Appl Intell (2017) 47:850–887
DOI 10.1007/s10489-017-0903-6

Human mental search: a new population-based metaheuristic optimization algorithm

Seyed Jalaleddin Mousavirad[1] · Hossein Ebrahimpour-Komleh[1]

Published online: 28 April 2017

Abstract Population-based metaheuristic algorithms have become popular in recent years with them getting used in different fields such as business, medicine, and agriculture. The present paper proposes a simple but efficient population-based metaheuristic algorithm called Human Mental Search (HMS). HMS algorithm mimics the exploration strategies of the bid space in online auctions. The three leading steps of HMS algorithm are: (1) the mental search that explores the region around each solution based on Levy flight, (2) grouping that determines a promising region, and (3) moving the solutions toward the best strategy. To evaluate the efficiency of HMS algorithm, some test functions with different characteristics are studied. The results are compared with nine state-of-the-art metaheuristic algorithms. Moreover, some nonparametric statistical methods, including Wilcoxon signed rank test and Friedman test, are provided. The experimental results demonstrate that the HMS algorithm can present competitive results compared to other algorithms.

Keywords Population-based metaheuristic · Levy flight · Optimization · Nonparametric statistical analysis · Human mental search · Stochastic optimization

✉ Seyed Jalaleddin Mousavirad
Jalalmoosavirad@gmail.com

✉ Hossein Ebrahimpour-Komleh
Ebrahimpour@kashanu.ac.ir

1 Department of Computer Engineering, Faculty of Computer and Electrical Engineering, University of Kashan, Kashan, Iran

1 Introduction

Optimization is the process of searching the optimal values for a particular problem. Optimization problems can be consulted in a variety of scientific fields such as economy, engineering, and medicine. Therefore, the development of optimization algorithms is necessary and many researchers all over the world are working in this field.

One of the main weaknesses of classic optimization algorithms is local optima stagnation, whereby they lack sufficient ability to find global optima. Some of them need derivation of search space as well. Therefore, these kinds of algorithms are not highly efficient in solving real-world problems.

In comparison to classic optimization algorithms, metaheuristic algorithms are problem-independent with stochastic operators for solving optimization problems. Randomness is one of the main characteristics of these algorithms. Metaheuristic algorithms are becoming increasingly popular because 1) they are more robust in avoiding local optima than classical optimization algorithms, and 2) they do not require the gradient of the cost function.

Metaheuristic algorithms are divided into two classes: single-based and population-based. Single-based metaheuristic algorithms start with a single solution and try to improve it over some iteration processes. Tabu Search [1], Simulated Annealing [2, 3], Variable Neighbourhood Search [4], Hill Climbing [5], and Iterated Local Search [6] are some of the most famous algorithms in this class. Unlike single-based metaheuristics, population-based metaheuristics start with a set of solutions (population). They then iteratively create a new population of solutions. In this way, information can be exchanged among the set of solutions. The main advantage of population-based metaheuristics is

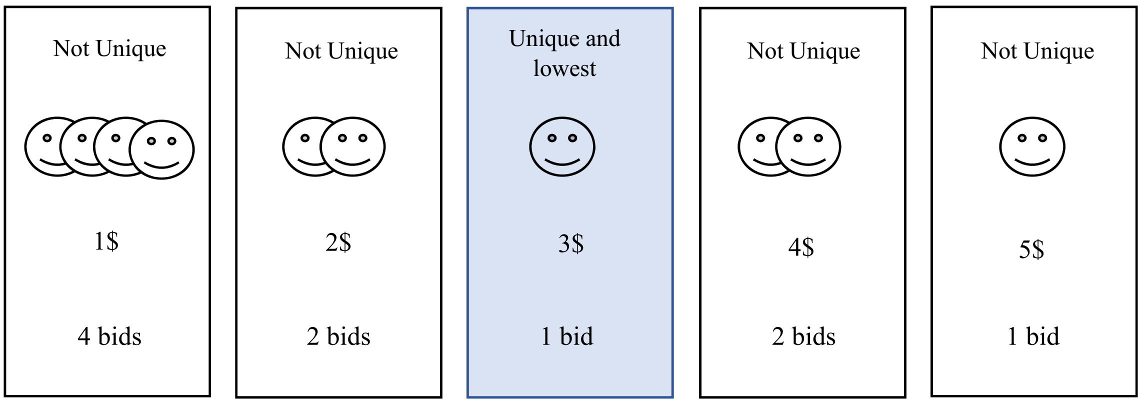


Fig. 1 Lowest unique bid auction

that they avoid getting stuck in the local optima. This type of metaheuristic algorithms is one of the most well-known optimization algorithms that has been widely applied in various applications such as medical systems [7, 8], car engine design [9], fault diagnosis [10], and food quality [11].

From another perspective, metaheuristics are divided into three categories: evolutionary, swarm-based, and physics-based algorithms. Evolutionary algorithms are inspired by evolutionary behaviours in nature. Genetic algorithm, (GA) which was proposed by Holland [12], is the most well-known evolutionary algorithm. The general idea of this algorithm is based on Darwin's theory of evolution. GA starts with random candidate solutions. Then, the recombination and mutation operators are applied to generate new solutions. Finally, a selection approach is used to select solutions for the next generation. Some of the other evolutionary-based metaheuristics are Evolution Strategy (ES) [13], Genetic Programming (GP) [14], Differential Evolution (DE) [15], Probability-based Incremental Learning (PBIL) [16], Evolutionary Programming (EP) [17] and Biogeographybased Optimization (BBO) [18]

The next category of metaheuristic algorithms, i.e. swarm-based algorithms (SA) is inspired by the social behaviour of animals in nature. Some of the popular SAs are Particle Swarm Optimization (PSO) [19, 20] inspired by the social and individual behaviour of birds, Artificial Bee Colony (ABC) [21] inspired by the food searching behaviour of bee swarm, Cuckoo Search (CS) [22] that mimics the unusual behaviour in the laying of eggs, Firefly Algorithm (FA) [23] inspired by the flashing characteristics of fireflies, Shuffled Frog Leaping Algorithm (SFLA) [24] that gets the idea from the social behaviour of frogs, Grey Wolf Optimizer (GWO) [25] that simulates the hunting behaviour of grey wolves, and Whale Optimization Algorithm(WOA) [26] inspired by the social behaviour of humpback whales.

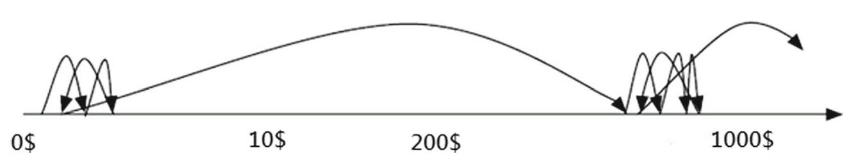
The third category of metaheuristic algorithms is physics-based algorithms which mimic physical rules in nature. Some of the most popular algorithms in this class are Simulated Annealing (SA) [2, 27], Gravitational Search Algorithm (GSA) [28], Water Cycle Algorithm (WCA) [29], and Mine Blast Algorithm (MBA) [30].

Human-based metaheuristic (HM)algorithms are introduced as a new category in some papers [26, 31]. These algorithms imitate human behaviours and characteristics. Harmony Search (HS) [32] and Imperialist Competitive Algorithm (ICA) [33] are two examples of human-based metaheuristics.

The two common characteristics among population-based metaheuristic algorithms are intensification (exploitation) and diversification (exploration). Intensification tries to find better solutions by searching around the best solutions. In contrast, diversification refers to the algorithm's ability to explore the promising area of search space. These two criteria are usually in conflict with each other, and finding a proper trade-off between intensification and diversification is one of the most important challenges in the development of metaheuristic algorithms.

According to No Free Lunch (NFL) theorem [34], there is no metaheuristic algorithm to solve all optimization problems optimally. In other words, a metaheuristic algorithm can be highly efficient for some problems, while it may be

Fig. 2 A typical example of consecutive bid values by a person



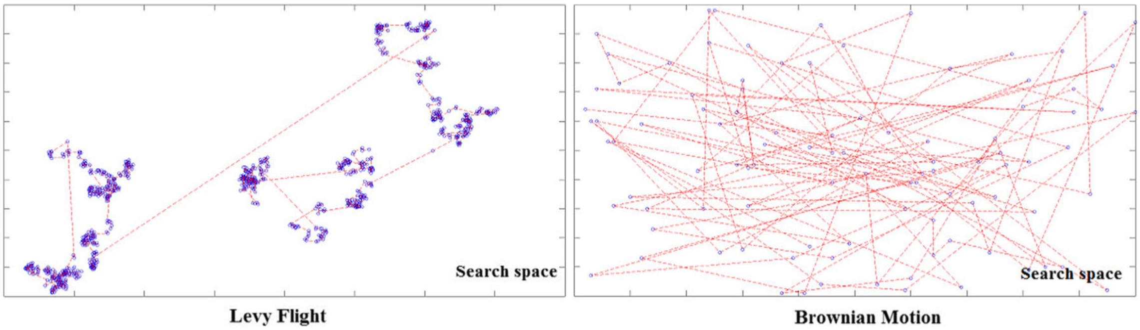


Fig. 3 An example of Levy flight against Brownian motion

poorly efficient for some others. Hence, the development of this research area is an open problem, and many researchers try to propose new metaheuristic algorithms or improve one of them.

The present study proposes a novel population-based metaheuristic algorithm called Human Mental Search (HMS). The HMS algorithm is inspired by the exploration strategies of the bid space in online auctions. The HMS algorithm has three leading operators: mental search, grouping, and moving. The mental search creates some new solutions around a solution based on Levy flight that leads to enhanced diversification and intensification properties, simultaneously. Another operator is grouping, whereby the solutions are grouped into some regions using a clustering algorithm. Finally, the moving operator tries other solutions to get close to the promising region. Preliminary studies indicate that HMS could outperform existing algorithms such as PSO, HS, SFLA, ABC, ICA, BBO, FA, GWO, and WOA. The remainder of this paper is organized as follows:

In Section 2, the proposed Human Mental Search (HMS) algorithm is explained. The statistical results for standard benchmarks are discussed in Section 3. Finally, Section 4 presents the conclusions of the present study and some recommendations for future researches.

2 Human Mental Search (HMS)

The current study proposes a new population-based metaheuristic algorithm based on the exploration strategies of the bid space in online auctions called Human Mental Search (HMS). This section first explains the source of inspiration and, then presents HMS algorithm

2.1 The source of inspiration

Recently, Radicchi et al [35, 36] demonstrated that humans apply the Levy flight strategy to explore the bids space in online auctions. The exploration of bid space is a search process, but of the mental kind because it works in an abstract space. To this end, they are considered participants in a new generation of online auction called Lowest Unique Bid (LUB). The auction winner might be able to buy an expensive product at the lowest price; cars, electronic devices, and even houses can be purchased with just a few hundred dollars.

The period of an auction is announced in advance. A bid could be of any value from a minimum value L to a maximum value H. Each time a participant makes a bid, he/she has to pay a fee. Every participant has permission to go for

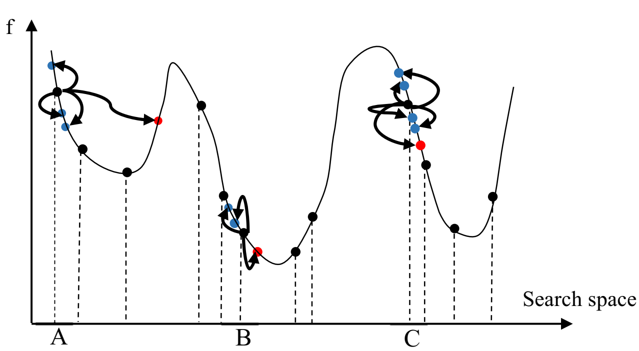


Fig. 4 Mental search

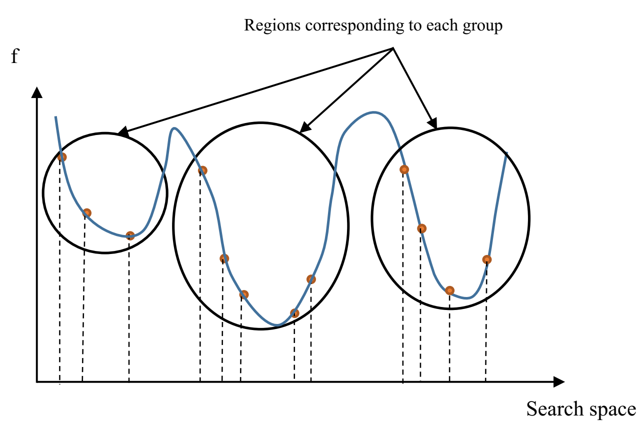


Fig. 5 Grouping operator for a problem with one dimension

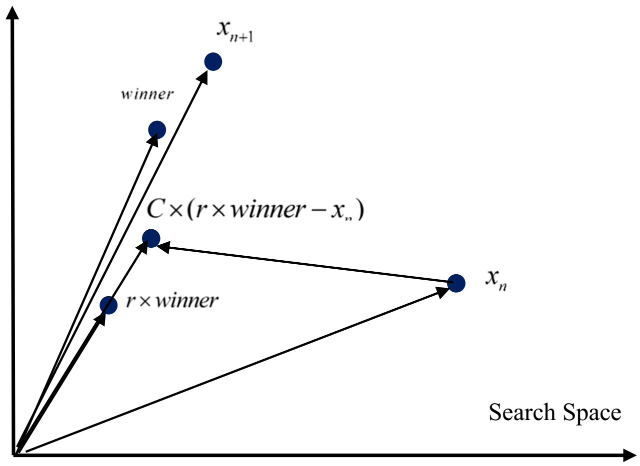


Fig. 6 Position updating in the HMS algorithm

multiple bids. The winner of the auction is a person who has placed the lowest unique number and can buy the product for the value of the winning bid. For example, in Fig. 1, the winner is the participant who made a bid of $3 because this bid shows the lowest unique bid. Other bids are not unique except for $5, which is not the lowest one.

The Highest Unique Bid (HUB) auction has the same mechanism except that the winner is the participant who has the highest unique bid. In Fig. 1, the participant with $5 is the winner in the HUB auction. Participants in LUB and HUB auctions attempt to find a single target whose position is determined by the bids.

Radicchi et al. [35, 36] showed that the bid space exploration performed by the participants has an explosive manner, which means that the consecutive bid values are close together but sometimes, the participants do longer jumps. Figure 2 shows a typical example of consecutive bid values by a person. In other words, the exploration of the bid space is consistent with Levy flight. At the end of each auction, the losing participants tend to pick the winner strategy, and so they get close to the winner's strategy for the next auction.

2.2 HMS algorithm

This subsection explains HMS algorithm. The following concepts are used to develop this algorithm:

1. Each participant has a strategy α,
2. Each person can provide a bid,
3. The next bid of every person is consistent with the Levy flight distribution,
4. Multiple bids are allowed,
5. The losing participants try to pick the winner's strategy for the subsequent auctions.

The HMS algorithm is explained in detail below.

2.2.1 Generating initial bids

The HMS algorithm is a population-based metaheuristic algorithm. Like other population-based metaheuristic algorithms, the searching process starts with the generation of a random population of candidate solutions. In this algorithm, each single solution is called a bid. In an N_{Var-}dimensional optimization problem, a bid is represented as follows:

$$bid = [x_1, x_2, \ldots, x_{N_{Var}}] \tag{1}$$

Cost value of a bid is obtained by evaluating the cost function, as:

$$Cost\ Value\ of\ \text{a}\ bid = f(bid) = f(x_1, x_2, \ldots, x_{N_{Var}}) \tag{2}$$

First, a bids matrix of size $N_{pop} \times N_{Var}$ is generated as follows:

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_{pop}^{N} \end{bmatrix} = \begin{bmatrix} x_1^1 & x_2^1 & \ldots & x_{N_{Var}}^1 \\ x_1^2 & x_2^2 & \ldots & x_{N_{Var}}^2 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{N_{popo}} & x_2^{N_{popo}} & \ldots & x_{V_{Var}}^{N_{pop}} \end{bmatrix} \tag{3}$$

where N_{pop} is the number of bids, N_{Var} is the number of variables, and X is the bids matrix.

2.2.2 Mental search

The mental search represents the number of consecutive values produced for each bid. In this stage, some new bids are created around a bid based on Levy flight. The number of other new bids for each bid is a random integer number between the upper and lower limits. Levy flight is a particular type of random walk determining step size with a Levy distribution. Random walk is a Markov chain in which the next position depends only on the current position. Figure 3 shows an example of Levy flight against a Brownian motion. As shown in the figure, there are a lot of small steps and sometimes long jumps in Levy flight. In other words, Levy flight increases the quality of diversification and intensification simultaneously. It is a valuable point that Levy flight is more efficient than Brownian motion to explore the unknown spaces.

The following equation shows the Levy distribution:

$$L(x) = \frac{1}{\pi} \int_0^{\infty} \exp(-\alpha q^{\beta}) \cos(qx) dx \tag{4}$$

where β is called distribution index which is limited to $0 < \beta \leq 2$ and α is the distribution scale factor.

To generate each new position in the mental search, Levy flight is applied based on (5):

$$NS = X^i + S \tag{5}$$

```
//Settings
L (lower bound), U (upper bound), M_L (minimum mental process), M_h (maximum mental process), N_pop (the number of bids), N_Var (the
number of variables), K (the number of clusters), iter (current iteration), MaxIter (maximum iteration)
//Initialization
begin
  X=Initialize a population of N_pop bids
  Calculate the cost of bids
  x*=Find the best bid in the initial population
  for i from 1 to N_pop do
      β_i =generate an integer random number between a lower and upper bound
  end-for
    for iter from 1 to MaxIter do
      //Mental Search
      for i from 1 to N_pop do
          q_i=generate an integer random number between M_L and M_h
      end-for
      for i from 1 to N_pop do
          for j from 1 to q_i do
              s = (2 − Iter * (2 / MaxIter)) * 0.01 * u / v^(1/β_i) * (x^i − x*)
              NS_j=X^i+s;
          end-for
          t=find NS with the lowest cost
          if cost(t)<cost(X^i)
              X^i =t
          end-if
    end-for
    //Clustering
      cluster N_pop bids into K clusters
      calculate the mean cost value of each cluster
      select cluster with the lowest mean cost value as the winner cluster
      winner=select the best bid in the winner cluster
    //Moving Bids toward the best strategy
    for i from 1 to N_pop do
      for n from 1 to N_Var do
          X^i_n = X^i_n + C * (r * winner_n − X^i_n)
      end-for
    end-for
    for i from 1 to N_pop do
          β_i =generate a random number between a lower and upper bound
    end-for
    x+ =Find the best bid in the current bids
    if cost(x+) <cost (x*) do
        x* =x+
    end-if
  end-for
end-begin
```

Fig. 7 The pseudo code for the HMS algorithm

And S is calculated as below:

$$S = (2 - iter^{*}(2\ \max\ iter))^{*}\alpha \oplus Levy \tag{6}$$

where *max iter* is the maximum iteration, *iter* is the current iteration, and α is a random number. The product $\oplus$ means entry-wise multiplications. Component of $(2 - iter^{*}(2/\max iter))$ is a reduction factor, and it is actually reduced from 2 to 0. This factor lays emphasis on the diversification and intensification. The bigger reduction factor shows the long jumps and it increases the process of diversification at the beginning of the algorithm, while the smaller reduction factor indicates the smaller jumps and it enhances the process of intensification in the later stages.

