4.19E+02	1.73E+03	1.23E+03	9.44E+02	1.05E+03	2.46E+03
f_{27}	Mean	3.28E+03	3.54E+03	3.22E+03	3.89E+03	3.45E+03	7.60E+03	4.47E+03	5.59E+03	8.39E+03	4.31E+03	**3.20E+03**
	STD	1.37E+02	9.25E+01	7.14E+03	6.74E+01	5.12E+01	7.07E+02	8.15E+02	1.40E+03	8.83E+02	4.19E+02	4.03E−04
f_{28}	Mean	**3.30E+03**	3.33E+03	3.32E+03	5.41E+03	6.69E+03	1.26E+04	4.56E+03	1.24E+04	1.46E+04	3.56E+03	**3.30E+03**
	STD	1.34E+01	7.99E+00	1.77E+01	1.08E+03	2.47E+03	1.18E+03	6.30E+02	1.42E+03	3.49E+02	5.28E+01	5.65E+01
f_{29}	Mean	5.74E+03	4.50E+03	4.81E+03	6.81E+03	5.88E+03	6.53E+04	8.90E+03	1.98E+04	5.05E+05	5.92E+03	**4.31E+03**
	STD	4.26E+02	5.20E+02	2.92E+02	3.58E+02	3.85E+02	1.44E+04	1.20E+03	8.52E+03	5.44E+05	7.07E+02	4.41E+02
f_{30}	Mean	4.11E+07	9.20E+05	3.84E+06	2.57E+08	1.16E+07	6.53E+09	2.27E+08	8.40E+09	9.66E+09	3.38E+07	7.74E+03
	STD	1.56E+06	1.36E+05	8.23E+05	8.58E+07	3.58E+06	2.12E+09	8.23E+07	7.49E+09	3.64E+09	6.16E+06	**3.93E+03**

Table 12
P-values at $\alpha = 0.05$ by Friedman test for CEC’17 functions.

Funs.	vs. SA	vs. PSO	vs. CMA-ES	vs. L-SHADE	vs. MFO	vs. EHO	vs. WOA	vs. GOA	vs. TEO	vs. HHO
$D = 30$	3.10E−07	4.26E−02	2.58E−02	2.41E−06	2.35E−05	1.05E−07	2.41E−06	2.41E−06	1.05E−07	3.89E−05
$D = 50$	2.76E−04	3.14E−01	7.63E−01	1.60E−05	9.11E−05	8.81E−07	1.60E−05	2.41E−06	8.81E−07	4.45E−04
Overall	5.89E−10	3.21E−02	7.46E−02	1.71E−10	8.59E−09	4.60E−13	1.71E−10	2.57E−11	4.60E−13	6.97E−08

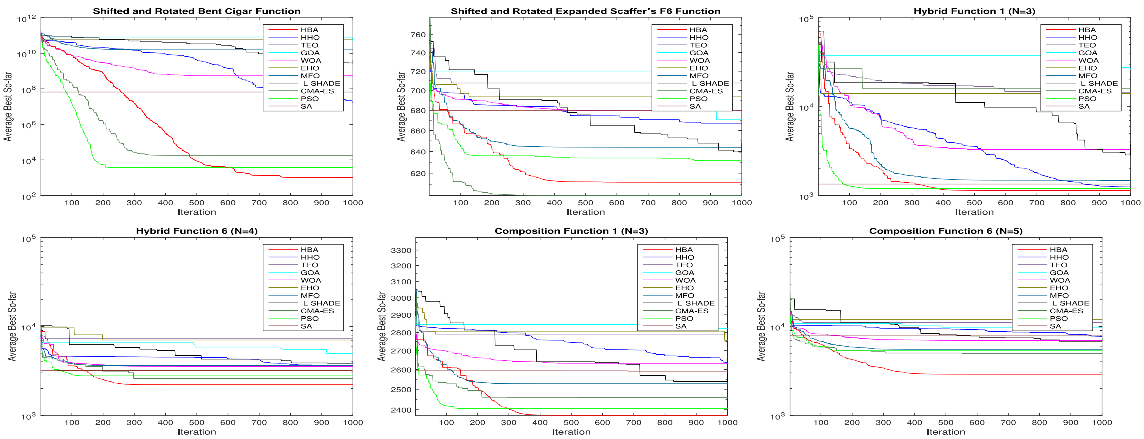


Fig. 12. Convergence curves of competitor algorithms on CEC'17 functions with 30 dimensions.