The generation of step size S in not trivial while using Levy flight. A simple method discussed in detail by Yang [37, 38] can be summarized as follows:

$$\begin{aligned} S &= (2 - iter^{*}(2/\max iter))^{*}\alpha \oplus Levy \\ &= (2 - iter^{*}(2/\max iter))^{*}0.01^{*}\frac{u}{v^{1/\beta}}{}^{*}(\mathrm{x}^{i} - x^{*}) \end{aligned} \tag{7}$$

where x^{*} is the best position obtained so far, and u and v are the random numbers from the normal distribution as below:

$$u : N(0, \sigma_u^2),\ v : N(\sigma_v^2) \tag{8}$$

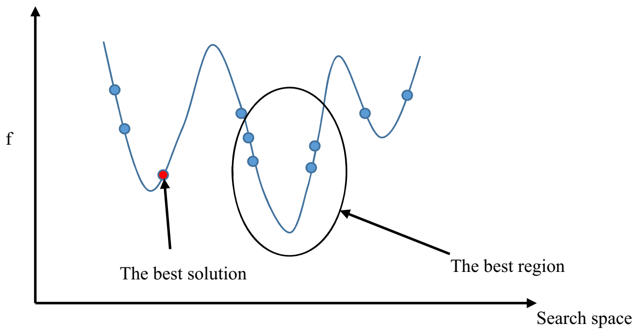


Fig. 8 The best solution may not be a good representative for the promising region

with

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta)\sin(\frac{\pi\beta}{2})}{\Gamma[(\frac{1+\beta}{2})]\beta 2^{(\beta-1)/2}} \right\}^{1/\beta}, \sigma_v = 1 \tag{9}$$

where Γ is the standard Gamma function.

One of the main parameters in Levy flight is β. This parameter is different for each bid because each person has a different strategy. For each bid, a random number is assigned to the parameter β between 0 and 2. The lower β shows the bigger jumps, and increases the ability to explore unknown area (or diversification). The higher β indicates the smaller jumps, and increases the intensification process.

Figure 4 illustrates mental search process for three specified bids A, B, and C in a one dimensional problem. The number of other new bids for bids A, B, and C are 4, 3, and 5 respectively. As can be observed, each bid produces other new bids (red and blue points in Fig. 4) with random positions around a bid that increases the intensification property. Moreover, sometimes there are long jumps that help the diversification property. Finally, each bid will be replaced with the best bid generated by using the mental search operator (red points in Fig. 4). This process must be conducted for all the bids.

2.2.3 Grouping the bids

Every person may make multiple bids. To simulate the multiple bids, a grouping procedure is proposed. Each group shows the bids belonging to a person. The process of grouping is performed by a clustering algorithm. Clustering is a pattern recognition technique for grouping a set of instances, whereby the instances in the same group are more similar to each other than to those in the other groups. In this paper, well-known clustering algorithm K-means algorithm [39] is chosen for this purpose. After grouping, the

Table 1 Default parameter settings

Algorithms	Parameters	Value
PSO	Cognitive constant(C_1) [19]	2
	Social constant(C_2) [19]	2
	Inertia constant (w) [19]	1 to 0
HS	Harmony memory considering rate [32]	0.9
	pitch adjusting rate [32]	0.1
SFLA	Number of memeplexes [24]	100
	Number of frogs [24]	30
ABC	limit [43]	n_e × dimension of problem
ICA	Number of empires [33]	5
	Coefficient associated with average power [33]	0.1
	Revolution rate [33]	0.2
	Deviation assimilation parameter [33]	$\pi/4$
	Direction assimilation parameter [33]	0.5
BBO	Habitat modification probability [18]	1
	Maximum immigration rate [18]	1
	Maximum emigration rate [18]	1
FA	light absorption coefficient(γ) [23]	1
	Attractiveness at r = 0 (β_0) [23]	1
	Scaling factor(α) [23]	0.2
GWO	No parameter	–
WOA	A constant for defining the shape of the logarithmic spiral(b) [26]	1

Table 2 Unimodal test functions

Function	D	Range	f_{min}
$F_1 = \sum_{i=1}^{n} x_i^2$	30	[−100,100]	0
$F_2 = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	[−10,10]	0
$F_3 = \sum_{i=1}^{n} (\sum_{j-1}^{i} x_j)^2$	30	[−100,100]	0
$F_4(x) = \max_i\{\lvert x_i \rvert, 1 \le i \le n\}$	30	[−100,100]	0
$F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[−30,30]	0
$F_6 = \sum_{i=1}^{n} ([x_i + 0.5])^2$	30	[−100,−100]	0
$F_7 = \sum_{i=1}^{n} i x_i^4 + random[0, 1]$	30	[−1.28,−1.28]	0

mean cost value of each group is calculated. It can be said that as the number of local optima goes up, a greater number of clusters is required. However, the number of local optima is unknown in advance.

In other words, the search space is divided into some regions with the promising region chosen by the mean cost value. Figure 5 illustrates the grouping operator for a problem with one dimension ($N_{var} = 1$). In this figure,

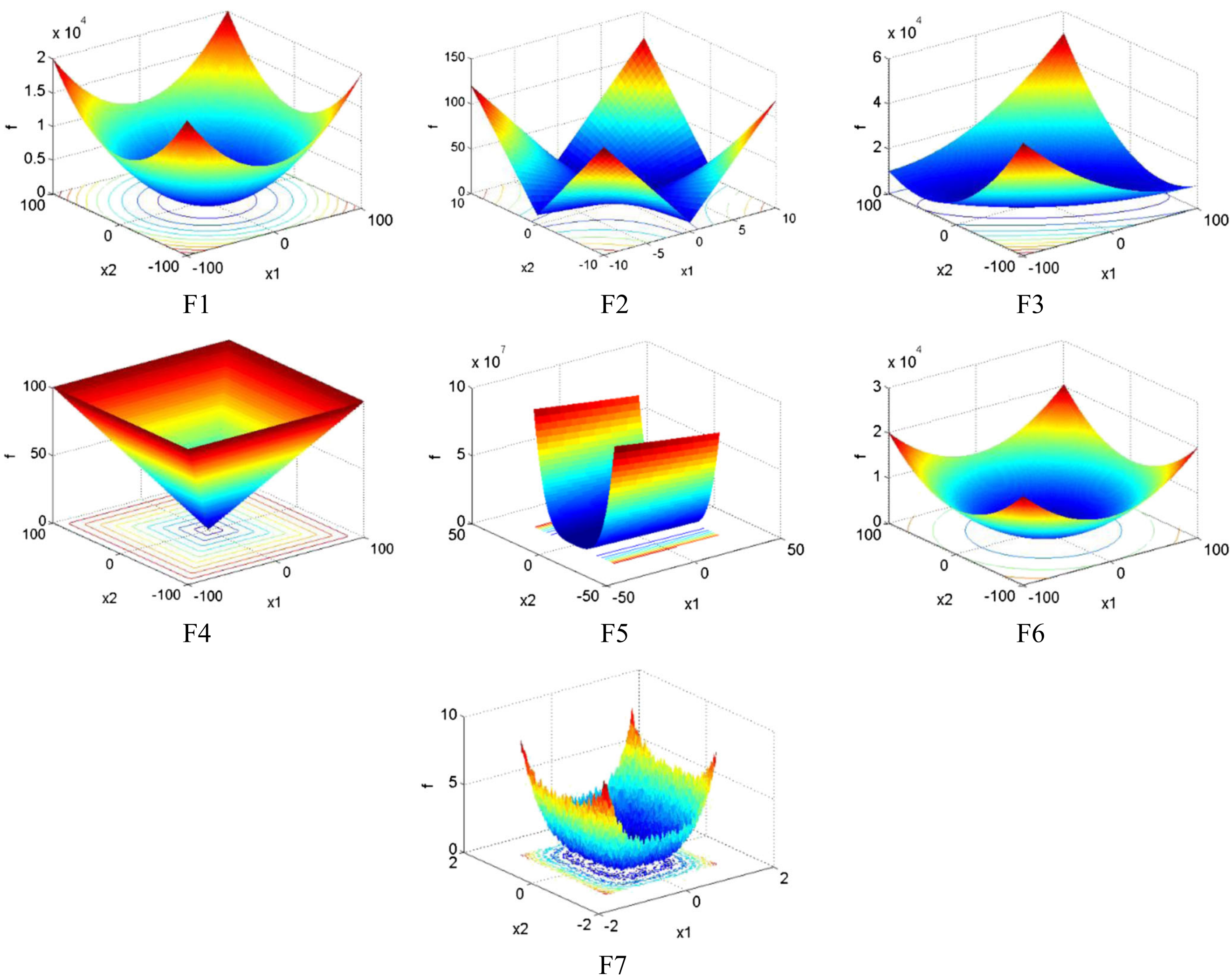


Fig. 9 Search space of unimodal test functions

Table 3 The statistical results of unimodal test functions

F	Statistics	HMS	PSO	HS	SFLA	ABC
F1	AVE	0(1)	321.8775(9)	291.9645(8)	6.7096E-19(4)	2.7422E04(10)
	STD	0	96.3709	82.9488	1.9050E-18	3.6898E03
	Min	0	97.7029	169.0476	3.2883E-21	2.2247E04
	Max	0	588.548	552.6958	9.5526E-18	3.3785E04
F2	AVE	0(1)	7.8858(9)	3.0447(8)	2.3890E-11(4)	84.2773(10)
	STD	0	3.2087	0.6654	2.1578E-11	6.7554
	Min	0	3.7057	1.9862	2.1201E-12	68.9766
	Max	0	18.9805	4.6476	7.8206E-11	101.9020
F3	AVE	0(1)	2901.9230(7)	21.3541(5)	1.2336(3)	2.7631E04(9)
	STD	0	1080.425	2.3304	1.4039	5.2132E03
	Min	0	1487.715	17.7543	0.1641	1.4971E04
	Max	0	5017.521	26.6807	6.8424	3.9297E04
F4	AVE	0(1)	14.0870(7)	21.7955(8)	0.0291(3)	60.5905(10)
	STD	0	3.2558	2.3199	0.0214	3.3712
	Min	0	9.2186	17.8125	0.0048	53.6154
	Max	0	19.5784	27.4162	0.0986	67.3042
F5	AVE	28.2242(5)	1.9504E04(8)	2.4305E04(9)	25.8845(2)	3.3098E07(10)
	STD	0.2476	2.4883E04	1.2178E04	18.5675	9.0472E06
	Min	27.5407	1.8487E03	7.1332E03	0.2431	1.5918E07
	Max	28.9079	1.3664E05	5.7314E04	84.0835	3.4229E07
F6	AVE	1.4690(4)	368.2913(9)	253.8506(8)	7.6109E-19(1)	2.6756E04(10)
	STD	1.1588	140.6384	87.7723	1.4446E-18	4.8617E03
	Min	0.0531	170.2104	99.9663	2.0549E-21	1.5266E04
	Max	4.1604	790.5266	568.3756	6.5459E-18	3.6423E04
F7	AVE	1.6218E-05(1)	0.1284(7)	0.2177(8)	0.0097(4)	19.3983(10)
	STD	1.7430E-05	0.0577	0.0567	0.0054	4.4983
	Min	1.0301E-07	0.0490	0.1346	0.0036	9.9985
	Max	7.2422E-05	0.3056	0.3666	0.0321	28.2581
Average rank		2	8	7.71	3	9.86
Overall rank		1	9	8	3	10

F	Statistics	ICA	BBO	FA	GWO	WOA
F1	AVE	1.9004E-06(5)	2.4929(7)	0.0055(6)	8.6478E-28(3)	1.3396E-73(2)
	STD	2.2412E-06	0.5285	0.0030	1.0918E-27	4.6770E-73
	Min	1.4215E-07	1.4883	0.0010	1.1649E-29	1.0164E-86
	Max	1.0793E-05	3.4210	0.0122	4.8614E-27	2.2296E-72
F2	AVE	1.1933E-04(5)	0.4938(6)	0.5284(7)	1.2648E-16(3)	4.3201E-50(2)
	STD	8.1609E-05	0.0672	0.2599	1.6947E-16	2.2718E-49
	Min	2.4031E-05	0.3665	0.1598	1.1922E-17	3.7886E-56
	Max	3.5749E-04	0.6424	1.3720	9.4598E-16	1.2459E-48
F3	AVE	7.7461(4)	486.7795(6)	8195.2356(8)	1.4881E-04(2)	4.3242E04(10)
	STD	2.4242	194.8450	5180.6050	7.4397E-04	1.2312E04
	Min	2.9868	239.9987	1100.9944	2.0834E-08	2.0771E04
	Max	12.9662	1.1041E03	21517.5125	0.0041	6.9017E04
F4	AVE	8.4321(6)	1.5562(4)	7.1223(5)	7.6278E-07(2)	43.9332(9)
	STD	3.2616	0.1586	2.5367	5.0380E-08	29.5684
	Min	2.9868	1.1853	1.3606	9.2926E-08	0.0945
	Max	15.9389	1.9220	14.3639	1.8519E-06	92.0258
F5	AVE	5.8725(1)	245.1556(7)	84.0091(6)	27.1047(3)	27.9634(4)
	STD	2.4985	252.8022	32.6139	0.7762	0.4133

Table 3 (continued)

	Min	1.9919	62.9754	28.7240	26.0949	26.9892
	Max	12.9347	1.2422E03	149.5544	28.7588	28.7592
F6	AVE	7.8532(6)	2.4665(5)	8.0577(7)	0.7865(3)	0.4143(2)
	STD	2.6143	0.5996	0.1428	0.3173	0.1744
	Min	1.3092	1.6068	7.8064	1.7492E-04	0.0719
	Max	12.9394	3.9749	8.3306	1.2616	0.7362
F7	AVE	1.1611(9)	0.0141(5)	0.1232(6)	0.0020(2)	0.0033(3)
	STD	1.1418	0.0045	0.0461	9.7713E-04	0.0034
	Min	2.9789E-07	0.0093	0.0411	4.9244E-04	1.0718E-04
	Max	3.9798	0.0269	0.2365	0.0051	0.0144
Average rank		5.14	5.71	6.43	2.57	4.57
Overall rank		5	6	7	2	4

there are 12 candidate solutions, which are divided into three groups.

2.2.4 Moving bids toward the best strategy

As mentioned earlier, the losers try to get close to the winner's strategy. After the bid groups are created, the bid group with the best mean cost value is selected as the winner group for other bids that determine a promising region. Then, the best bid in the winner cluster is selected in order to move the rest of the bids toward it. It is worth mentioning that the best cost value among all the bids might not belong to the winner group.

The following formula is proposed in this regard:

$$^{t+1}x_x^i = ^t x_x^i + C^*(r \times ^t winner_n - ^t x_n^t) \tag{10}$$

where $^{t+1}x_n^i$ indicates the nth element of X^i at iteration of $t+1$,$^t winner_n$ is the nth element of the best bid in the winner cluster t is the current iteration, C is a constant number (In this paper, $C = 2$), and r is a random number drawn from the uniform distribution between 0 and 1. Figure 6 shows how a bid updates its position using the moving operator.

2.2.5 General structure of HMS algorithm

In this section, the HMS algorithm for solving optimization problems is explained. The pseudo code for the HMS algorithm is presented in Fig. 7. Similar to other population-based metaheuristic algorithms, the HMS algorithm starts with a set of random bids(X). Then, the cost of each bid is calculated. For each bid, an integer number (q) is generated that shows the number of mental searches for each bid. In this step, the mental search operator is applied, which generates some new bids (*NS*) around each bid X^i using Levy flight. Then, the best solution generated in the previous step is replaced by X^i if its cost value is better than X^i. Later, the search space is divided into some groups by using a clustering operator. The winner group is the group with the best mean cost value that determines a promising region. In the next step, the other bids move toward the best bid in the winner cluster.At each stage, the best bid is saved. This process will continue until a stop condition is satisfied.

To see how the HMS algorithm can be effective in solving optimization problems, some points are noted below:

- Mental search allows obtaining neighbouring solutions around a solution. Therefore, it enhances the quality of intensification simultaneously. In addition, this operator increases their diversification property because sometimes there are the long jumps.
- Reduction factor allows the HMS algorithm to move smoothly from diversification to intensification.
- The grouping operator quickly finds the promising regions of the search space. It clearly differs from other population-based metaheuristic algorithms which, try to find the promising region by using the best solution. The best solution may not be a good representative for the promising region.
- The best solution in the winner cluster guides other solutions toward the promising regions of the search space.
- There is a high probability of solving local optima stagnation because of Levy flight.
- HMS algorithm is a population-based metaheuristic algorithm. Therefore, it intrinsically takes the advantages of high diversification and the local optima avoidance as compared to the single-based metaheuristic algorithms.
- The best solution of each iteration is saved (elite).
- HMS algorithm has very few parameters to be adjusted.

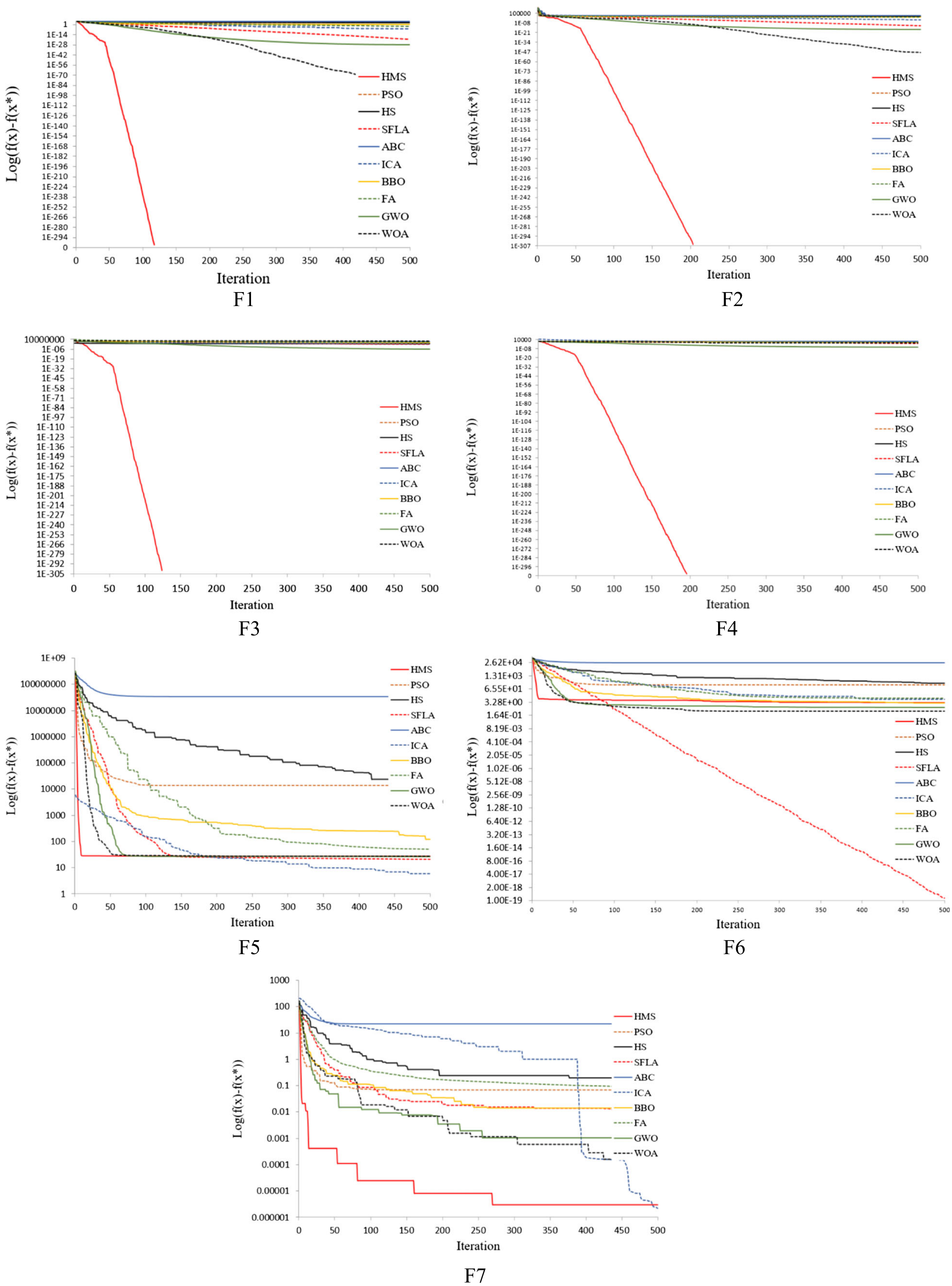


Fig. 10 Convergence curves on unimodal test functions

Table 4 Multimodal test functions

Function	D	Range	f_{min}
$F_8(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert})$	30	[−500,500]	$-481.9829 \times Dim$
$F_9(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	[−5.12,5.12]	0
$F_{10}(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) + 20 + e$	30	[−32,32]	0
$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1$	30	[−600,600]	0
$F_{12}(x) = \frac{\pi}{n}\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30	[−50,50]	0
$F_{13}(x) = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	[−50,50]	0

- HMS algorithm is a gradient-free algorithm and considers problems as a black-box. Hence, various problems in different fields can be solved by using the HMS algorithm.

2.2.6 Differences between HMS and other population-based metaheuristic algorithms

This section points out some distinctions of the HMS algorithm vis-à-vis other population-based metaheuristic algorithms. Some of the most important distinctions are listed below.

- One of the major differences, is that the HMS algorithm uses a clustering algorithm to determine the promising region. Most of metaheuristic algorithms such as PSO, ICA, and DE find the promising regions by using the best solutions. However, the best solution may not be a good representative for the promising region. Figure 8 shows that the best solution (red circle) sometimes does not show the best region. In the HMS algorithm, we used a clustering procedure to find the promising region. As a result, finding the promising region is based on several similar (close) solutions, thereby increasing the probability of finding a promising region.

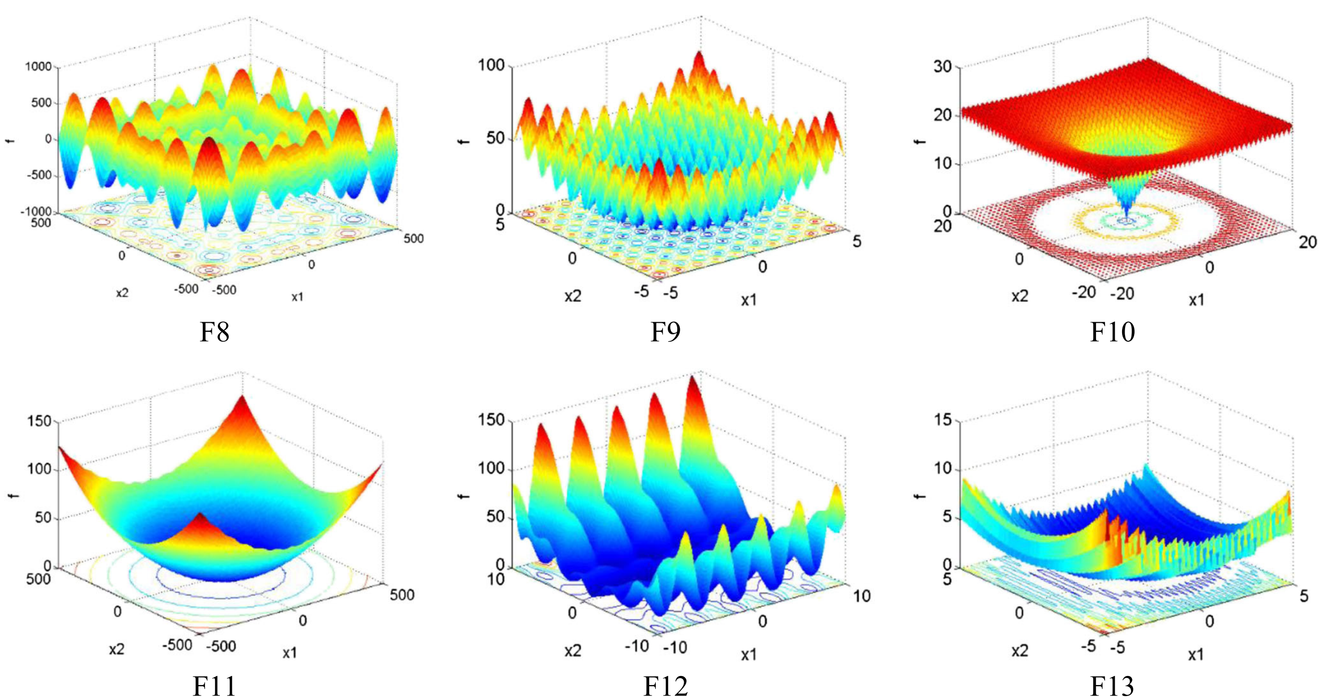


Fig. 11 Search space in multimodal test functions

Table 5 The statistical results of multimodal test functions

F	Statistics	HMS	PSO	HS	SFLA	ABC
F_8	AVE	−9308.3218(2)	−5839.7546(10)	−12405.1614(1)	−7119.1248(8)	−9113.9842(3)
	STD	600.9386	802.4674	64.5581	1475.2159	459.6007
	Min	−9911.0143	−8892.3159	−12504.3145	−8818.6241	−1055.5181
	Max	−6563.8127	−4542.8145	−12185.4515	−4107.6214	−8403.6147
F_9	AVE	0(1.5)	87.9261(9)	19.3135(5)	68.7188(7)	151.9296(10)
	STD	0	16.7137	3.1289	19.7114	13.0229
	Min	0	49.7366	13.1085	29.8488	124.3694
	Max	0	112.3647	26.1129	111.4349	172.1271
F_{10}	AVE	8.8816E-16(1)	6.6234(9)	4.9583(7)	0.3039(5)	18.4379(10)
	STD	0	0.9991	0.5705	0.6440	0.2734
	Min	8.8816E-16	4.0641	3.9277	2.5218E-11	17.8017
	Max	8.8816E-16	8.6646	6.4929	2.3168	18.9170
F_{11}	AVE	0(1)	4.1369(8)	3.5463(7)	0.0102(4)	240.8548(10)
	STD	0	1.2931	0.8018	0.0140	39.8329
	Min	0	2.1139	2.2758	0	142.4818
	Max	0	6.6833	5.2237	0.0613	304.8039
F_{12}	AVE	0.0178(2)	9.1955(9)	6.4039(7)	0.3153(5)	184237.4110(10)
	STD	0.0281	4.3929	1.6343	0.3836	104481.1280
	Min	3.5323E-04	2.3361	3.5329	1.9521E-20	55290.2189
	Max	0.1278	21.5076	10.1699	1.3601	420019.5411
F_{13}	AVE	0.2194(4)	143.5001(8)	220.7775(9)	0.0095(2)	920691.2156(10)
	STD	0.2599	341.9978	250.9874	0.0201	31057.1275
	Min	0.0114	25.5510	29.224	7.7792E-21	30391.9127
	Max	1.2401	1891.8135	999.1900	0.0974	154619.415
Average rank		2.25	8.83	6.00	5.17	8.83
Overall rank		1	9.5	7	6	9.5

F	Statistics	ICA	BBO	FA	GWO	WOA
F_8	AVE	−8512.5212(6)	−8099.3154(7)	−8969.5712(5)	−6266.9171(9)	−9045.2153(4)
	STD	632.2129	576.9913	678.2812	608.7592	1581.2441
	Min	−8932.5193	−9026.4128	−10615.5481	−7471.6148	−12428.7667
	Max	−7025.3152	−6994.4621	−7397.4209	−5061.2412	−5806.4789
F_9	AVE	3.2897(3)	50.2997(6)	75.5336(8)	4.2873(4)	0(1.5)
	STD	1.8242	14.7106	13.2396	7.8023	0
	Min	0.9950	28.9240	50.3038	5.6843E-14	0
	Max	7.9597	101.2374	108.7309	36.0367	0
F_{10}	AVE	6.0145(8)	0.6089(6)	0.2699(4)	1.0415E-13(3)	3.8488E-15(2)
	STD	2.6799	0.0909	0.3801	1.4897E-14	2.8119E-15
	Min	1.9906	0.3919	0.0214	6.4837E-14	8.8818E-16
	Max	13.9302	0.7566	1.5101	1.2879E-13	7.9936E-15
F_{11}	AVE	11.5789(9)	1.0024(6)	0.0084(3)	0.0028(2)	0.0114(5)
	STD	3.8413	0.0267	0.0047	0.0074	0.0436
	Min	4.0174	0.9207	0.0022	0	0
	Max	19.9567	1.0374	0.0216	0.0234	0.1909
F_{12}	AVE	6.7856(8)	0.0080(1)	3.2959(6)	0.0480(4)	0.0219(3)
	STD	2.8135	0.0037	0.7816	0.0197	0.0135
	Min	1.0241	0.0025	1.9497	0.0135	0.0044
	Max	13.5279	0.0209	4.9055	0.1041	0.0611

Table 5 (continued)

F_{13}	AVE	6.4195(7)	0.1236(3)	0.0019(1)	0.6069(6)	0.5574(5)
	STD	2.3270	0.0315	7.5304E-04	0.2392	0.1735
	Min	2.9857	0.0605	4.2089E-04	0.1144	0.1957
	Max	10.9447	0.1935	0.0045	1.2920	0.9305
Average rank		6.83	4.83	4.50	4.67	3.42
Overall rank		8	5	3	4	2

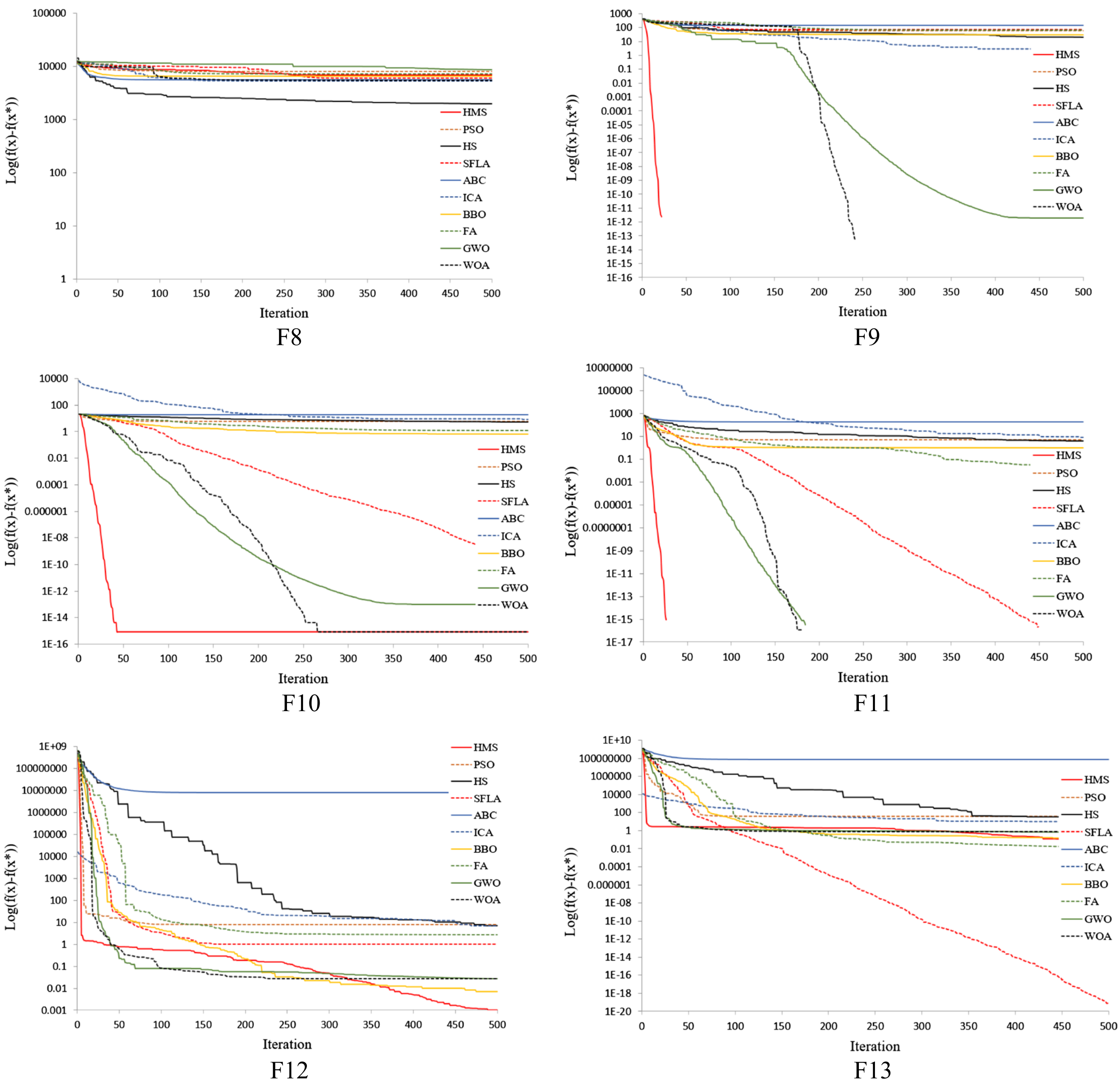


Fig. 12 Convergence curves on multimodal test functions

Table 6 Fixed-dimension multimodal test functions

Function	D	Range	f_{min}
$F_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j+\sum_{i=1}^{2}(x_i - a_{ij})^6}\right)^{-1}$	2	[−65,65]	0.998004
$F_{15}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2+b_i x_2)}{b_i^2+b_i x_3+x_4}\right]^2$	4	[−5,5]	0.00030
$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[−5,5]	−1.0316285
$F_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos x_1 + 10$	2	[−5,5]	0.398
$F_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]$ $\times \left[30 + (2x_1 - 3x_2)^2 + (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)\right]$	2	[−2,2]	3
$F_{19}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2)$	3	[1,3]	−3.86
$F_{20}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$	6	[0,1]	−3.32
$F_{21}(x) = -\sum_{i=1}^{5} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	−10.1532
$F_{22}(x) = -\sum_{i=1}^{7} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	−10.4028
$F_{23}(x) = -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	−10.5363

- Since the HMS algorithm is mostly similar to strategies such as PSO and DE, a comparison between HMS algorithm and these algorithms is given below:
    1) HMS, PSO, and DE algorithms use the solution movement, but the movement strategy is different. In the PSO algorithm, the movement direction of each agent is calculated by using only the two best positions, $pbest_i$, and $gbest$. In the DE algorithm, each agent moves on the basis of the differences among other solutions. But in the HMS algorithm, the agent direction is calculated on the basis of the best solution in the winner cluster. As has been achieved so far, it is likely that the best solution in the winner cluster is not necessarily the best one.
    2) Unlike PSO and DE, HMS searches around a solution using Levy flight.
    3) PSO uses a type of memory for updating the velocity (because of $pbest_i$, and $gbest$). However, HMS is memory-less and only the current position of solutions affects the updating procedure.
- In mental search, we use an operator based on Levy flight. Although Levy flight can also be seen in the Cuckoo search (CS) algorithm. CS algorithm has used Levy flight for generating a new solution by using the following equation:

$$x_i^{t+1} = x_i^t + \alpha.levy \tag{11}$$

where α is a constant (step size).

In the following, the main differences between the HMS and the CS algorithms are explained:

1) The proposed algorithm has used a different strategy to generate a new solution. In (7), a reduction factor, $(2 - iter * (2/\max iter))$, is used to increase the efficiency of the algorithm. It is reduced from 2 to 0. This factor increases the ability of the diversification and the intensification. At the beginning of the algorithm, the reduction factor has a big value which enhances the process of diversification. In the later stages, it meets a reduction, which increases the intensification property
2) The point that parameter β in (7) is different for each solution, leads to an increase in efficiency. Lower β emphasizes on the diversification and higher β shows the intensification.
3) The HMS algorithm uses moving solutions toward the best strategy, which is not observed in the CS algorithm.
4) As previously mentioned, the HMS algorithm uses a clustering algorithm to find the promising region.
5) Eventually, it's noticed that the search ideas of these algorithms are different. CS mimics the behaviour of cuckoos, while HMS simulates the human mental search.

3 Experimental results

In this section, experimental results are presented from different aspects to study the proposed algorithm's efficiency. The test functions can be divided into seven groups: unimodal, multimodal, fix-dimension, high dimensional, composite functions, shifted, and rotated test functions, and classic engineering problems [17, 40–42]. The proposed algorithm is compared with nine state-of-the-art population-based metaheuristic algorithms, which are briefly described below.

- Particle Swarm Optimization (PSO) [19, 20]: It is one of the most well-known population-based metaheuristic algorithms inspired by the social behaviour of birds.

Each search agent updates its position using its velocity, its own best position, and the best position of the overall search agents.