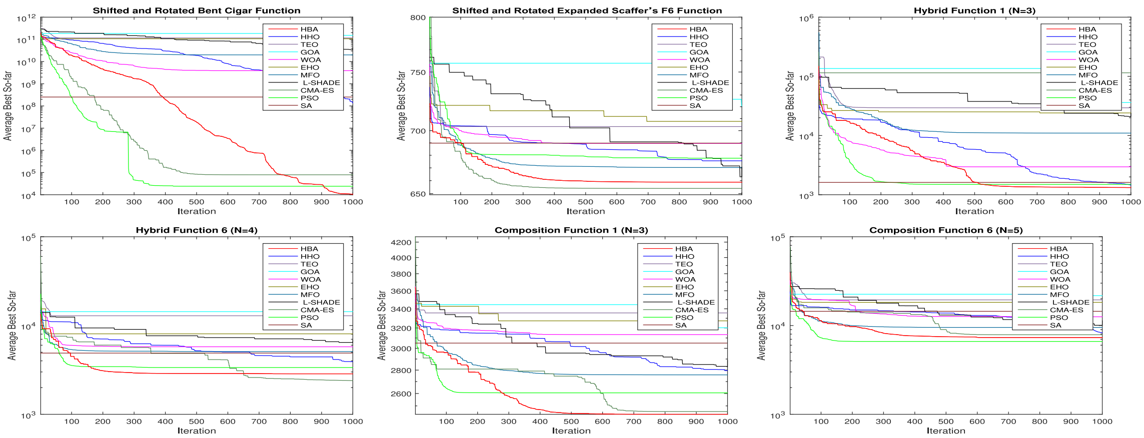


Fig. 13. Convergence curves of competitor algorithms on CEC'17 functions with 50 dimensions.

3.3.3. Exploration–exploitation analysis

The in-depth information about HBA search efficiency is provided by Fig. 14 which demonstrates exploration and exploitation behavior of the algorithm during search process while solving hard optimization problems mentioned in CEC'17 test-suite. Fig. 14 presents exploration–exploitation illustrated by HBA on selected functions, like Shifted and Rotated Bent Cigar Function (f_1), Shifted and Rotated Expanded Scaffer's F6 Function (f_6), Hybrid Function 1 (N = 3) (f_{11}), Hybrid Function 6 (N = 4) (f_{16}), Composition Function 1 (N = 3) (f_{21}), and Composition Function 6 (N = 5) (f_{26}). The linear graphs show the two important capabilities shown by HBA on CEC'17 functions with 50 dimensions. It is clear from the graphs that HBA mostly started with high exploration and low exploitation, but as the search progresses, the HBA algorithm increased its convergence towards promising regions by ensuring effective exploitation strategy. However, it is also clear that HBA maintained divergent population throughout iterations to ensure avoidance of local optimum regions. The vibrant exploration and exploitation ratios during iterations show that HBA maintained explorative strategy up to certain level, hence found better optimal solutions compared to the algorithms selected in this study. This magnifies the importance of trade-off balance between exploration and exploitation even towards the end of search process.

Furthermore, the overall exploration and exploitation percentage ratios maintained by HBA and other algorithms are displayed in Fig. 15. It can be suggested that HBA mostly maintained overall exploitation percentage higher than exploration CEC'17 test-suite. That said, HBA maintained exploration percentage ratio adequately higher, more than around 40% for avoiding trapping in local regions. On the other hand, exploitation percentage ratio was higher

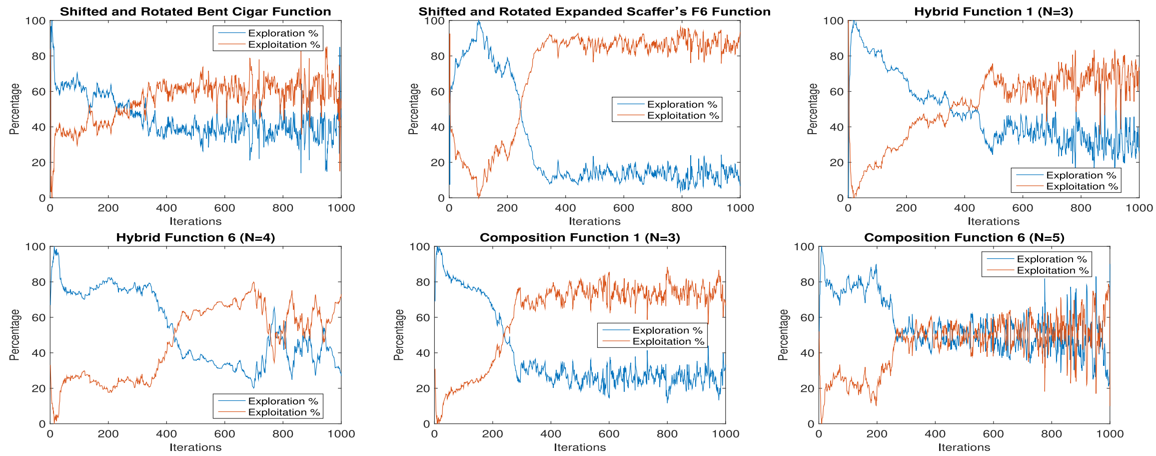


Fig. 14. Exploration and exploitation phases in HBA on the CEC'17 functions with 50 dimensions.