- Harmony Search (HS) [32]: HS is one of the most popular population-based metaheuristic algorithms. It is inspired by the improvisation of music players. The HS algorithm has three main operators: harmony memory consideration, pitch adjustment, and randomization.
- Shuffled Frog-leaping Algorithm (SFLA) [24]: The SFLA algorithm mimics some interesting behaviour of

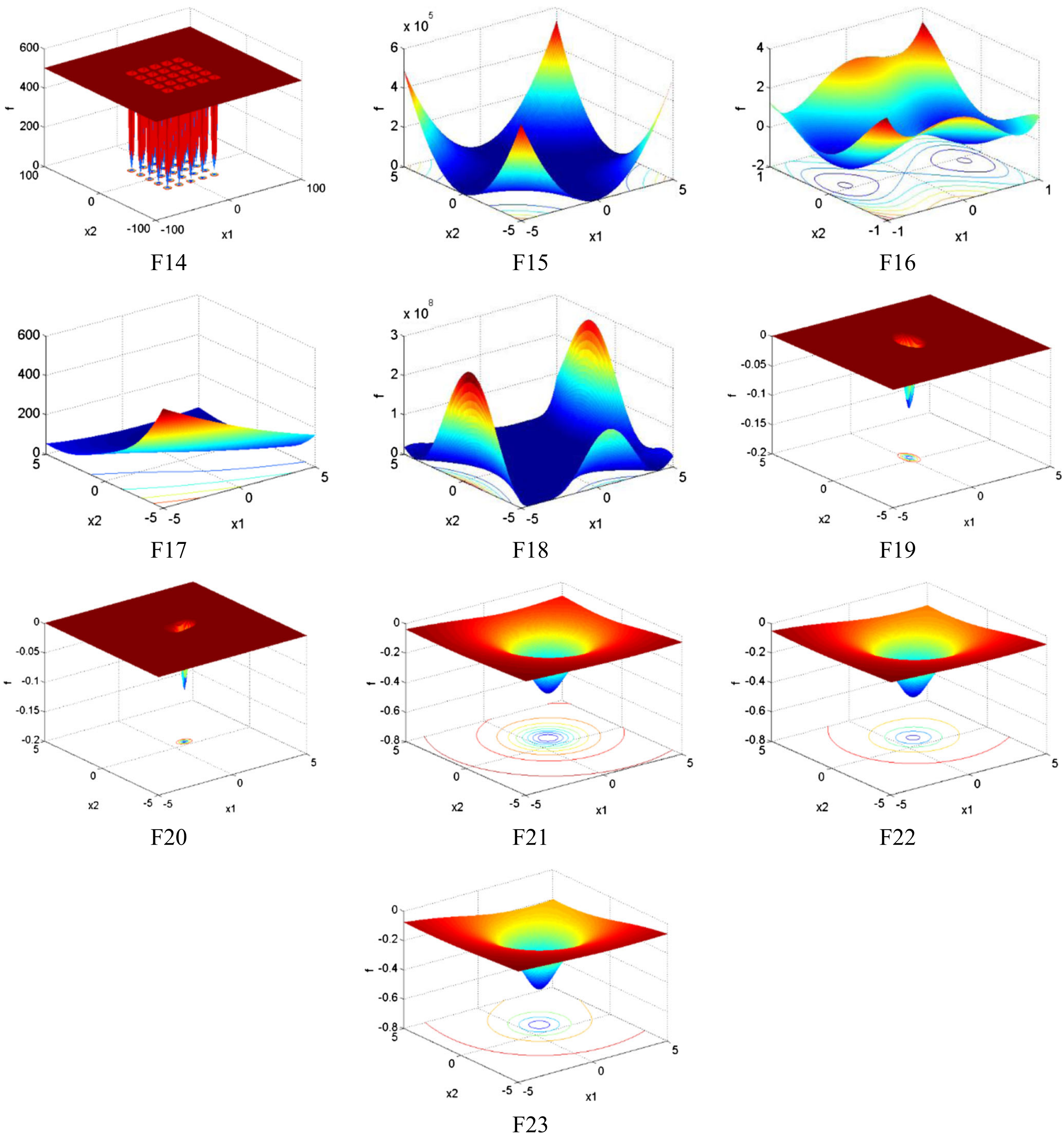


Fig. 13 Search space in the fixed-dimension multimodal test functions

Table 7 The statistical results of fixed-dimension multimodal test functions

F	Statistics	HMS	PSO	HS	SFLA	ABC
F_{14}	AVE	0.9980038(2)	5.1763(8)	0.9980038(2)	1.4598(5)	1.0756(4)
	STD	1.8638E-12	4.2373	1.7178E-11	1.1526	0.2565
	Min	0.9980038	0.9980038	0.9980038	0.9980038	0.9980038
	Max	0.9980038	17.3744	0.9980038	5.9288	1.9920
F_{15}	AVE	5.0453E-04(1)	0.0018(6)	0.0081(10)	0.0011(4)	5.4034E-04(2)
	STD	1.3783E-04	0.0051	0.0095	0.0037	1.7401E-04
	Min	3.2256E-04	3.0749E-04	6.1599E-04	3.0749E-04	3.0751E-04
	Max	9.8307E-04	0.0204	0.0204	0.0204	8.4922E-04
F_{16}	AVE	−1.0316284(4)	−1.0316284(4)	−1.0316283(8)	−1.0316284(4)	−1.0316284(4)
	STD	1.1685E-16	6.3877E-16	1.5457E-07	6.7752E-16	6.7752E-16
	Min	−1.0316284	−1.0316284	−1.03162845	−1.0316284	−1.0316284
	Max	−1.0316283	−1.0316284	−1.03162776	−1.0316284	−1.0316284
F_{17}	AVE	0.397912(1)	0.399142(5)	0.399614(9)	0.399226(7)	0.399793(10)
	STD	1.0994E-08	1.6484E-06	5.6512E-8	4.6514E-08	3.4187E-6
	Min	0.397887	0.388843	0.399513	0.399012	0.399131
	Max	0.398778	0.399617	0.399851	0.399215	0.399981
F_{18}	AVE	3.000000(3)	3.000000(3)	3.900004(7)	3.000000(3)	3.000000(3)
	STD	2.1614E-11	2.0764E-10	4.9295	1.4307E-10	1.1033E-10
	Min	3.000000	3.000000	3.000000	3.000000	3.000000
	Max	3.000000	3.000000	30.000073	3.000000	3.000000
F_{19}	AVE	−3.862782(3)	−3.837014(10)	−3.862782(3)	−3.862782(3)	−3.862782(3)
	STD	2.0844E-15	0.1411	4.2074E-08	2.7201E-12	2.6962E-15
	Min	−3.862782	−3.862782	−3.862782	−3.862782	−3.862782
	Max	−3.862782	−3.089764	−3.862781	−3.862782	−3.862782
F_{20}	AVE	−3.316846(2)	−3.284144(6)	−3.282364(7)	−3.274437(8)	−3.321995(1)
	STD	0.0218	0.0599	0.0570	0.0592	1.3550E-15
	Min	−3.321995	−3.321995	−3.3219951	−3.321995	−3.321995
	Max	−3.203102	−3.137641	−3.2031019	−3.203102	−3.321995
F_{21}	AVE	−10.1530(2)	−7.6542(6)	−5.4098(10)	−7.0665(9)	−10.1531(1)
	STD	9.2202E-04	3.4088	3.6710	3.4523	6.6219E-15
	Min	−10.1532	−10.1532	−10.1532	−10.1532	−10.1531
	Max	−10.1481	−2.6305	−2.6305	−2.6305	−10.1531
F_{22}	AVE	−10.3975(1)	−6.7078(9)	−7.2199(8)	−7.8400(7)	−10.0513(4)
	STD	0.0156	3.8161	3.7141	3.2246	1.3381
	Min	−10.4029	−10.4029	−10.4029	−10.4029	−10.4029
	Max	−10.3383	−1.8376	−2.7519	−2.7659	−5.1288
F_{23}	AVE	−10.5346(1)	−7.7236(7)	−6.1815(10)	−7.4011(8)	−10.1767(4)
	STD	1.0940E-15	3.7855	3.6684	3.6788	1.3609
	Min	−10.5364	−10.5362	−10.5364	−10.5346	−10.5346
	Max	−10.5364	−2.4217	−2.4217	−2.4273	−5.1285
Average rank		2	6.4	7.4	5.8	3.6
Overall rank		1	8	10	5	2
	Statistics	ICA	BBO	FA	GWO	WOA
F_{14}	AVE	0.9980038(2)	6.3540(9)	12.6705(10)	4.5228(7)	2.5079(6)
	STD	0.0100E-10	3.9150	7.0660E-10	4.2310	2.5393
	Min	0.9980038	0.9980038	12.6705	0.9980038	0.9980038
	Max	0.9980038	13.6189	12.6705	12.6705	10.7632

Table 7 (continued)

F_{15}	AVE	0.0020(7)	0.0036(8)	0.0012(5)	0.0085(9)	7.0593E-04(3)
	STD	0.0050	0.0067	3.0550E-04	0.0099	5.3449E-04
	Min	4.6118e-04	3.1627E-04	6.5889E-04	3.0771E-04	3.1176E-04
	Max	0.0204	0.0204	6.5889E-04	0.0209	0.0023
F_{16}	AVE	−1.0316284(4)	−1.0044229(9)	−1.0000000(10)	−1.0316284(4)	−1.0316284(4)
	STD	5.6835E-16	0.1490	3.6302E-11	3.1859E-08	3.5426E-09
	Min	−1.0316284	−1.0316284	−1.0000000	−1.0316284	−1.0316284
	Max	−1.0316284	−0.2154638	−1.0000000	−1.0316284	−1.0316284
F_{17}	AVE	0.399156(6)	0.397937(3)	0.399261(8)	0.397931(2)	0.397942(4)
	STD	1.6153E-06	2.6158E-08	2.6512E-06	2.4912E-04	2.1035E-05
	Min	0.399153	0.397937	0.399153	0.39788	0.397896
	Max	0.399159	0.397937	0.399291	0.399254	0.397945
F_{18}	AVE	3.000000(3)	4.800000(8)	84.000000(10)	5.700025(9)	3.000053(6)
	STD	2.2357E-10	6.850112	3.6866E-08	14.7885	1.0922E-04
	Min	3.000000	3.000000	84.000000	3.000000	3.000000
	Max	3.000000	30.00000	84.000000	84.000002	3.000054
F_{19}	AVE	−3.862712(7)	−3.862782(3)	−3.862727(6)	−3.860972(8)	−3.854679(9)
	STD	6.3254E-10	2.1202E-15	1.4630E-09	0.0027	0.0114
	Min	−3.862712	−3.862782	−3.862727	−3.862781	−3.811439
	Max	−3.862712	−3.862782	−3.862727	−3.854905	−3.862781
F_{20}	AVE	−3.289134(4)	−3.294253(3)	−3.286279(5)	−3.269263(9)	−3.178864(10)
	STD	0.0195	0.0511	0.0605	0.0690	0.2184
	Min	−3.296257	−3.321995	−3.172536	−3.321991	−3.321406
	Max	−3.201554	−3.203102	−3.321995	−3.083268	−2.431429
F_{21}	AVE	−7.9538(5)	−7.1451(7)	−8.1559(4)	−9.9817(3)	−7.1426(8)
	STD	3.6591	3.3762	3.3688	0.9305	3.1426
	Min	−10.02651	−10.1532	−10.1532	−10.1529	−10.1494
	Max	−2.3698	−2.6305	−2.6305	−5.0552	−0.8810
F_{22}	AVE	−8.3581(5)	−5.7709(10)	−10.3729(3)	−10.3813(2)	−7.7928(6)
	STD	2.3652	3.6280	1.4292E-06	7.7919E-04	3.0641
	Min	−8.9362	−10.4029	−10.3729	−10.3826	−10.4022
	Max	−3.6523	−1.8376	−10.3729	−10.3793	−2.7634
F_{23}	AVE	−8.6214(6)	−9.6529(5)	−10.5321(2)	−10.2641(3)	−6.8682(9)
	STD	2.6219	5.3651E-05	9.6785E-07	1.4812	3.3754
	Min	−10.2756	−9.6542	−10.5321	−10.5362	−10.5355
	Max	−6.1028	−9.6513	−10.5321	−2.4216	−1.6763
Average rank		4.9	6.5	6.3	5.6	6.5
Overall rank		3	8	6	4	8

frogs. Actually, this algorithm is a memetic algorithm that combines local and global searches, simultaneously.

- Artificial Bee Colony (ABC) [43]: The ABC algorithm imitates the foraging behaviour of honey bees. There are two groups of bees in the ABC algorithm: scouts who search the area surrounding the nest for new food sources, and onlookers who find a food source through the information shared by the employed artificial bees
- Imperialist Competitive Algorithm (ICA) [33]: This algorithm is inspired by the imperialistic competition among countries. The solutions are divided into two groups based on their power: imperialists and colonies. The two leading operators in this algorithm are assimilation and revolution. Assimilation makes the colonies get closer to the imperialist and revolution is a random sudden change in the position of some solutions.
- Biogeography-based Optimization (BBO) [18]: BBO is based on the mathematical model of biogeography. There are two main operators in the BBO algorithm: migration and mutation. Information is shared among

solutions that depend on the emigration rate and the immigration rate of each solution. In addition, mutation is used to enhance the diversity of the population.

- Firefly Algorithm (FA) [23]: FA mimics the flashing behaviour of fireflies. In this algorithm, for any two fireflies, the less cost value will be attracted by the more cost value.
- Grey Wolf Optimizer (GWO) [25]: The GWO algorithm is one of the studies conducted recently in this field. This algorithm simulates the hunting method of grey wolves. GWO has four primary operators: hunting, searching for prey, encircling, and attacking it. This algorithm has provided competitive performance in comparison to other algorithms.
- Whale Optimization Algorithm (WOA) [26]: This algorithm is one of latest population-based metaheuristic algorithms that mimics the hunting behaviour of humpback whales. WOA includes three operators: searching for prey, encircling prey, and bubble-net foraging behaviour of humpback whales. This algorithm has presented competitive results compared to other state-of-the-art population-based metaheuristic algorithms.

In all the experiments, the population size and the number of iterations are set as 30 and 500, respectively, for all

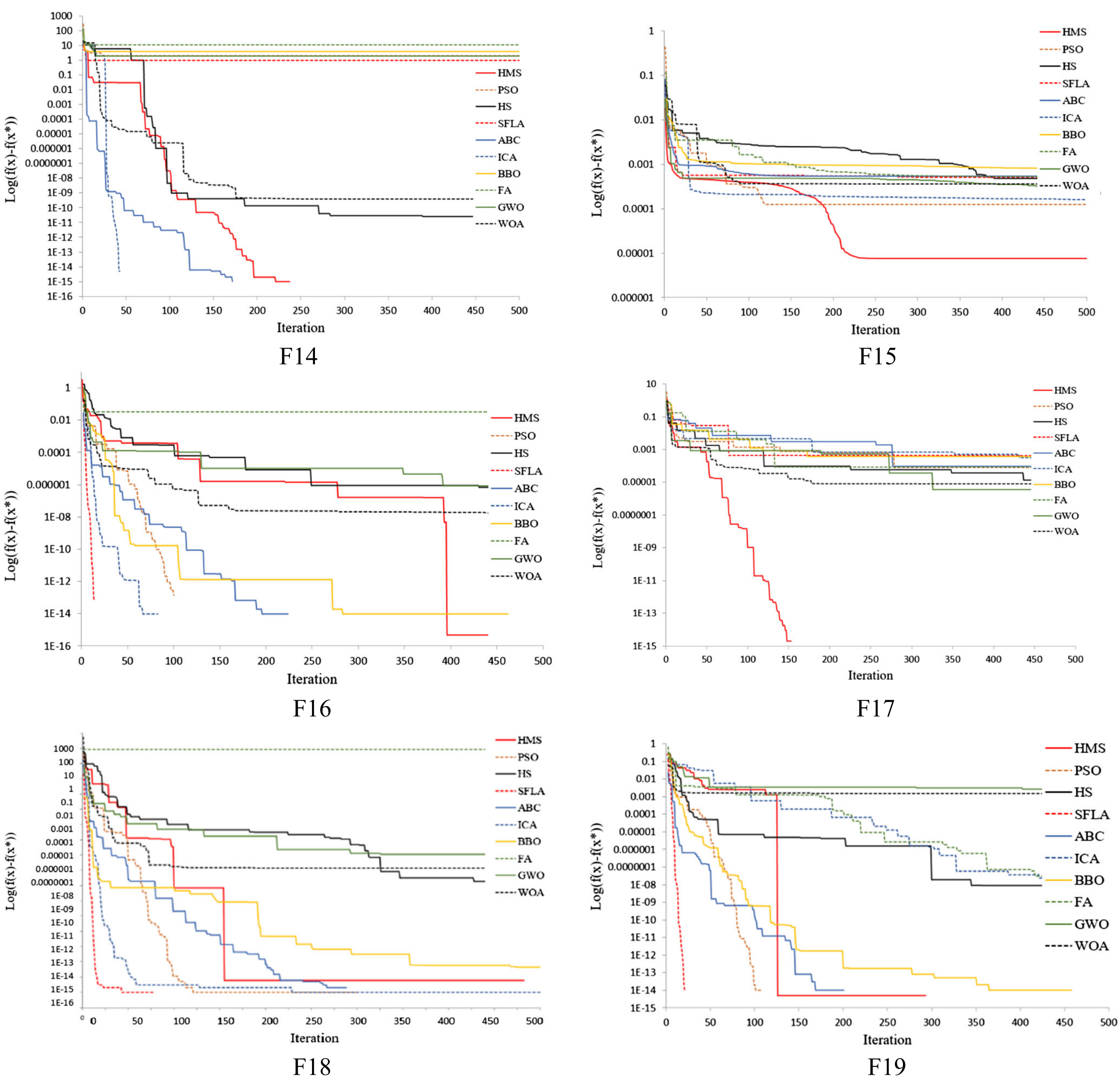


Fig. 14 The convergence curves on fixed-dimension multimodal test functions

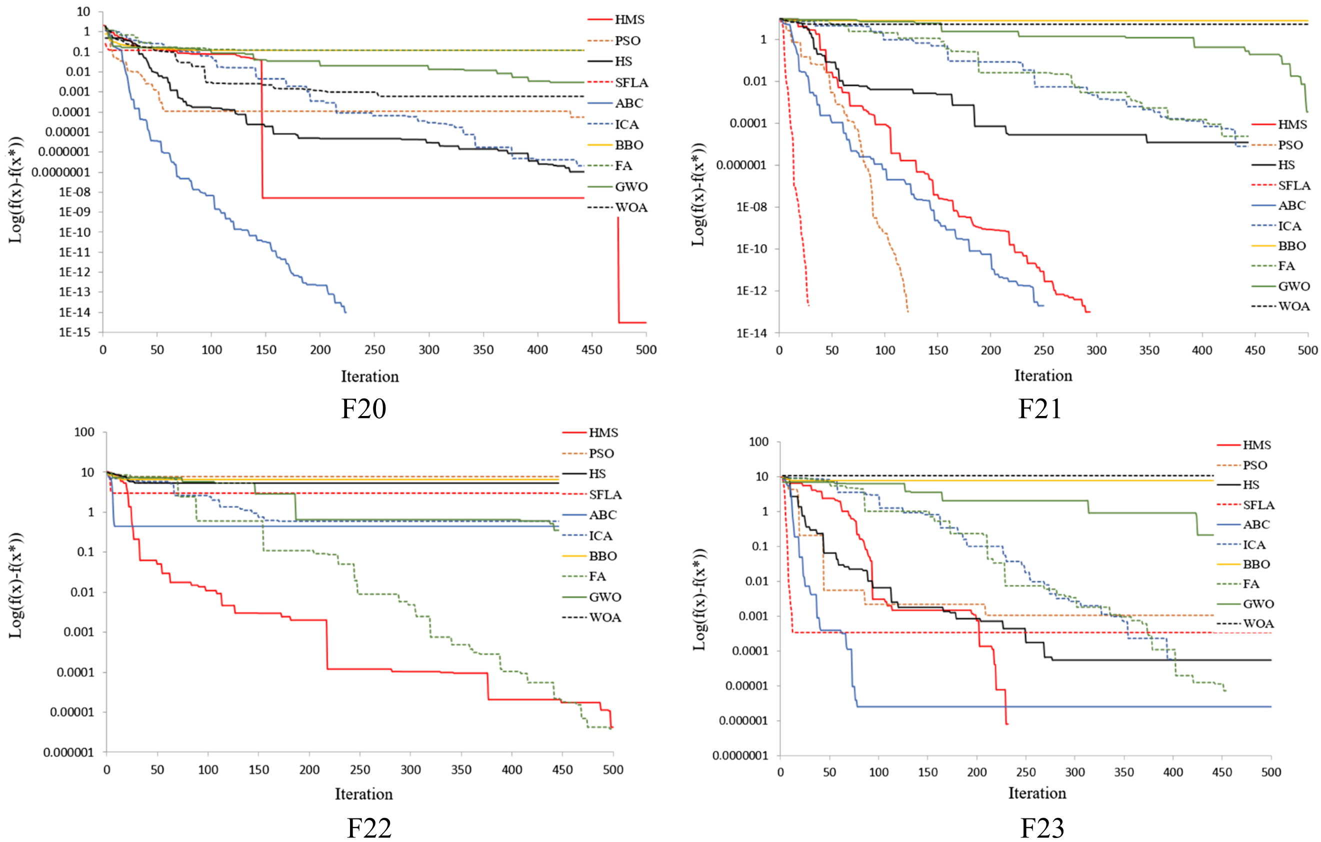


Fig. 14 (continued)

the algorithms. For the HMS algorithm, the number of clusters in the experiments is supposed to be 5. In addition, the default parameter values for C, M_L, and M_H are set as 2, 2, and 10, respectively. The default parameter settings for the other algorithms compared are listed in Table 1. It is worth mentioning that parameter tuning is a trial and error process. Therefore, finding a proper value for the parameters is time-consuming, and so we have not made an effort to find the best parameters. Hence, the analysis of parameters needs more research in the future.

Metaheuristic algorithms are stochastic algorithms. Therefore, the results are not the same in each run of the algorithm. Consequently, each algorithm is run 50 times and the statistical results including average, standard deviation, minimum, and maximum are reported.

3.1 Results of the algorithm on unimodal test functions

The unimodal functions have single global optimum and no local optima, and thus they can be used to evaluate the intensification of algorithms. These test functions are listed in Table 2 in which D is the dimension of the problem, *Range* is the boundary of the search space, and f_{min} is the optimum value. Figure 9 shows the typical 2D plot of test functions. The results of unimodal functions are reported in Table 3. According to this table, the HMS algorithm provides very competitive results in most of the test cases. The HMS algorithm finds the best solution for F1, F2, F3, F4, and F7, but fails to find the best solution for F5 and F6. Nevertheless, its performance is acceptable. These results can also be observed from the convergence curves in Fig. 10. According to Fig. 10, the convergence rate of HMS is high in most cases. This is due to the grouping operator introduced previously

The rank of each algorithm is shown from the smallest average to the highest average in Table 3. To this end, the average rank of each algorithm, and subsequently the overall ranks are reported. The results show that the HMS algorithm achieved the highest rank, which indicates its competence regarding intensification.

3.2 Results of the algorithm on multimodal test functions

Multimodal test functions have many local optima. These functions are useful for examining the diversification of the algorithm and escaping from the local optima. Table 4 presents the details of multimodal test functions. A 2D plot

Table 8 The statistical results of high dimensional test functions

F	Statistics	HMS	PSO	HS	SFLA	ABC
F_{30}	AVE	0(1)	1.5726E04(6)	1.3116E05(8)	4.4589E05(9)	5.0335E05(10)
	STD	0	1.8113E03	6.5347E03	1.1290E04	9.8706E03
	Min	0	1.2377E04	1.1418E05	4.2526E05	4.8096E05
	Max	0	2.0122E04	1.4068E05	4.6985E05	5.2321E05
F_{31}	AVE	0(1)	156.8753(5)	296.3713(8)	5.5107E84(10)	6.2941E25(9)
	STD	0	12.3644	9.6628	1.6213E85	3.01999E26
	Min	0	130.3256	275.0813	7.1127E76	888.9498
	Max	0	179.7425	313.8893	7.3778E85	1.6449E27
F_{32}	AVE	0(1)	1.9921E05(3)	1.7694E06(8)	1.6958E06(6)	1.7433E06(7)
	STD	0	4.9198E04	2.3577E05	1.3892E05	2.9197E05
	Min	0	9.4213E04	1.3185E06	1.4393E06	1.0205E06
	Max	0	3.1652E05	2.2144E06	1.9649E06	2.3252E06
F_{33}	AVE	0(1)	33.1202(3)	82.0031(8)	67.7457(5)	98.8834(10)
	STD	0	2.7025	0.9841	1.0111	0.8226
	Min	0	26.8675	79.1329	65.5161	91.6721
	Max	0	38.7144	83.3742	70.2679	95.1664
F_{34}	AVE	197.0797(1)	4.1310E06(5)	3.2875E08(8)	1.5132E09(9)	1.9762E09(10)
	STD	0.2143	1.1278E06	2.6852E07	6.2308E07	9.5497E07
	Min	197.0241	2.1386E06	2.6988E08	1.3853E09	1.8397E09
	Max	198.8557	7.0199E06	3.8620E08	1.6677E09	2.1839E09
F_{35}	AVE	45.7007(3)	1.5875E04(6)	1.3089E05(8)	4.4758E05(9)	5.0564E05(10)
	STD	0.4879	2.1562E03	8.8120E03	8.4940E03	9.4677E03
	Min	44.8109	1.2304E04	1.1371E05	4.3059E05	4.8402E05
	Max	46.5381	2.1225E04	1.4420E05	4.6198E05	5.2333e05
F_{36}	AVE	1.3296E-05(1)	12.8865(5)	931.9836(8)	4.7249E03(9)	6.0920E03(10)
	STD	1.3816E-05	2.6560	82.1684	196.0919	409.1337
	Min	1.6212E-07	8.4394	795.2210	4.3441E03	4.8272E03
	Max	7.3710E-05	19.6937	1.1026E03	5.1583E03	6.7166E03
F_{37}	AVE	−2.952E04(5)	−2.2858E04(7)	−5.3458E04(3)	−1.1670E04(10)	−1.8499E04(8)
	STD	2.883E03	2.4079E03	1.2999E03	918.5048	795.8096
	Min	−3.531E04	−2.7682E04	−5.6805E04	−1.4452E04	−1.9644E04
	Max	−2.261E04	−1.6963E04	−5.1334E04	−1.0443E04	−1.6678E04
F_{38}	AVE	0(1)	1.4365E03(8)	1.2649E03(7)	3.0613E03(9)	2.6686E03(10)
	STD	0	70.7638	47.0148	36.9157	70.9750
	Min	0	1.2537E03	1.1711E03	2.9731E03	2.4661E03
	Max	0	1.5617E03	1.3548E03	3.1225E03	2.7708E03
F_{39}	AVE	8.8818E-16(1)	11.3312(5)	17.7619(7)	20.7198(10)	20.1281(9)
	STD	0	0.5800	0.1685	0.0362	0.0464
	Min	8.8818E-16	10.3797	17.3136	20.5972	20.0270
	Max	8.8818E-16	12.7463	18.0206	20.7782	20.1940
F_{40}	AVE	0(1.5)	148.8701(6)	1.1755E03(8)	4.0241E03(9)	4.5481E03(10)
	STD	0	20.2986	49.5853	73.8511	112.0118
	Min	0	104.0716	10.0551E03	3.8665E03	4.2888E03
	Max	0	187.4714	1.2707E03	4.1381E03	4.7179E03
F_{41}	AVE	1.0309(3)	4.5756E04(5)	5.0453E08(8)	2.8806E09(9)	4.4458E09(10)
	STD	0.0285	5.1036E04	5.8615E07	1.3324E08	2.6288E08
	Min	0.9667	202.8065	3.8000E08	2.5990E09	3.5062E09

Table 8 (continued)

F_{42}	AVE	19.4815(3)	2.3014E06(5)	1.1824E09(8)	6.1629E09(9)	8.5301E09(10)
	STD	0.1009	1.0705E06	1.1889E08	2.8682E08	4.9642E08
	Min	19.2381	5.4419E05	9.0772E08	5.3535E09	7.1216E09
	Max	19.6715	4.5768E06	1.4271E09	6.5751E09	9.4849E09
Average rank		1.80	5.31	7.46	8.69	9.46
Overall rank		1	5	8	9	10
F	Statistics	ICA	BBO	FA	GWO	WOA
F_{30}	AVE	9.8617E03(5)	1.1125E03(4)	2.3430E04(7)	9.2015E-08(3)	5.3182E-71(2)
	STD	3.2471E03	84.7199	4.2076E03	6.1483E-08	2.5841E-10
	Min	6.2667E03	909.2366	1.6608E04	3.2913E-08	5.1914E-81
	Max	1.7129E04	1.2776E03	3.4369E04	3.1237E-07	1.4152E-69
F_{31}	AVE	228.3489(7)	39.231(4)	227.3975(6)	3.3703E-05(3)	7.8590E-49(2)
	STD	55.1634	2.9843	14.3488	9.0712E-06	2.8757E-48
	Min	148.3171	33.8524	205.2130	2.1939E-05	3.6497E-57
	Max	371.2864	48.5847	258.0721	6.2358E-05	1.2321E-47
F_{32}	AVE	5.9663E05(5)	2.3123E05(4)	2.8345E08(10)	1.8616E04(2)	4.6119E06(9)
	STD	7.8567E04	3.6379E04	3.7962E07	8.4802E03	1.2566E06
	Min	4.4086E05	1.6926E05	2.1154E08	3.9210E03	2.4852E06
	Max	7.8470E05	3.2057E05	3.6650E08	3.3157E04	7.3812E06
F_{33}	AVE	94.2710(9)	36.5689(4)	72.0320(6)	24.2905(2)	81.7410(7)
	STD	1.3616	2.0162	5.3872	6.8181	20.9828
	Min	90.3825	31.9035	61.9658	12.3904	21.4004
	Max	96.3929	40.8605	81.7761	39.6724	98.5179
F_{34}	AVE	8.5196E06(7)	4.4911E04(4)	8.0270E06(6)	198.0339(3)	197.7540(2)
	STD	6.3057E06	6.4151E03	2.7874E06	0.4679	0.1895
	Min	3.4437E06	3.4992E04	4.2563E06	195.8648	197.3493
	Max	2.9701E07	6.3840E04	1.4759E07	198.3127	198.0046
F_{35}	AVE	9.4243E03(5)	1.1154E03(4)	2.4963E04(7)	29.2361(2)	11.4810(1)
	STD	3.6849E03	78.9891	4.3514E03	1.5876	4.0354
	Min	4.8990E03	979.7128	1.5942E04	26.5710	4.4941
	Max	2.1202	1.2831E03	3.4712E04	33.1143	20.8568
F_{36}	AVE	46.7428(6)	0.6002(4)	139.1196(7)	0.0184(3)	0.0031(2)
	STD	39.9407	0.0962	25.6236	0.0073	0.0029
	Min	14.4487	0.4024	89.8936	0.0084	2.2245E-04
	Max	188.4045	0.7417	199.7680	0.0433	0.0129
F_{37}	AVE	−1.25E04(9)	−4.1833E04(4)	−5.3993E04(2)	−2.9357E04(6)	−7.0312E04(1)
	STD	2.2361E03	1.8554E03	1.7927E03	2.4428E03	1.1908E04
	Min	−2.3516E04	−4.4668E04	−5.7363E04	−3.4189E04	−8.3797E04
	Max	−1.0015E04	−3.8311E04	−5.0220E04	−2.4358E04	−4.9327E04
F_{38}	AVE	629.8474(4)	871.7112(5)	1.2319E03(6)	22.6712(3)	7.5791E-15(2)
	STD	48.5355	51.0479	82.1383	14.6199	4.1513E-14
	Min	552.0894	733.9129	1.0634E03	1.1713E-05	0
	Max	709.7808	968.1199	1.4319E03	61.8622	2.2737E-13
F_{39}	AVE	17.8343(8)	4.5262(4)	14.5738(6)	2.2673E-05(3)	4.3225E-15(2)
	STD	0.9127	0.1128	0.6897	6.2867E-06	2.3756E-15
	Min	14.9149	4.2370	13.2917	1.3240E-05	8.8818E-16
	Max	18.9794	4.7747	15.8347	3.8594E05	7.9936E-15
	Max	1.0976	2.3392E05	6.2997E08	3.0460E09	4.8667E09

Table 8 (continued)

F_{40}	AVE	94.7528(5)	11.1871(4)	206.8672(7)	0.0028(3)	0(1.5)
	STD	32.2501	0.8330	30.1673	0.0110	0
	Min	63.7153	9.3736	133.2408	7.0073E-09	0
	Max	222.6046	12.4996	299.5214	0.0515	0
F_{41}	AVE	1.6951E07(7)	19.8444(4)	1.4453E05(6)	0.5522(2)	0.0628(1)
	STD	3.7016E07	3.7585	1.8727E05	0.0494	0.0317
	Min	1.5168E05	12.1975	6.4645E03	0.4588	0.0160
	Max	1.8536E08	30.1200	8.7124E05	0.7060	0.1458
F_{42}	AVE	2.6582E07(7)	335.7491(4)	7.7316E06(6)	16.9810(2)	6.8141(1)
	STD	2.8040E07	110.9103	4.3347E06	0.6140	2.4383
	Min	3.3414E06	191.0075	3.3504E06	15.8937	2.5568
	Max	9.9531E07	658.2434	2.1896E07	18.2067	11.7262
Average rank		6.46	4.08	6.31	2.85	2.58
Overall rank		7	4	6	3	2

Table 9 The complex test functions

Function	D	Range	f_{min}
$C_{24}(CF1)$ $f_1, f_2, ..., f_{10}$ = Sphere Function $[\sigma_1, \sigma_2, ..., \sigma_{10}] = [1, 1, ..., 1]$ $[\lambda_1, \lambda_2, ..., \lambda_{10}] = [5/100, 5/100, ..., 5/100]$	10	$[-5, 5]$	0
$C_{25}(CF2)$ $f_1, f_2, ..., f_{10} = Griewank's$ Function $[\sigma_1, \sigma_2, ..., \sigma_{10}] = [1, 1, ..., 1]$ $[\lambda_1, \lambda_2, ..., \lambda_{10}] = [5/100, 5/100, ..., 5/100]$	10	$[-5, 5]$	0
$C_{26}(CF3)$ $f_1, f_2, ..., f_{10} = Griewank's$ Function $[\sigma_1, \sigma_2, ..., \sigma_{10}] = [1, 1, ..., 1]$ $[\lambda_1, \lambda_2, ..., \lambda_{10}] = [1, 1, ..., 1]$	10	$[-5, 5]$	0
$C_{27}(CF4)$ $f_1, f_2 = Ackley's$Function, f_3, f_4 = Rastrigin's Function, f_5, f_6 = Weierstrass Function f_7, f_8 = Griewank's Function, $f_9 f_{10}$ = Sphere Function $[\sigma_1, \sigma_2, ..., \sigma_{10}] = [1, 1, ..., 1]$ $[\lambda_1, \lambda_2, ..., \lambda_{10}] = [5/32, 5/32, 1, 1, 5/0.5, 5/100, 5/100, 5/100, 5/100]$	10	$[-5, 5]$	0
$C_{28}(CF5)$ $f_1, f_2 = Rastrigin's$ Function, f_3, f_4 = Weierstrass's Function, f_5, f_6 = Weierstrass Function f_7, f_8 = Ackley's Function, $f_9 f_{10}$ = Sphere Function $[\sigma_1, \sigma_2, ..., \sigma_{10}] = [1, 1, ..., 1]$ $[\lambda_1, \lambda_2, ..., \lambda_{10}] = [1/5, 1/5, 5/0.5, 5/0.5, 5/100, 5/100, 5/32, 5/32, 5/100, 5/100]$	10	$[-5, 5]$	0
$C_{29}(CF6)$ $f_1, f_2 = Rastrigin's$ Function, f_3, f_4 = Weierstrass's Function, $f_5, f_6 = Griewank's$ Function f_7, f_8 = Ackley's Function, $f_9 f_{10}$ = Sphere Function $[\sigma_1, \sigma_2, ..., \sigma_{10}] = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]$ $[\lambda_1, \lambda_2, ..., \lambda_{10}] = [0.1 \times 1/5, 0.2 \times 1/5, 0.3 \times 5/0.5, 0.4 \times 5/0.5, 0.5 \times 5/100,$ $0.6 \times 5/100, 0.7 \times 5/32, 0.8 \times 5/32, 0.9 \times 5/100, 1 \times 5/100]$	10	$[-5, 5]$	0

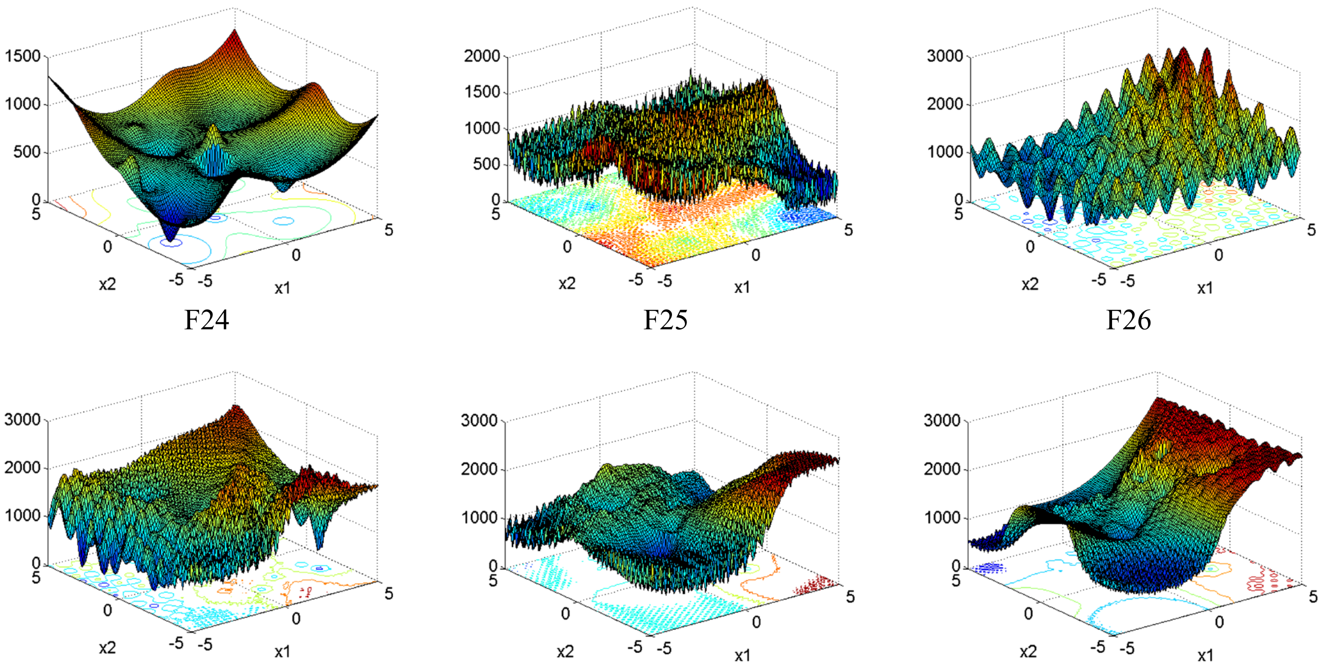


Fig. 15 Search space in the complex test functions

of the search space in these functions is shown in Fig. 11. Table 5 presents the results of the HMS algorithm on multimodal test functions. The HMS algorithm gives the best performance for functions F9, F10, and F11 and the second best performance in most of the remaining test functions. The rank of algorithms is computed on the basis of the average value of each test function, and the average and overall ranks are calculated for each test function. In the context of this table, the HMS algorithm earns the best rank compared to the others. As per the characteristics of multimodal functions, it can be said that the HMS algorithm has an excellent capability for diversification. The convergence curves of the test functions are shown in Fig. 12, which indicates that the HMS algorithm converges faster than the other algorithms in most of the cases. Although HMS and WOA can find the best optimum in F9 function, according to Fig. 12, the HMS algorithm can find the best optimum in less than 30 iterations.