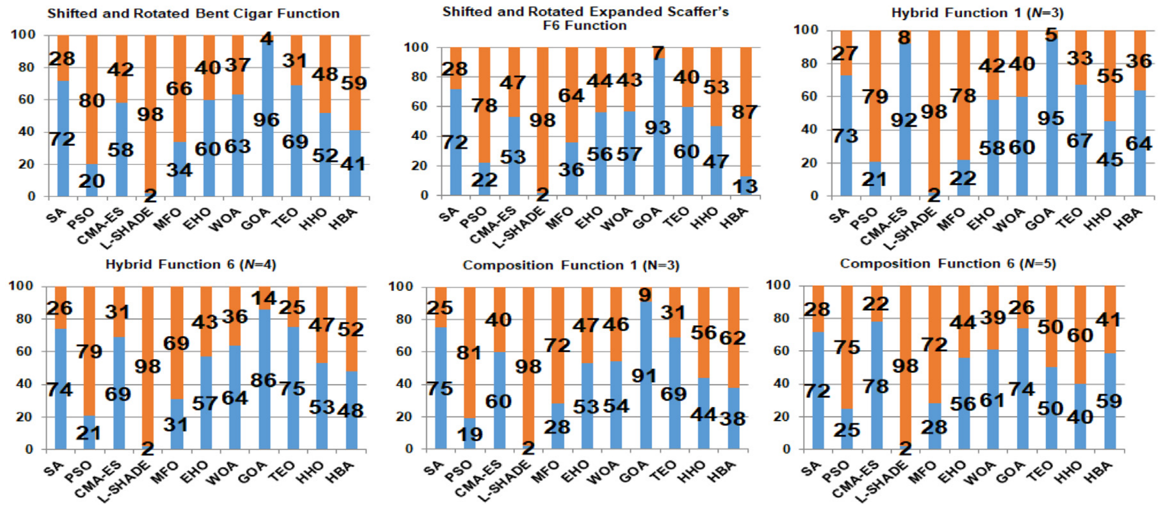
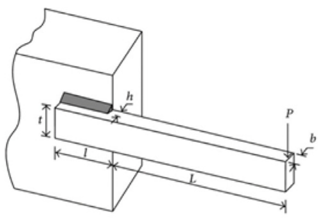


Fig. 15. Average exploration and exploitation for the competitor algorithms on the CEC'17 test suite.

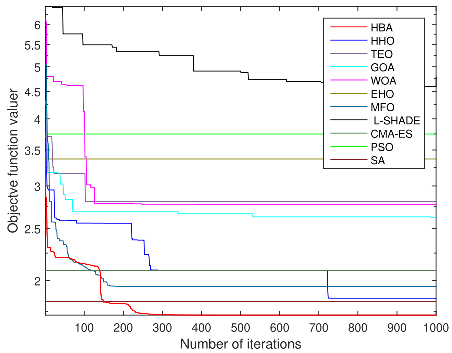
80% on Shifted and Rotated Expanded Scaffer's F6 Function (f_6). Interestingly, the stacked bar charts of HBA and HHO show not much difference in exploration–exploitation behavior of the algorithms. The TEO algorithm which also produced promising results after HBA and HHO remained more explorative approach on most of the CEC'17 functions where exploration percentage was higher than exploitation. Among the lowest performers, GOA was highly explorative algorithm, even its exploration percentage was 90% or above on 10 functions. PSO and MFO were among the methods where exploration percentage was highly dominated by exploitation percentage, hardly exploration jumped above 30% in these functions. CMA-ES showed different exploration/exploitation ratios on all 50-dimensional CEC'17 functions. The L-SHADE algorithm, similar to performance on standard benchmark functions, maintained 2%:98% exploration–exploitation ratio on all of test functions on this test-suit as well.

4. Experimental results on real-world engineering problems

Generally, optimization of engineering designs is a potential research area, where several optimization approaches have been investigated [48]. In our study, three engineering design problems were solved by HBA. Following



(a) Schematic design.



(b) Function values versus number of iterations.

Fig. 16. Welded beam design problem.

Table 13
The best solution obtained from competitor algorithms for the welded beam problem.

Algorithm	h	l	t	b	f_{cost}
SA	0.2037	3.5260	9.0488	0.2058	1.73160
PSO	0.3952	2.5657	9.3772	0.4165	3.55553
CMA-ES	0.5617	4.3786	4.6772	0.9286	2.28384
L-SHADE	0.4819	3.2140	5.4763	0.5753	3.43372
MFO	0.2057	3.4705	9.0366	0.2057	1.73485
EHO	1.0149	4.7616	4.8130	0.8722	3.36770
WOA	0.1972	4.0391	8.2355	0.2802	2.17591
GOA	0.4069	2.1411	6.3834	0.4123	2.43534
TEO	0.2539	3.5440	7.4619	0.3040	3.31491
HHO	0.1961	3.7449	9.0061	0.2071	1.75163
HBA	0.2057	3.4704	9.0366	0.2057	**1.72451**

subsections present detail of the engineering design problems, as well as, results achieved by the metaheuristic algorithms. For mathematical expressions of these engineering design problems, [48] can be referred.

4.1. Welded beam design problem

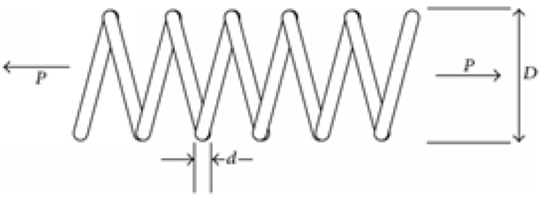
Coello [8] first proposed this problem, since then it is used as benchmark for performance evaluation of optimization methods. Here, the cost of a welded beam design (Fig. 16) is minimized keeping in view certain constraints.

The function values of HBA during iterations are shown in Fig. 16b. It is clear from the convergence graph that the design cost is reduced to near optimal cost at early iterations. The best solutions achieved by HBA and other algorithms are presented in Table 13. Additionally, a comparison of the statistical results is given in Table 14, the mean solution obtained by HBA is the best among all other counterparts.