3.3 Results of the algorithm on fixed-dimension multimodal functions

These functions are similar to the multimodal functions; however, their dimensions are low and fixed. Table 6 presents the details of the test functions. A 2D plot of the functions is shown in Fig. 13. The results of the HMS algorithm on fixed-dimension multimodal functions are given in Table 7 and Fig. 14. Table 7 shows that the HMS algorithm has the best performance on the majority of the test functions. The rank of each algorithm is computed in Table 7. It can be observed from the last row of this table, that the HMS algorithm is ranked first for the fixed-dimension functions.

3.4 Results of the algorithm on high dimensional test functions

To demonstrate the competence of the HMS algorithm, the results for solving the 200 dimensional problems on unimodal and multimodal test functions in the previous subsections are reported in Table 8. As seen in Table 8, the HMS algorithm has the best optimization algorithm in nine out of 13 test functions. WOA can also find the best solution for five out of 13 test functions. This table also shows that the HMS algorithm was ranked first, while the second rank went to the WOA algorithm. The weak performance of most of the algorithms shows that high-dimensional test functions are very challenging, which certifies the HMS algorithm's high efficiency to solve high-dimensional problems.

3.5 Results of the algorithm on complex test functions

This category of test functions comprises composite functions, which are very challenging because they have enormous number of local optima and various shapes for different

Table 10 The statistical results of the complex test functions

F	Statistics	HMS	PSO	HS	SFLA	ABC
F_{24}	AVE	43.3333(1)	151.2346(9)	50.0130(4)	46.6667(3)	45.5340(2)
	STD	50.4007	98.9541	77.6785	68.1445	42.6686
	Min	2.8939E-22	0.3016	0.0051	0	1.6959
	Max	100.0000	500.0832	300.0063	200	122.2131
F_{25}	AVE	30.5536(1)	216.6786(9)	129.1837(4)	126.7211(3)	92.07469(2)
	STD	39.6146	127.8241	86.5841	87.9543	59.5877
	Min	1.9756	24.4316	1.7992	1.4742	21.0993
	Max	209.7773	461.9401	309.2404	308.6885	211.6246
F_{26}	AVE	160.7135(1)	368.2971(8)	174.0393(2)	269.8949(6)	530.8543(9)
	STD	42.4334	140.8033	41.0074	82.7444	52.9639
	Min	98.4181	172.5069	107.6542	124.3234	423.2817
	Max	250.1514	812.3136	248.6738	462.6605	688.6266
F_{27}	AVE	321.0389(1)	522.7619(7)	345.8215(2)	399.8532(4)	547.5697(8)
	STD	23.8997	171.2765	90.0195	88.1694	57.0612
	Min	285.2585	304.4968	290.8705	286.6253	405.5549
	Max	371.0226	900.6063	601.8049	606.4522	645.6092
F_{28}	AVE	14.7996(1)	189.6117(9)	109.0081(4)	18.0720(2)	130.2136(6)
	STD	30.1686	168.9923	112.4644	90.9136	39.2891
	Min	1.6755	22.3700	2.1381	4.1406	7.7485
	Max	108.0322	547.7524	502.6115	405.7694	185.1681
F_{29}	AVE	615.2485(2)	830.7096(8)	851.8544(9)	751.8907(4)	552.7838(1)
	STD	192.7996	147.7384	125.8700	195.1207	72.9954
	Min	400.6273	506.2891	501.0607	500.3244	425.8555
	Max	903.4616	903.1997	908.1489	902.7782	716.5288
Average rank		1.17	8.33	4.17	3.67	4.67
Overall rank		1	9	3	2	4
F	Statistics	ICA	BBO	FA	GWO	WOA
F_{24}	AVE	76.6667(6)	70.0000(5)	294.8472(10)	92.8042(8)	87.5969(7)
	STD	81.7200	98.7857	0.1026	111.3791	127.3527
	Min	1.8985E-19	2.2188E-05	294.7147	1.1314	0.8964
	Max	200.00	300.0001	295.1253	418.7817	503.0055
F_{25}	AVE	134.1509(5)	153.1039(6)	258.4430(10)	155.2829(7)	200.1003(8)
	STD	95.7920	100.5286	8.4312	98.1962	102.1304
	Min	16.8001	3.3119	247.1308	10.1578	31.9708
	Max	360.5961	318.1269	274.2228	374.5334	355.7350
F_{26}	AVE	177.8931(3)	295.4858(7)	589.0137(10)	220.6713(4)	247.6690(5)
	STD	106.0588	127.1419	30.5648	71.8672	116.4241
	Min	25.6857	108.3607	527.9047	124.1943	129.6501
	Max	415.4764	637.2571	682.1685	456.0947	483.1406
F_{27}	AVE	511.3155(6)	509.5334(5)	900.0000(10)	360.2287(3)	588.1781(9)
	STD	129.3741	88.4501	2.9630E-06	96.0163	161.6945
	Min	329.0808	331.8615	900.0000	251.9535	294.6066
	Max	754.4257	717.1701	900.0000	727.3915	900.0000
F_{28}	AVE	117.7276(5)	105.8573(3)	264.2366(10)	150.2026(7)	176.8292(8)
	STD	130.5783	125.7549	2.5525	129.2170	126.7631
	Min	7.5025	1.3674	260.6912	7.2744	25.5658
	Max	553.0262	536.9510	270.1089	525.0230	548.0652
F_{29}	AVE	805.7795(6)	685.0116(3)	912.5634(10)	795.5623(5)	811.6532(7)
2-9	STD	153.3447	201.1673	44.8205	174.6982	171.0266
	Min	500.4118	500.0081	822.9428	501.0221	507.5448
	Max	907.1116	903.3026	938.8868	906.5216	922.5448
Average rank		5.17	4.83	10	5.67	7.33
Overall rank		6	5	10	7	8

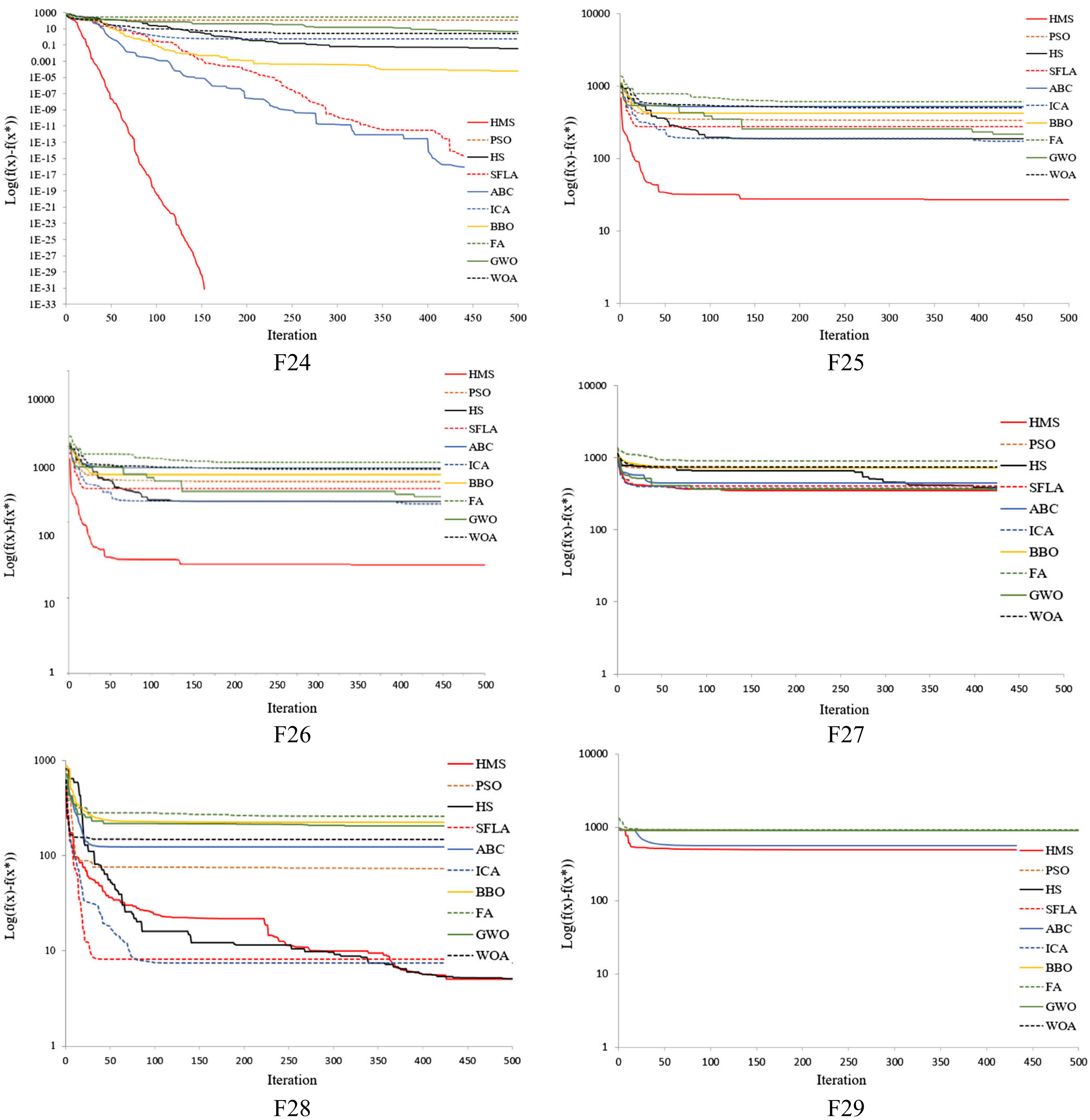


Fig. 16 The convergence curves on the complex test functions

regions of the search space. The details of the composite functions are presented in Table 9. The search space of the complex test functions is shown in Fig. 15. These functions can check the balance between intensification and diversification, which is a cardinal characteristic of population-based metaheuristics. Table 10 presents the statistical results of 50 independent runs. According to this table, the HMS algorithm is very competitive in comparison to others; it especially presents the most efficient results for all the composite functions except for F29, for which it gives the second best result. It can also be observed that HMS has the overall first rank to solve composite test functions. It is evident that the HMS algorithm can balance intensification and diversification more than the other algorithms compared. Figure 16 shows that the HMS algorithm converges toward the optimum solution faster than the others.

Table 11 Shifted and rotated test functions

F		D	Range	f_{min}
	Unimodal Functions			
F_{43}	Shifted Sphere Function	30	[−100,100]	+450
F_{44}	Shifted Schwefel's problem 1.2	30	[−100,100]	+450
F_{45}	Shifted Rotated High Conditioned Elliptic Function	30	[−100,100]	+450
F_{46}	Shifted Schwefel's Problem 1.2 with Noise in Fitness	30	[−100,100]	+450
F_{47}	Schwefel's problem 2.6 with Global Optimum on Bounds	30	[−100,100]	−310
	Basic Functions			
F_{48}	Shifted Rosenbrock's Function	30	[−100,100]	+390
F_{49}	Shifted Rotated Griewanl's Function without Bounds	30	[0,600]	−180
F_{50}	Shifted Rotated Ackley's with Global Optimum on Bounds	30	[−32,32]	+260
F_{51}	Shifted Rastrigin's Function	30	[−5,5]	−330
F_{52}	Shifted Rotated Rastrigin's Function	30	[−5,5]	−330
F_{53}	Shifted Rotated Weierstrass's Function	30	[−0.5,0.5]	+90
F_{54}	Schwefel's Problem 2.13	30	[−100,100]	−460
	Expanded Functions			
F_{55}	Expanded Extended Griewank's+Rosenbrock's(F8F2)	30	[−3,1]	−130
F_{56}	Expanded Rotated Extended Scaffe's F6	30	[−100,100]	−300
	Hybrid Composition Function			
F_{57}	Hybrid Composition Function 1	30	[−5,5]	+120

3.6 Results of the algorithm on shifted and rotated test functions

To further verify the effectiveness of the HMS algorithm, a set of 15 CEC 2005 optima shifted and rotated test functions were evaluated in this subsection. In these functions, the optimum is shifted or rotated to other locations to provide the more challenging test functions. A summary of these functions is shown in Table 11, and their further details can be found in [44]. The 2D plot of these functions can be seen in Fig. 17. Table 12 presents the experimental results of the algorithms. From the results, the HMS algorithm performs the best on seven test algorithms and the second best in five test functions. The BBO algorithm was the best in two test functions. The rank of each algorithm is computed in Table 12. It can be seen that the HMS algorithm ranks first for the shifted and rotated test functions.

3.7 Nonparametric statistical analysis results

In this section, nonparametric statistical methods are applied to compare the HMS algorithm with the other algorithms. Such statistical results are necessary due to the stochastic nature of metaheuristics [45]. Nonparametric statistical tests are divided into two classes: pairwise comparisons and multiple comparisons. The pairwise comparison is a comparison between two algorithms, while the multiple comparisons compare more than two algorithms. In this paper, two well-known nonparametric tests, the Wilcoxon signed rank test (pairwise comparison) and the Friedman test (multiple comparisons), are conducted for this purpose. Further details about nonparametric statistical tests for metaheuristics can be found in [45].

The null hypothesis H_0 states that there is no difference between two algorithms, whereas the alternative hypothesis H_1 indicates a difference. A level of statistical significance (α) is used to determine the probability of the rejecting null hypothesis while it is true. If the p-value is less than α then H_0 is rejected.

Table 13 shows the results of the Wilcoxon signed ranked test. However, this P-value is not suitable because the same data is evaluated several times [45, 46] and an accumulated error is obtained because of the combination of pairwise comparisons. Thus, a post hoc test is necessary to control Family-Wise Error Rate (FWER), which is the probability that at least one comparison test would reject a correct null hypothesis. For this purpose, the well-known Holm method [47] is applied for post hoc analysis (see [45] for more information about this method). Table 12 presents the unadjusted and adjusted P-values. It is evident that the HMS algorithm gives a significant improvement over all the compared algorithms with a significance level of 0.05.

Another statistical test conducted to demonstrate the effectiveness of HMS algorithm is Friedman test. As mentioned earlier, it is a multiple comparison test. The details of this statistical approach can be found in [45]. Table 14 shows the average ranks computed with the Friedman statistical test. As per this table, the HMS algorithm provides the best rank. In Table 14, the P-value substantially depicts existence of significant differences among the examined algorithms.

3.8 Sensitivity analysis

The following will discuss the sensitivity analysis on the input parameters of the HMS algorithm. Sensitivity analysis is the study of how different values of input parameters impact the output. This analysis represents the robustness of the HMS algorithm to changes in the input parameters. It is clear that fewer values are preferable because it shows easier tuning procedure.

For this purpose, F10 and F13 are studied to evaluate sensitivity analysis (this approach is inspired by the method applied in [46, 48] for sensitivity analysis). The reason for choosing the F10 function is that the HMS algorithm provides the best results. Therefore, this discussion aims to show that this superiority is kept after changing the value of the input parameters. The reason for choosing F13 function is that the HMS algorithm does not provide the best result, and so the following discussion shows that the HMS algorithm can provide better results by changing the input parameters.

As shown in Figs. 18 and 19, the results are highly dependent on the number of clusters. Hence, finding the appropriate number of clusters is necessary to achieve a better solution. F7 is a unimodal function. Therefore, it has one local optimum; so, a large number of clusters do not need to find a suitable region. As seen in Fig. 18, the performance decreases by increasing the number of clusters. F13 has many local optima. Therefore, more clusters are