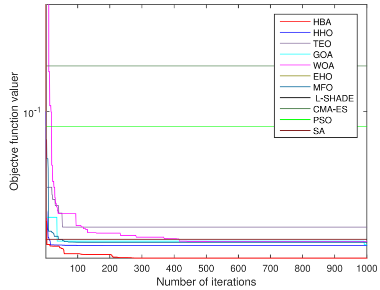
4.2. Tension/compression spring design problem

The detail of this problem is well explained in [10]. It is also a cost minimization problem based on certain constraints as shown in Fig. 17.

The convergence ability of HBA and other algorithms is illustrated in Fig. 17b. The best results produced by HBA and other approaches are compared in Table 15. While, the statistical results comparison is made in Table 16. The results indicate that the best solution is obtained by HBA.



(a) Schematic design.



(b) Function values versus number of iterations.

Fig. 17. Tension/compression string design problem.

Table 14
The results obtained from competitor algorithms for the welded beam problem.

Algorithm	Best	Mean	Worst	STD
SA	1.73160	1.81150	1.85517	1.33E−01
PSO	3.55553	4.63E+04	1.19E+05	6.34E+04
CMA-ES	2.28384	6.18E+03	1.85E+04	1.06E+03
L-SHADE	3.43372	3.90E+04	1.17E+05	6.75E+04
MFO	1.72485	1.87579	2.22657	1.95E−01
EHO	3.36770	5.21E+03	1.56E+04	9.02E+03
WOA	2.17591	2.74682	3.14529	1.88E−01
GOA	2.43534	3.22555	3.82049	7.98E−01
TEO	3.31491	3.73222	3.87797	5.91E−01
HHO	1.75163	1.85755	2.01234	2.59E−02
HBA	1.72085	1.72485	1.724854	9.18E−10

Table 15
The best solution obtained from competitor algorithms for the tension/compression spring problem.

Algorithm	d	D	N	f_{cost}
SA	0.0570	0.4953	6.2225	0.01321
PSO	0.0701	0.9605	2.0000	0.01884
CMA-ES	0.0973	1.1488	13.54530	0.85621
L-SHADE	0.0839	0.93420	4.50460	0.04275
MFO	0.0523	0.3715	10.4720	0.01272
EHO	0.1047	1.1992	7.9896	0.10478
WOA	0.0523	0.3720	10.447	0.01269
GOA	0.0516	0.3360	13.500	0.01389
TEO	0.0583	0.5360	5.3885	0.01346
HHO	0.0570	0.4991	6.2180	0.01281
HBA	0.0506	0.3552	11.373	**0.01207**

4.3. Speed reducer design problem

In this constrained optimization problem (Fig. 18), the weight of speed reducer is minimized with help of certain design variable and constraints [41].

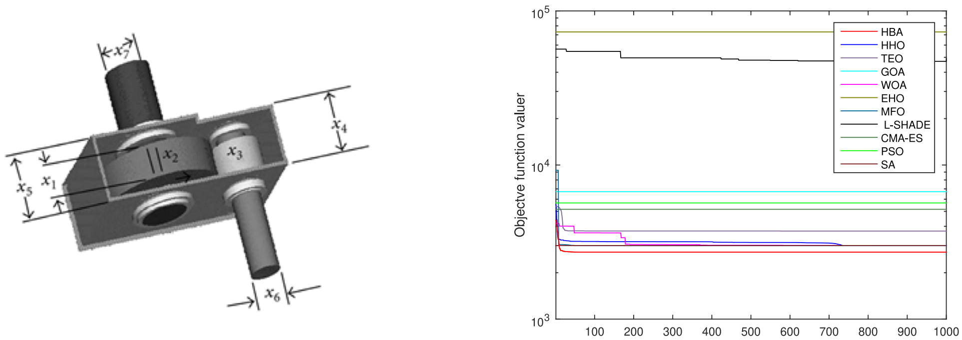
The best solutions found by the competitor algorithms during search process are shown in Fig. 18b. Table 17 provides a comparison of the best solutions found by the selected methods. While, the statistical results are compared in Table 18. According to results, the best solution is achieved by HBA.

Table 16
The results obtained from competitor algorithms for the tension/compression spring problem.

Algorithm	Best	Mean	Worst	STD
SA	0.01321	0.01554	0.0175	2.18E−03
PSO	0.01887	0.01952	0.0200	6.01E−04
CMA-ES	0.85621	1.02626	2.7953	4.72E+01
L-SHADE	0.04275	0.08951	2.1117	1.20E+02
MFO	0.01272	0.01332	0.0146	1.51E−03
EHO	0.10478	1.19922	7.9896	4.61E+01
WOA	0.01267	0.01399	0.0170	1.09E−03
GOA	0.01389	0.01594	0.0178	1.97E−03
TEO	0.01346	8.95E+04	9.31E+05	2.53E+05
HHO	0.01281	0.01394	0.0168	9.01E−04
HBA	0.01201	0.01270	0.0128	6.47E−05

Table 17
The best solution obtained from the competitor algorithms for the speed reducer problem.

Algorithm	X_1	X_2	X_3	X_4	X_5	X_6	X_7	f_{cost}
SA	3.4979	0.7000	17.0000	7.9205	7.9513	3.3518	5.2853	3.00485E+03
PSO	3.5452	0.7362	21.3911	7.4170	8.2339	3.7644	5.3590	6.07692E+03
CMA-ES	2.6000	0.8000	17.0000	7.3000	7.8000	2.9000	5.0000	8.96248E+03
L-SHADE	3.4367	0.7179	17.2544	8.1541	7.9808	3.2999	5.3498	7.36125E+03
MFO	3.4976	0.7000	17.0000	7.3000	7.8000	3.3501	5.2857	2.99854E+03
EHO	3.4889	0.7782	23.2193	7.8490	8.1021	3.5603	5.2459	7.35047E+04
WOA	3.4976	0.7000	17.0000	7.8041	7.8000	3.3965	5.2854	2.99703E+03
GOA	3.5126	0.7033	17.2246	7.9131	7.9627	3.6567	5.2784	3.16932E+03
TEO	3.4261	0.7000	17.6222	7.7408	7.9775	3.4145	5.2758	3.59559E+03
HHO	3.4965	0.7000	17.0000	7.3000	7.8000	3.3519	5.2856	2.99710E+03
HBA	3.4976	0.7000	17.0000	7.3000	7.8000	3.3501	5.2857	**2.9554E+03**



(a) Schematic design.

(b) Function values versus number of iterations.

Fig. 18. Speed reducer design problem.

4.4. Pressure vessel design problem

In the pressure vessel design problem, proposed by Kannan and Kramer [33], the aim is to minimize the total cost, including the cost of material, forming, and welding. A cylindrical vessel (four design variables) is capped at both ends by hemispherical heads as shown in Fig. 19.

The best solutions found by the selected methods during 1000 iterations are plotted in Fig. 19b. Table 19 presents a comparison of the best solutions found by the algorithms. The statistical results are compared in Table 20. As shown in Table 20, the best mean value is generated by HBA.

Table 18
The results obtained from competitor algorithms for the speed reducer problem.

Algorithm	Best	Mean	Worst	STD
SA	3.00485E+03	3.00579257E+03	3.0070E+03	1.13E+00
PSO	6.07692E+03	1.12076129E+04	1.8777E+04	6.69E+03
CMA-ES	7.36125E+03	5.34506364E+04	1.0478E+05	4.89E+04
L-SHADE	7.36125E+03	5.34506364E+04	1.0478E+05	4.89E+04
MFO	2.99554E+03	2.99554243E+03	2.9955E+03	3.54E−12
EHO	7.35047E+04	8.86717739E+04	1.0100E+05	1.40E+04
WOA	2.99603E+03	2.99603122E+03	2.9960E+03	1.03E+03
GOA	3.16932E+03	3.94974122E+03	5.4067E+03	1.26E+03
TEO	3.59559E+03	4.03088593E+03	4.6646E+03	5.61E+02
HHO	2.99610E+03	3.00225966E+03	2.9961E+03	8.15E+00
HBA	2.59554E+03	2.99554243E+03	2.9955E+03	2.311E−12

Table 19
The best solution obtained from the competitor algorithms for the pressure vessel design problem.

Algorithm	$X-1$	X_2	X_3	X_4	f_{cost}
SA	0.9101	0.4501	47.1253	123.460	6.171595E+03
PSO	1.7621	1.6724	74.9999	30.2758	2.413495E+04
CMA-ES	1.8990	1.5064	88.7274	50.8583	7.976024E+03
L-SHADE	1.7913	0.6373	61.4220	29.4625	1.050206E+04
MFO	0.8264	0.4077	42.9043	166.905	5.967881E+03
EHO	2.5816	1.4787	47.1647	148.769	1.998475E+04
WOA	0.9663	0.5025	50.1186	97.4080	6.399544E+03
GOA	0.9783	0.4842	46.4507	152.987	7.534568E+03
TEO	1.0553	1.9922	54.9284	64.7313	1.454789E+04
HHO	0.8063	0.3927	41.6113	182.771	5.956738E+03
HBA	0.7764	0.3832	40.3196	200.000	**1.27672E+03**

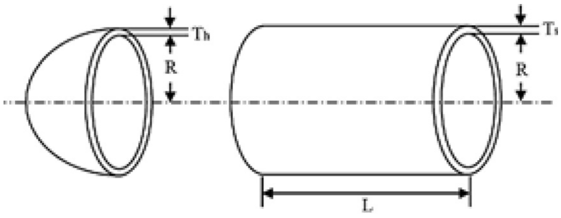
Table 20
The results obtained from competitor algorithms for the pressure vessel design problem.

Algorithm	Best	Mean	Worst	STD
SA	6.1715995E+03	6.43011666E+03	6.81221848E+03	3.38E+02
PSO	2.4134995E+04	2.69624661E+04	3.02532553E+04	3.09E+03
CMA-ES	7.9760241E+03	8.29566995E+03	2.48872542E+04	1.44E+03
L-SHADE	1.05020587E+04	3.9032666E+04	6.76215214E+04	2.86E+04
MFO	5.9678281E+03	6.54587749E+03	7.30294401E+03	4.14E+02
EHO	1.0987475E+04	2.30115669E+04	3.29566638E+04	1.11E+04
WOA	6.3990544E+03	1.05646549E+04	1.53847239E+04	2.35E+03
GOA	7.5304568E+03	7.72061080E+03	7.88181631E+03	1.77E+02
TEO	1.4514789E+04	1.56568300E+04	1.74616034E+04	1.58E+03
HHO	5.9567388E+03	6.50170020E+03	7.30424998E+03	3.34E+02
HBA	5.2766792E+03	5.87670812E+03	5.87720827E+03	1.001E−01

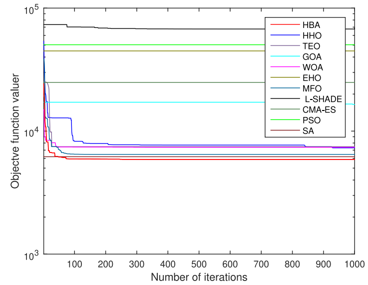
The significance of HBA performance against the counterparts on engineering design problems is validated using Friedman test (Table 21). According to the p-values, HBA did not produce significantly different solutions that other selected algorithms; however, small difference in solutions for real-life design problems have significant impact on production cost.