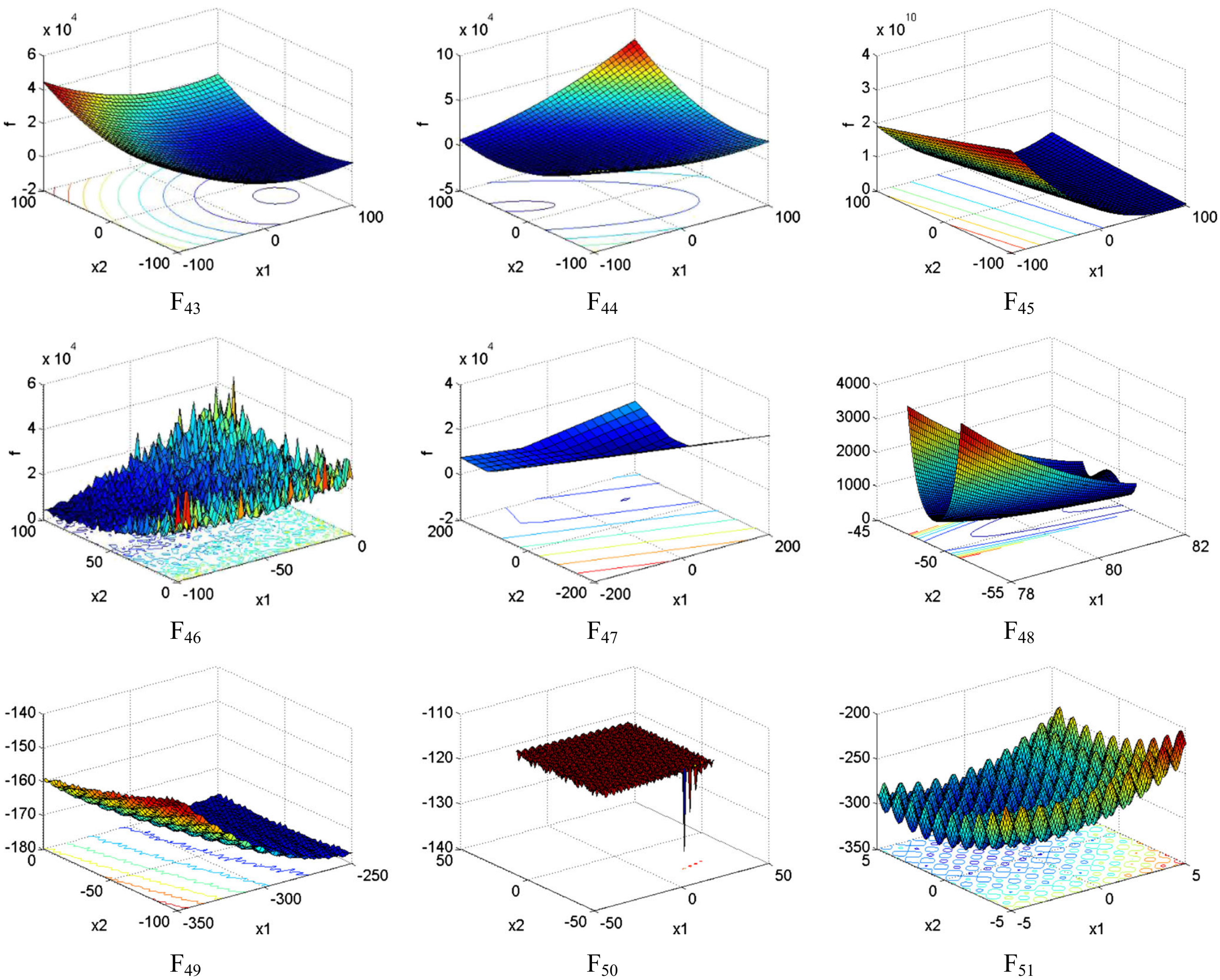


Fig. 17 Search space in shifted and rotated test functions

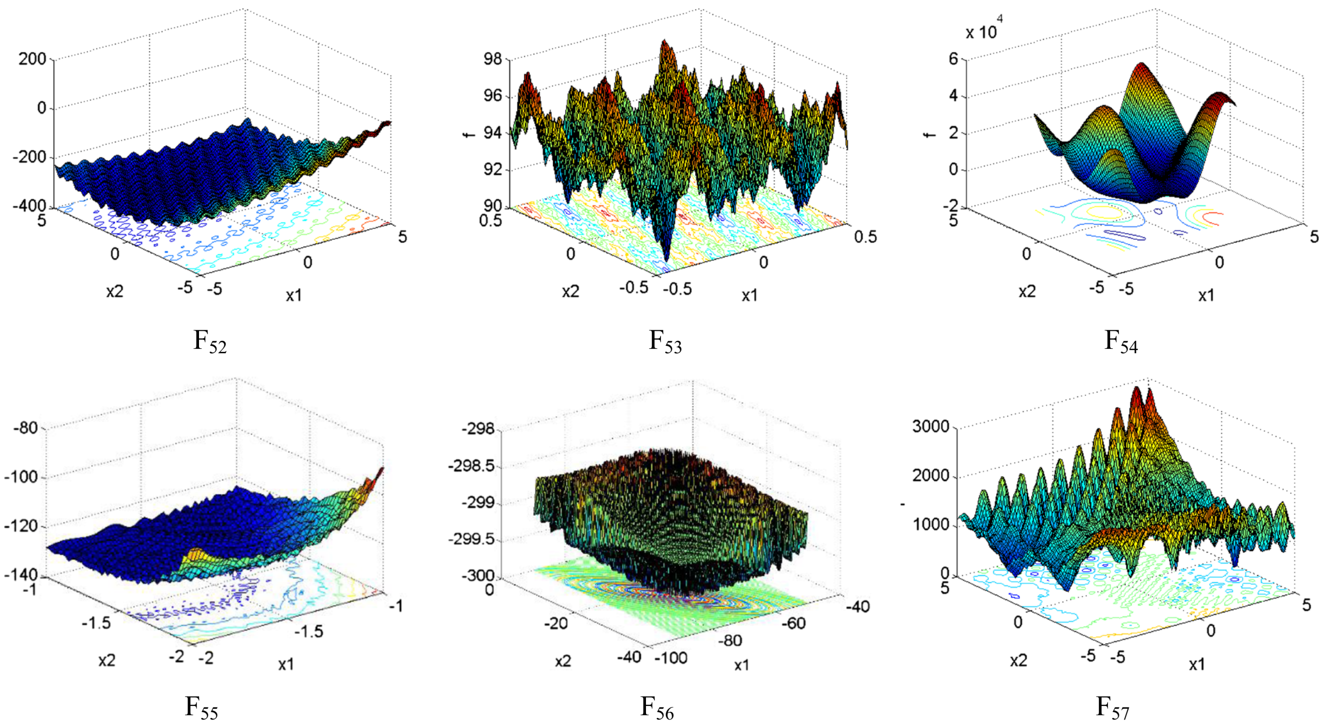


Fig. 17 (continued)

needed. It is also seen in Fig. 19, as expected, that increasing the number of clusters enhances the performance of the algorithm.

Another parameter in the HMS algorithm is the maximum and minimum mental processes. To analyse the sensitivity of these parameters, the algorithm was tested with different values. According to the experimental results, the maximum and minimum mental processes significantly affect the performance. According to Figs. 18 and 19, increasing the number of minimum and maximum mental processes enhances performance because of more searches being carried out around a specific point.

3.9 Classical engineering problems

In this section, the performance of HMS algorithm is evaluated with three engineering design problems: pressure vessel design, welded beam design, and three-bar truss design. These problems are constrained. Constraint optimization is the process of finding the optimal value of an objective function with respect to some constraints on decision variables. There are two types of constraints on decision variables: inequality and equality constraints. To solve these problems, a constrained handling approach should be added to the optimization algorithm. There are different approaches for constrained handling such as penalty functions, repair algorithms, and special operators. Finding a proper constraint handling approach is out of the scope of this work, and therefore one of the simplest methods, death penalty, is chosen to optimize a constraint problem. To this end, death penalty will assign a big objective function value if a solution breaks any of constraints.

3.9.1 Pressure vessel design

Pressure vessel design is a popular engineering problem in the literature (Fig. 20). The aim of this problem is to minimize the fabrication cost of a vessel. This problem has four variables and four equality constraints. The four variables are thickness of the head (T_h), thickness of the shell (T_s), inner radius(R), and length of the cylindrical section without considering the head (L). These parameters are shown in Fig. 20. This problem is formulated as follows:

$$
\begin{aligned}
&\text{Consider} && \vec{x} = [x_1, \mathrm{x}_2, \mathrm{x}_3, \mathrm{x}_4] = [T_s, T_h, \mathrm{R}, \mathrm{L}],\\
&\text{Minimize} && f(\vec{x}) = 0.6224\mathrm{x}^1\mathrm{x}_3\mathrm{x}_4 + 1.7781\mathrm{x}_2\mathrm{x}_3^2 + 3.1661\mathrm{x}_1^2\mathrm{x}_4 + 19.84\mathrm{x}_1^2\mathrm{x}^3,\\
&\text{Subject to} && g_1(x) = -x_1 + 0.0193x_3 \le 0,\\
& && g_2(x) = -x_3 + 0.0095x_3 \le 0,\\
& && g_3(x) = -\pi x_3^2 x_4 - \tfrac{4}{3}\pi x_3^3 + 1296000 \le 0,\\
& && g_4(x) = x_4 - 240 \le 0,\\
&\text{Variable range:} && 0 \le x_1 \le 99,\\
& && 0 \le x_2 \le 99,\\
& && 0 \le x_3 \le 200,\\
& && 0 \le x_4 \le 200,
\end{aligned}
\qquad (12)
$$

Table 12 The statistical results of shifted and rotated test functions

F	Statistics	HMS	PSO	HS	SFLA	ABC
F_{43}	AVE	−450.0000(1.5)	903.4478(6)	−235.6340(5)	−447.6251(4)	5.2336E04(10)
	STD	2.1354E-12	1.2332E03	54.7480	3.9495E-14	7.8579E03
	Min	−450.0000	−342.2565	−335.8402	−447.6251	3.4501E04
	Max	−450.0000	4.8145E03	−91.1460	−447.6251	6.9546E04
F_{44}	AVE	4.1116E03(3)	4.1802E03(4)	2.1442E04(6)	−434.6152(1)	4.9181E04(8)
	STD	2.2208E03	3.1802E03	6.1949E03	2.6125E-11	8.5048E03
	Min	1.7264E03	2.1202E03	1.2443E04	−434.6152	3.2241E04
	Max	9.7375E03	1.8275E04	3.8348E04	−434.6152	6.2820E04
F_{45}	AVE	2.3768E07(1)	2.5736E07(3)	1.0655E08(6)	2.6154E07(4)	3.4381E08(9)
	STD	1.3493E07	1.1615E07	4.0750E07	1.2541E07	1.5891E08
	Min	4.3849E06	1.1940E07	4.7368E07	4.9652E07	6.1469E07
	Max	5.0466E07	5.9494E07	2.0924E08	5.6149E06	7.5380E08
F_{46}	AVE	1.2486E04(2)	1.1601E04(1)	2.8580E04(4)	2.3145E05(8)	8.0669E04(6)
	STD	5.0319E03	6.0649E03	7.9527E03	6.3256E04	1.0211E04
	Min	4.9709E03	6.1974E03	1.5183E04	3.9654E04	4.8929E04
	Max	2.3580E04	3.5965E04	4.9162E04	5.3659E05	9.6495E04
F_{47}	AVE	5.2124E03(1)	7.9969E03(4)	5.6031E03(3)	4.1965E04(10)	2.7923E04(9)
	STD	2.3239E03	2.0394E03	950.3237	2.3146E04	2.1070E03
	Min	480.5589	3.4181E03	3.5666E03	780.6195	2.3921E04
	Max	9.6619E03	1.2942E04	7.4150E03	8.9812E04	3.1913E04
F_{48}	AVE	3.5238E06(3)	5.1803E07(5)	1.5974E06(2)	6.6519E06(4)	1.6459E10(9)
	STD	1.0954E07	8.4480E07	5.8541E05	2.3654E07	5.3045E09
	Min	765.7768	2.1011E05	6.0825E05	956.3215	7.5652E09
	Max	4.4678E07	3.9960E08	2.4887E06	6.6214E07	3.0001E10
F_{49}	AVE	2.5334E03(2.5)	2.7967E03(7)	2.5444E03(5)	2.9562E03(8)	3.5756E03(6)
	STD	0.0186	180.6535	5.6160	6.6521	75.7184
	Min	2.5334E03	2.5334E03	2.5358E03	2.9532E03	3.4401E03
	Max	2.5334E03	3.2519E03	2.5632E03	2.9862E03	3.6789E03
F_{50}	AVE	−119.1862(2)	−118.9290(7)	−118.9007(8)	−118.2153(10)	−119.1328(3)
	STD	0.1737	0.0754	0.0467	0.0952	0.0697
	Min	−119.5394	−119.2061	−118.9937	−118.9325	−119.7904
	Max	−118.9447	−118.8113	−118.8338	−118.1172	−119.5000
F_{51}	AVE	−256.9424(2)	−165.1436(5)	−294.6861(1)	−112.6214(6)	−108.3558(7)
	STD	18.6105	29.8408	6.8411	25.5241	36.2118
	Min	−288.6811	−216.7463	−305.3924	−140.6176	−174.7872
	Max	−215.2024	−112.5330	−272.9448	−87.3621	−27.5370
F_{52}	AVE	−183.7616(2)	−53.7168(5)	−94.4106(4)	7.7495E04(10)	177.9152(7)
	STD	31.1374	43.6019	17.4468	271.9707	66.4876
	Min	−265.3385	−170.3420	−136.4036	7.6848E04	1.3167
	Max	−117.3145	14.1257	−63.8771	7.8065E04	271.9397
F_{53}	AVE	114.0726(1)	118.6942(4)	132.6930(8)	8.8141E04(10)	124.2766(6)
	STD	4.0225	3.9264	1.2428	33.4959	1.9378
	Min	108.7577	109.5464	129.3807	8.8073E04	120.1839
	Max	127.8069	124.3175	135.1521	8.8218E04	127.2957
F_{54}	AVE	3.3920E04(4)	9.5614E05(9)	4.0175E05(6)	7.4508E05(7)	1.4557E04(1)
	STD	1.0668E04	2.3017E05	8.5910E04	5.6959E05	6.1302E03
	Min	1.7378E04	5.1171E05	2.4836E05	2.5106E03	4.3329E03
	Max	5.3321E04	1.3951E06	6.1135E05	1.4746E06	2.7820E04

Table 12 (continued)

F	Statistics	HMS	PSO	HS	SFLA	ABC
F_{55}	AVE	−124.6913(1)	−111.3386(6)	−119.1150(5)	−123.3829(3)	−41.1588(9)
	STD	1.6775	3.3220	2.0039	1.3633	36.3903
	Min	−127.3296	−121.6412	−122.1549	−127.7733	−109.9746
	Max	−121.0190	−106.4985	−115.4047	−121.5490	25.7872
F56	AVE	−286.9038(1)	−286.7639(3)	−286.1862(9)	−286.5418(4)	−286.4730(6)
	STD	0.3453	0.3497	0.1630	0.3045	0.2351
	Min	−287.8873	−287.6708	−286.5805	−287.3669	−287.1164
	Max	−286.3678	−286.2418	−285.8933	−286.1534	−286.1384
F57	AVE	423.4599(1)	788.0670(7)	516.8825(3)	531.6254(4)	930.9383(10)
	STD	80.6935	101.0248	34.1908	73.26	137.3253
	Min	309.8642	644.8685	401.2835	450.3652	589.7312
	Max	622.1217	1.0560E03	595.5613	600.6214	1.1241E03
Average rank		1.87	5.1	5	6.2	7.1
Overall rank		1	4	3	6	8
F	Statistics	ICA	BBO	FA	GWO	WOA
F_{43}	AVE	−450.0000(1.5)	−447.2722(3)	4.4705E04(9)	1.3410E04(8)	3.1509E03(7)
	STD	9.4105E-06	0.6148	148.5581	3.5560E03	1.066E03
	Min	−450.0000	−448.3444	4.4369E04	7.5799E03	1.5981E03
	Max	−450.0000	−445.5684	4.4957E04	2.1599E04	5.5716E03
F_{44}	AVE	6.6837E03(5)	463.1837(2)	1.3242E06(10)	2.3535E04(7)	1.0420E05(9)
	STD	2.9684E03	335.6409	7.8393E04	6.5639E03	2.2595E04
	Min	3.0555E03	−142.3967	1.2153E06	1.3116E04	4.7013E04
	Max	1.5192E04	1.1306E03	1.5145E06	5.0439E04	1.3992E05
F_{45}	AVE	2.6214E08(8)	8.8601E06(2)	4.1796E08(10)	5.4708E07(5)	1.5142E08(7)
	STD	9.5621E-06	3.3548E06	2.8676E07	3.6531E07	5.7674E07
	Min	2.6214E08	3.1327E06	3.6122E08	1.0227E07	5.8670E07
	Max	2.6214E08	1.7904E07	4.7231E08	1.4808E08	2.9525E08
F_{46}	AVE	6.6214E05(9)	1.9975E04(3)	3.4759E06(10)	3.1940E04(5)	2.0044E05(7)
	STD	1.6254E-06	9.5093E03	5.9106E05	1.0160E04	6.7777E04
	Min	6.6214E05	5.2718E03	2.5791E06	1.4198E04	9.1839E04
	Max	6.6214E05	4.5265E04	4.8898E06	6.8342E04	3.7125E05
F_{47}	AVE	1.2651E04(6)	5.5950E03(2)	2.2592E04(7)	1.1515E04(5)	2.3648E04(8)
	STD	2.6541E-07	1.3572E03	530.9176	3.2941E03	4.1496E03
	Min	1.2651E04	3.6354E03	2.1607E04	5.6785E03	1.4263E04
	Max	1.2651E04	9.6217E03	2.3584E04	1.8497E04	3.2199E04
F_{48}	AVE	5.3242E07(6)	2.2963E03(1)	2.2766E10(10)	6.4994E08(8)	2.3799E08(7)
	STD	4.3265E-06	1.8645E03	1.8100E08	9.4421E08	2.0841E08
	Min	5.3242E07	865.8989	2.2463E10	2.5051E07	4.0199E07
	Max	5.3242E07	7.4972E03	2.3137E10	3.7042E09	1.0744E09
F_{49}	AVE	6.6521E04(10)	2.5334E03(2.5)	4.6183E03(9)	392.4048(1)	2.5398E03(4)
	STD	2.3254E-06	0.1533	12.9747	209.6337	12.4783
	Min	6.6521E04	2.5334E03	4.5999E03	65.0165	2.5334E03
	Max	6.6521E04	2.5342E03	4.6514E03	808.2970	2.5770E03
F_{50}	AVE	−118.6232(9)	−119.0103(5)	−119.3484(1)	−118.9376(6)	−119.0289(4)
	STD	0.0712	0.0667	0.1018	0.0675	0.0862
	Min	−119.002	−119.1169	−119.6322	−119.1075	−119.2696
	Max	−118.4678	−118.8466	−119.1574	−118.8336	−118.8969

Table 12 (continued)

F	Statistics	HMS	PSO	HS	SFLA	ABC
F_{51}	AVE	2.5760E04(10)	−256.1489(3)	−92.4980(8)	−198.3721(4)	−46.5267(9)
	STD	2.6164E-12	22.1705	15.4849	27.3246	56.0462
	Min	2.5760E04	−284.6623	−113.7938	−240.6488	−181.3916
	Max	2.5760E04	−190.4792	−57.1800	−132.9924	57.8575
F_{52}	AVE	4.6281E04(9)	−202.6800(1)	102.0053(6)	−113.5040(3)	188.4374(8)
	STD	3.2658E-10	28.6949	18.8055	59.8083	72.5813
	Min	4.6281E04	−257.2305	67.6118	−219.9535	61.4092
	Max	4.6281E04	−120.2844	160.1057	3.3148	363.9211
F_{53}	AVE	8.7940E04(9)	118.7584(5)	115.6625(2)	116.2943(3)	129.8792(7)
	STD	2.9601E-11	2.6734	2.3523	6.9588	2.2229
	Min	8.7940E04	114.4213	111.2818	103.6269	121.8271
	Max	8.7940E04	127.2091	119.8181	134.1975	133.6288
F_{54}	AVE	3.6401E04(3)	3.4787E04(5)	2.9238E04(2)	1.0303E06(10)	7.4994E05(8)
	STD	1.8570E04	1.8215E04	1.9832E04	4.0719E05	2.2676E05
	Min	1.1686E04	6.6879E03	6.8073E03	2.4828E05	3.4020E05
	Max	7.4605E04	9.7064E04	9.5814E04	1.5548E06	1.1264E06
F_{55}	AVE	−123.9024(2)	−122.8261(4)	−13.2207(10)	−95.7207(8)	−99.9633(6)
	STD	0.6872	1.7160	8.8421	7.5058	6.2787
	Min	−127.9000	−125.1976	−28.4172	−105.1569	−110.0725
	Max	−124.4251	−118.0617	7.9105	−79.0267	−83.2286
F56	AVE	−286.3202(8)	−285.5187(10)	−286.5392(5)	−286.8498(2)	−286.3400(7)
	STD	0.3085	0.0653	0.3948	0.4003	0.3474
	Min	−286.9987	−285.6874	−287.5455	−287.7181	−287.1975
	Max	−285.9708	−285.4062	−285.8028	−286.0779	−285.8948
F57	AVE	447.9970(2)	552.0418(5)	843.3938(8)	654.4284(6)	909.9902(9)
	STD	134.6112	73.0665	166.0494	77.3970	196.0838
	Min	137.3106	405.8580	654.0254	523.3246	613.6923
	Max	634.7209	645.6321	1.1477E03	834.4494	1.2191E03
Average rank		6.5	3.57	7.13	5.4	7.13
Overall rank		7	2	9.5	5	9.5

Table 13 Wilcoxon signed ranked test

Comparison	Unadjusted P-value	Adjusted P-value
HMS versus PSO	9.0844E-10	8.1759E-09
HMS versus HS	3.8263E-07	1.5305E-06
HMS versus SFLA	1.2925e-08	7.7550E-08
HMS versus ABC	3.7766e-09	2.6436E-08
HMS versus ICA	9.2590e-10	8.1759E-09
HMS versus BBO	1.2699e-05	1.2699E-05
HMS versus FA	5.3733e-08	2.6866E-07
HMS versus GWO	7.3693e-07	2.2107E-06
HMS versus WOA	1.2231e-06	2.4462E-06