5. Analysis and discussion

The extensive experiments performed in previous section show the significance of the proposed HBA algorithm. The substantial test suite included not only standard benchmark functions but also complex optimization problems from CEC'17 — both suites solved with 30 and 50 dimensions. Moreover, the real-world engineering design



(a) Schematic design.



(b) Function values versus number of iterations.

Fig. 19. Speed reducer design problem.

Table 21
P-values at α = 0.05 by Friedman test for engineering problems.

vs. SA	vs. PSO	vs. CMA-ES	vs. L-SHADE	vs. MFO	vs. EHO	vs. WOA	vs. GOA	vs. TEO	vs. HHO
2.73E−01	1.00E−01	1.00E−01	1.00E−01	3.99E−01	1.00E−01	2.73E−01	2.73E−01	1.00E−01	2.73E−01

problems were also used, as these problems maintain highly non-convex optimization landscape. Overall, 52 optimization problems were employed to be solved by HBA and other commonly used metaheuristic approaches including SA, PSO, CMA-ES, L-SHADE, MFO, EHO, WOA, GOA, TEO, and HHO. The counterpart approaches selected for comparison purpose comprised of classical methods like SA, PSO, CMA-ES, and L-SHADE, as well as, established ones like MFO, EHO, WOA, GOA; and the most recent algorithms like TEO and HHO. This study carried rich analytical approach by not only observing end results in the form of mean and standard deviations found over certain number of runs, or convergence graphs, but also in-depth search behavioral evidence in the form exploration and exploitation measurements.

As per statistical results (Tables 6–8) of standard test suite, there is significant gap between the results of HBA and those of generated by SA, PSO, CMA-ES, L-SHADE, MFO, EHO, WOA, and GOA. However, TEO and HHO produced solutions relatively closer to HBA, but not much. This is further affirmed by t-test provided in Table 9. Moreover, HBA achieved better results in less time by efficiently converging to global optimum locations (Figs. 5–7). This suggests that HBA maintained superior explorative ability due to ample population diversity throughout search process (Figs. 8–11). Similar observations can be made on complex optimization problems of CEC'17 test-suite. The experimental results of scaled problem landscapes (Tables 10 and 11) also reveal the superiority and competitive solutions generated by HBA against the competitor methods. The convergence analysis in this connection also suggests that HBA is able to avoid immature convergence and stagnation issues by effective search strategies using robust randomization and exploration–exploitation switching strategy (Figs. 12 and 13). This brings stable trade-off balance between exploration and exploitation even when the search process is progressing towards the end. A vibrant population of candidate solutions in HBA ensures rigorous search of the landscape (Figs. 14 and 15). Furthermore, the importance of HBA is also highlighted by solving real-world design problems (Tables 14–20); showing flexibility of the proposed approach to wider range of optimization problems.

To better understand robustness of the proposed HBA algorithm, following features can be highlighted:

1. Exploitation behavior is effectively ensured with the help of intensity parameter (I) which guides population individuals to already identified promising regions.
2. The density factor (α) is used for dynamic time-varying randomization to ensure smooth transition from exploration to exploitation.
3. HBA uses flag F that does not allow population candidates trapping in local regions by directing search towards new regions for further improvement of solutions. This happens time by time throughout search process, hence HBA brings better opportunity of finding even better solutions even towards the end of iterations.
4. The digging and honey phases maintain constructive impact on solution update process by balancing exploration and exploitation properties of search mechanism.

6. Conclusion and future works

In this study, a novel nature-inspired population-based algorithm that mimics the foraging behavior of honey badger is proposed and termed as Honey Badger Algorithm (HBA). HBA aims to balance the exploration and exploitation abilities by efficiently traversing the search-space and avoiding suboptimal regions. The efficiency of the proposed HBA is evaluated through 24 standard benchmark functions, 29 functions of CEC'17 test-suite, and four engineering design problems. The search efficacy of the proposed approach is investigated in terms of the statistical results, exploration–exploitation ratios, and convergence curves. Moreover, in order to have a fair comparison, many other optimization techniques such Simulated annealing (SA), Particle Swarm Optimization (PSO), Covariance Matrix Adaptation Evolution Strategy (CMA-ES), Success-History based Adaptive Differential Evolution variants with linear population size reduction (L-SHADE), Moth-flame Optimization (MFO), Elephant Herding Optimization (EHO), Whale Optimization Algorithm (WOA), Grasshopper Optimization Algorithm (GOA), Thermal Exchange Optimization (TEO) and Harris hawks optimization (HHO) are also tested on the same experimental environment. The experimental result revealed that the HBA is effectively applicable to solve problems with complex search-space. The empirical study also affirmed the superiority in terms of convergence speed and trade-off exploration–exploitation balance. Future studies will be made to HBA, such as adding chaotic maps and binary and multi-objective capabilities, to solve other real-scale optimization problems.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] M.M. Ahmed, E.H. Houssein, A.E. Hassanien, A. Taha, E. Hassanien, Maximizing lifetime of large-scale wireless sensor networks using multi-objective whale optimization algorithm, Telecommun. Syst. 72 (2019) 243–259.
[2] A.V. Akopyan, Geometry of the cardioid, Amer. Math. Monthly 122 (2) (2015) 144–150.
[3] C. Begg, K. Begg, J. Du Toit, M. Mills, Scent-marking behaviour of the honey badger, mellivora capensis (mustelidae), in the southern kalahari, Anim. Behav. 66 (5) (2003) 917–929.
[4] C. Begg, K. Begg, J. Du Toit, M. Mills, Life-history variables of an atypical mustelid, the honey badger mellivora capensis, J. Zool. 265 (1) (2005) 17–22.
[5] E. Bonabeau, D.d.R.D.F. Marco, M. Dorigo, G. Théraulaz, G. Theraulaz, et al., Swarm Intelligence: from Natural to Artificial Systems, Vol. 1, Oxford University Press, 1999.
[6] I. BoussaïD, J. Lepagnot, P. Siarry, A survey on optimization metaheuristics, Inform. Sci. 237 (2013) 82–117.
[7] S. Cheng, Y. Shi, Q. Qin, Q. Zhang, R. Bai, Population diversity maintenance in brain storm optimization algorithm, J. Artif. Intell. Soft Comput. Res. 4 (2) (2014) 83–97.
[8] C.A.C. Coello, Use of a self-adaptive penalty approach for engineering optimization problems, Comput. Ind. 41 (2) (2000) 113–127.
[9] M. Črepinšek, S.-H. Liu, M. Mernik, Exploration and exploitation in evolutionary algorithms: A survey, ACM Comput. Surv. 45 (3) (2013) 1–33.
[10] G. Dhiman, V. Kumar, Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications, Adv. Eng. Softw. 114 (2017) 48–70.
[11] J.S. Gong Dunwei, X. Ji, Evolutionary algorithms with preference polyhedron for interval multi-objective optimization problems, Inform. Sci. 233 (2013) 141–161.
[12] N. Hansen, A. Ostermeier, Completely derandomized self-adaptation in evolution strategies, Evol. Comput. 9 (2) (2001) 159–195, http://dx.doi.org/10.1162/106365601750190398.
[13] F.A. Hashim, E.H. Houssein, K. Hussain, M.S. Mabrouk, W. Al-Atabany, A modified henry gas solubility optimization for solving motif discovery problem, Neural Comput. Appl. 32 (2019) 10759–10771.
[14] F.A. Hashim, E.H. Houssein, M.S. Mabrouk, W. Al-Atabany, S. Mirjalili, Henry gas solubility optimization: A novel physics-based algorithm, Future Gener. Comput. Syst. 101 (2019) 646–667.
[15] F.A. Hashim, K. Hussain, E.H. Houssein, M.S. Mabrouk, W. Al-Atabany, Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems, Appl. Intell. 51 (2020) 1531–1551.
[16] F. Hashim, M.S. Mabrouk, W. Al-Atabany, Gwomf: Grey wolf optimization for motif finding, in: 2017 13th International Computer Engineering Conference (ICENCO), IEEE, 2017, pp. 141–146.
[17] M.H. Hassan, E.H. Houssein, M.A. Mahdy, S. Kamel, An improved manta ray foraging optimizer for cost-effective emission dispatch problems, Eng. Appl. Artif. Intell. 100 (2021) 104155.
[18] A.E. Hassanien, M. Kilany, E.H. Houssein, H. AlQaheri, Intelligent human emotion recognition based on elephant herding optimization tuned support vector regression, Biomed. Signal Process. Control 45 (2018) 182–191.
[19] A. Heidari, M. Seyedali, F. Hossam, A. Ibrahim, M. Majdi, C. Huiling, Harris hawks optimization: Algorithm and applications, Future Gener. Comput. Syst. 97 (2019) 849–872.

[20] V. Heptner, Mammals of the Soviet Union: Vol. 2, Part 1b: Carnivora (Weasels, Additional Species), Smithsonian Institution Libraries & The National Science Foundation, 2001.
[21] J.H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, MIT Press, Cambridge, Mass, USA, 1975.
[22] E.H. Houssein, B.E.-d. Helmy, D. Oliva, A.A. Elngar, H. Shaban, A novel black widow optimization algorithm for multilevel thresholding image segmentation, Expert Syst. Appl. 167 (2021) 114159.
[23] E.H. Houssein, B.E.-d. Helmy, H. Rezk, A.M. Nassef, An enhanced archimedes optimization algorithm based on local escaping operator and orthogonal learning for PEM fuel cell parameter identification, Eng. Appl. Artif. Intell. 103 (2021) 104309.
[24] E.H. Houssein, M.A. Mahdy, M.J. Blondin, D. Shebl, W.M. Mohamed, Hybrid slime mould algorithm with adaptive guided differential evolution algorithm for combinatorial and global optimization problems, Expert Syst. Appl. 174 (2021) 114689.
[25] E.H. Houssein, M.A. Mahdy, A. Fathy, H. Rezk, A modified marine predator algorithm based on opposition based learning for tracking the global MPP of shaded PV system, Expert Syst. Appl. 183 (2021) 115253.
[26] E.H. Houssein, M.R. Saad, F.A. Hashim, H. Shaban, M. Hassaballah, Lévy Flight distribution: A new metaheuristic algorithm for solving engineering optimization problems, Eng. Appl. Artif. Intell. 94 (2020) 103731.
[27] K. Hussain, N. Neggaz, W. Zhu, E.H. Houssein, An efficient hybrid sine-cosine harris hawks optimization for low and high-dimensional feature selection, Expert Syst. Appl. 176 (2021) 114778.
[28] K. Hussain, M.N.M. Salleh, S. Cheng, Y. Shi, Metaheuristic research: a comprehensive survey, Artif. Intell. Rev. 52 (2018) 2191–2233.
[29] K. Hussain, M.N.M. Salleh, S. Cheng, Y. Shi, On the exploration and exploitation in popular swarm-based metaheuristic algorithms, Neural Comput. Appl. 31 (2018) 7665–7683.
[30] K. Hussain, M.N.M. Salleh, S. Cheng, Y. Shi, R. Naseem, Artificial bee colony algorithm: A component-wise analysis using diversity measurement, J. King Saud Univ.-Comput. Inf. Sci. 32 (7) (2018) 794–808.
[31] C. James, Introduction to Stochastics Search and Optimization, John Wiley and Sons, Hoboken, NJ, 2003.
[32] M. Jamil, X.-S. Yang, A literature survey of benchmark functions for global optimization problems, J. Math. Model. Numer. Optim. 4 (2) (2013) 150–194.
[33] B. Kannan, S.N. Kramer, An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design, J. Mech. Des. 116 (2) (1994) 405–411.
[34] H.M. Kanoosh, E.H. Houssein, M.M. Selim, Salp swarm algorithm for node localization in wireless sensor networks, J. Comput. Netw. Commun. 2019 (2019).
[35] D. Kapner, T. Cook, E. Adelberger, J. Gundlach, B.R. Heckel, C. Hoyle, H. Swanson, Tests of the gravitational inverse-square law below the dark-energy length scale, Phys. Rev. Lett. 98 (2) (2007) 021101.
[36] A. Kaveh, A. Dadras, A novel meta-heuristic optimization algorithm: thermal exchange optimization, Adv. Eng. Softw. 110 (2017) 69–84.
[37] A. Kaveh, A. Dadras, A novel meta-heuristic optimization algorithm: thermal exchange optimization, Adv. Eng. Softw. 110 (2017) 69–84.
[38] M. Kazemzadeh-Parsi, A modified firefly algorithm for engineering design optimization problems, Iran. J. Sci. Technol. Trans. Mech. Eng. 38 (M2) (2014) 403.
[39] Keith, C. Begg, The Honey Badger, 0000. http://www.honeybadger.com/index.html.
[40] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.
[41] E. Mezura-Montes, C.A.C. Coello, Useful infeasible solutions in engineering optimization with evolutionary algorithms, in: Mexican International Conference on Artificial Intelligence, Springer, 2005, pp. 652–662.
[42] S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, Knowl.-Based Syst. 89 (2015) 228–249.
[43] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.
[44] N. Neggaz, E.H. Houssein, K. Hussain, An efficient henry gas solubility optimization for feature selection, Expert Syst. Appl. 152 (2020) 113364.
[45] J.A. Parejo, A. Ruiz-Cortés, S. Lozano, P. Fernandez, Metaheuristic optimization frameworks: a survey and benchmarking, Soft Comput. 16 (3) (2012) 527–561.
[46] I. Rechenberg, Evolutionsstrategien, Springer Berlin Heidelberg, 1978, pp. 83–114.
[47] E. Russell, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, IEEE, 1995, pp. 39–43.
[48] A. Sadollah, A. Bahreininejad, H. Eskandar, M. Hamdi, Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems, Appl. Soft Comput. 13 (5) (2013) 2592–2612.
[49] S. Saremi, S. Mirjalili, A. Lewis, Grasshopper optimisation algorithm: theory and application, Adv. Eng. Softw. 105 (2017) 30–47.
[50] R. Tanabe, A.S. Fukunaga, Improving the search performance of SHADE using linear population size reduction, in: 2014 IEEE Congress on Evolutionary Computation (CEC), 2014, pp. 1658–1665, http://dx.doi.org/10.1109/CEC.2014.6900380.
[51] G.-G. Wang, S. Deb, X.-Z. Gao, L.D.S. Coelho, A new metaheuristic optimisation algorithm motivated by elephant herding behaviour, Int. J. Bio-Inspired Comput. 8 (6) (2016) 394–409.
[52] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82.
[53] G. Wu, R. Mallipeddi, P. Suganthan, Problem Definitions and Evaluation Criteria for the CEC 2017 Competition on Constrained Real-Parameter Optimization, Technical Report, National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, 2017, URL http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC2017.
[54] M. Yazdani, F. Jolai, Lion optimization algorithm (LOA): a nature-inspired metaheuristic algorithm, J. Comput. Des. Eng. 3 (1) (2016) 24–36.