Table 14 The Friedman ranks

Algorithms	Friedman
HMS	1.82
PSO	6.46
HS	6.30
SFLA	5.93
ABC	7.11
ICA	5.94
BBO	4.73
FA	6.74
GWO	4.46
WOA	5.30
P-value	2.4648E-25

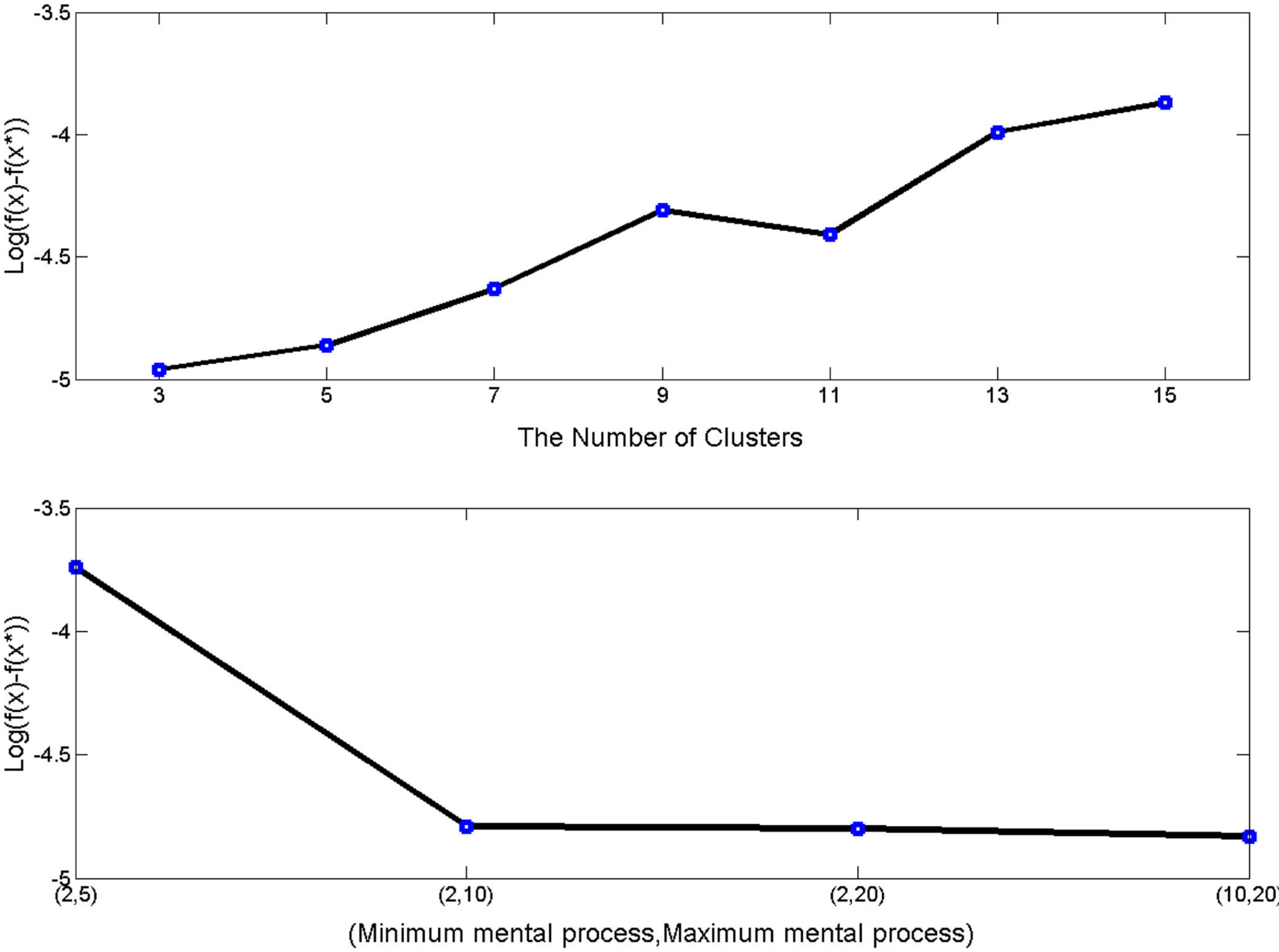


Fig. 18 Plot of input parameters corresponding to F7 test function solved by HMS

This problem is popular among researchers. The literature review shows that metaheuristic and mathematical approaches have been applied to optimize the variable of this problem. Some metaheuristic methods applied on this problem are GA [49–51] PSO [52] DE [53] ES [54], ACO [55] GWO [25], and WOA [26]. The mathematical methods are augmented Lagrangian multiplier [56] and branch-and-bound [57]. The results of the HMS algorithm compared to the other algorithms in literature are presented in Table 15. As can be observed, the HMS algorithm outperforms all the other compared algorithms.

3.9.2 Welded beam design

The aim of this problem is to design a welded beam with the lowest fabrication cost. The overall structure of this problem is shown in Fig. 21. There are four variables to be optimized, including the weld thickness (h), the length of the bar's attached part (l), the bar's height (t), and the bar's thickness (b). These variables should satisfy seven constraints.

This problem can be written as follows:

$$\begin{aligned}
&\text{Consider} \quad \vec{x} = [x_1, x_2, x_3, x_4] = [h, l, t, b], \qquad (13)\\
&\text{Minimize} \quad f(\vec{x}) = 1.10471 x_l^2 x_2 + 0.04811 x_3 x_4 (14.0 + x_2),\\
&\text{Subject to} \quad g_1(\vec{x}) = \tau(\vec{x}) - \tau_{\max} \le 0,\\
&\qquad g_2(\vec{x}) = \alpha(\vec{x}) - \alpha_{\max} \le 0,\\
&\qquad g_3(\vec{x}) = \delta(\vec{x}) - \delta_{\max} \le 0,
\end{aligned}$$

$$\begin{aligned}
&g_4(\vec{x}) = x_1 - x_4 \le 0,\\
&g_5(\vec{x}) = P - P_c(\vec{x}) \le 0,\\
&g_6(\vec{x}) = 0.125 - x_1 \le 0,\\
&g_7(\vec{x}) = 0.10471 x_1^2 + 0.04811 x_3 x_4 (14.0 + x_2) - 5.0 \le 0\\
&\tau(\vec{x}) = \sqrt{(\dot{\tau}) + 2\dot{\tau}\dot{\tau}\frac{x_2}{2R} + (\dot{\tau})^2},\\
&\dot{\tau} = \frac{P}{\sqrt{2x_1 x_2}},\\
&\dot{\tau} = \frac{MR}{J},\\
&M = P(L + \frac{x_2}{2}),
\end{aligned}$$

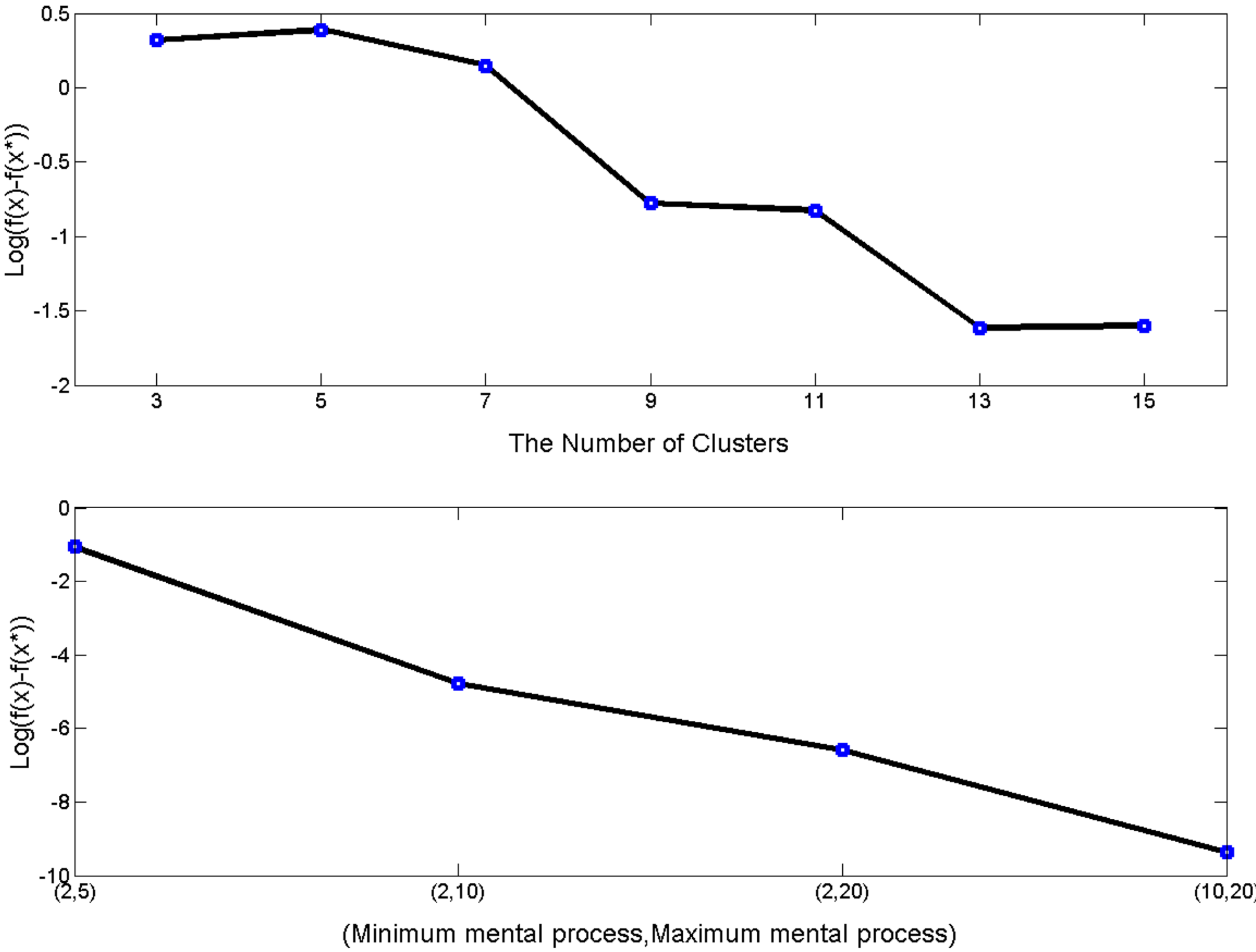


Fig. 19 Plot of input parameters corresponding to F13 test function solved by HMS

Table 15 Comparison results for the pressure vessel design problem

Algorithm	Optimum variables				Optimum cost
	T_s	T_h	R	L	
HMS	0.8313	0.4113	43.0432	165.4836	5992.2729
WOA [26]	0.8125	0.4375	42.0982	176.6389	6059.7410
GWO [25]	0.8125	0.4345	42.0891	176.7587	6051.5639
GA [50]	0.8125	0.4345	40.3239	200.0000	6288.7445
GA [49]	0.8125	0.4375	42.0973	176.6540	6059.9463
GA [51]	0.9375	0.5000	48.3290	112.6790	6410.3811
ES [54]	0.8125	0.4375	42.0980	176.6405	6059.7456
DE [53]	0.8125	0.4375	42.0984	176.6376	6059.7456
PSO [52]	0.8125	0.4375	42.0912	176.7465	6061.0777
ACO [55]	0.8125	0.4375	41.1036	176.5726	6059.0888
Lagrangian multiplier [56]	1.1250	0.6250	58.2910	43.6900	7198.0428
Branch and bound [57]	1.1250	0.6250	47.7000	117.7010	8129.1036

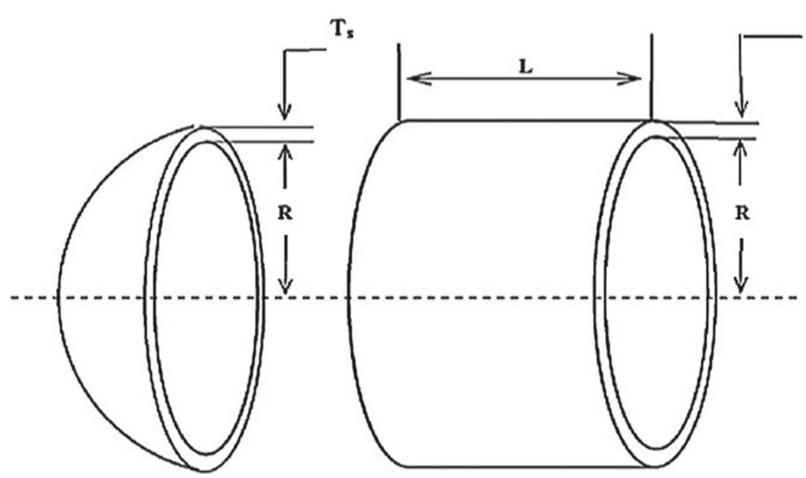


Fig. 20 Pressure vessel design problem

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}$$

$$\text{where: } J = 2\left\{\sqrt{2}x_1 x_2 \left[\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\},$$

$$\alpha(\vec{x}) = \frac{6PL}{x_4 x_3^2},$$

$$\delta(\vec{x}) = \frac{6PL^3}{E x_3^2 x_4},$$

$$P_c(\vec{x}) = \frac{4.013E\sqrt{\frac{x_3^2 x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right),$$

$$P = 6000lb,\ L = 14in.,\ \delta_{\max} = 0.25in,$$

$$E = 13600psi,\ \mathrm{E} = 12 \times 10^6 psi,$$

$$\tau_{\max} = 13600psi,\ \sigma_{\max} = 30000psi$$

$$\text{Variable range: } = 0.1 \le x_1 \le 2,$$

$$= 0.1 \le x_2 \le 10,$$

$$= 0.1 \le x_3 \le 10,$$

$$= 0.1 \le x_4 \le 2$$

Keeping in mind the popularity of this problem, numerous lines can be found on it in the literature. Some are GA [58, 59], HS [60], CPSO [61], GWO [25], WOA [26], Richardson's random method [62], Simplex method [62], Davidon-Fletcher-Powell [62], and Griffith and Stewart's successive linear approximation [62]. The comparison results are presented in Table 16. As seen from the table, the HMS algorithm can find a design with the minimum cost.

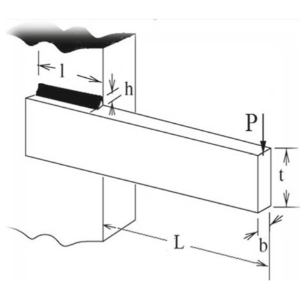


Fig. 21 Welded beam design problem

3.9.3 Three-bar truss design problem

The objective of the three-bar truss design problem is to design a truss with the minimum weights. The overall structure of this problem is shown in Fig. 22. It is popular in the literature [63–66] due to its difficult constraint search space [64, 67]. In fact, constraints are the most important issues in designing a truss. It can be formulated as below:

$$\text{Consider} \quad \vec{x} = [x_1, x_2] = [A_1, A_2],$$

$$\text{Minimize} \quad f(\vec{x}) = (2\sqrt{2}x_1 + x_2)^* l,$$

$$\text{Subject to} \quad g_1(\vec{x}) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1 x_2} P - \sigma \le 0,$$

$$g_2(x) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1 x_2} P - \sigma \le 0,$$

$$g_3(x) = \frac{1}{\sqrt{2}x_2 + x_1} P - \sigma \le 0,$$

$$\text{Variable range: } 0 \le x_1, x_2 \le 1,$$

$$\text{where} \quad l = 100cm \quad p = 2kN/cm^2 \quad \sigma = 2kN/cm^2 \tag{14}$$

Table 17 compares the results of the HMS algorithm with the other algorithms. The results of the algorithms demonstrate that the HMS algorithm outperforms four algorithms. In addition, HMS shows very close results to MFO, DEDS, and PSO-DE algorithms. It shows that the HMS algorithm can solve the three-bar truss design problem effectively. It should be noted that the TSA algorithm does not satisfy one of the constraints [67].

Table 16 Comparison results for the welded beam design problem

Algorithm	Optimum variables				Optimum cost
	h	l	t	b	
HMS	0.2054	3.4777	9.0386	0.2057	1.7255
WOA [26]	0.2054	3.4843	9.0374	0.2063	1.7305
GWO [25]	0.2056	3.4784	9.0368	0.2058	1.7262
GA [58]	0.1829	4.0483	9.3666	0.2059	1.8240
GA [59]	0.2489	6.1730	8.1789	0.2533	2.4331
HS [60]	0.2442	6.2231	8.2915	0.2443	2.3807
CPSO [52]	0.2024	3.5442	9.0482	0.2057	1.7315
Richardson's random [62]	0.4575	4.7313	5.0853	0.6600	4.1185
Simplex [62]	0.2792	5.6256	7.7512	0.2796	2.5307
Davidon-fletcher-powell [62]	0.2434	6.2552	8.2915	0.2444	2.3841
linear approximation [62]	0.2444	6.2189	8.2915	0.2444	2.3815

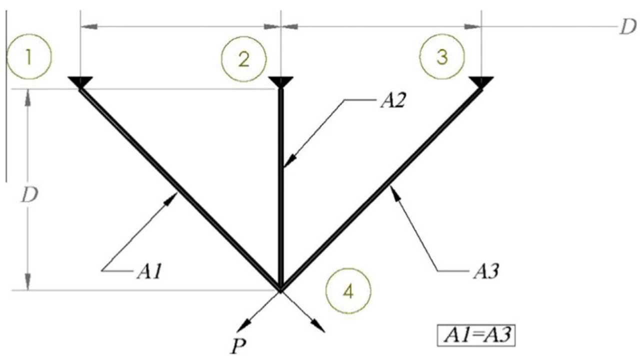


Fig. 22 Three-bar truss design problem [64]

Table 17 Comparison results for three-bar truss design problem

Algorithms	Optimum variables		Optimum cost
	x_1	x_2	
HMS	0.7883	0.4094	263.8960
MFO [64]	0.7882	0.4095	263.8960
DEDS [63]	0.7887	0.4083	263.8958
PSO-DE [65]	0.7887	0.4082	263.8958
CS [67]	0.7887	0.4090	263.9716
Ray and Sain [66]	0.795	0.395	264.3
Tsa [68]	0.788	0.408	263.68(infeasible)

4 Conclusion

In this study, we proposed a simple but powerful population-based metaheuristic called human mental search (HMS). HMS is inspired by the exploration strategy of the bid space in online auctions. The HMS algorithm employs three important behaviours namely mental search, grouping, and moving. First, each solution produces other new solutions based on Levy flight. Levy flight simultaneously enhances the quality of the diversification and the intensification. The solutions are then placed in the different groups. We used K-means algorithm, a well-known clustering algorithm for grouping the solutions. In moving strategy, each solution moves toward the best group. To verify the efficiency of the HMS algorithm, several test functions that are commonly applied in the literature were conducted, including unimodal, multimodal, fix-dimension, complex, high dimensional, shifted, and rotated functions. The performance of the HMS algorithm was compared with nine state-of-the-art population-based metaheuristics. The results revealed that the HMS algorithm provides superior performance in most cases. Moreover, three classic engineering problems were evaluated to show the HMS algorithm's efficiency in unknown and challenging search spaces.

Also, some nonparametric statistical methods, including Wilcoxon signed rank test and Friedman test, were provided. The results of the Wilcoxon signed rank test followed by the post hoc analysis indicated an improvement of HMS over all the compared algorithms with the level of significance being $\alpha = 0.05$. The Friedman rank test showed that HMS was ranked first. Moreover, the P-value confirmed the existence of significant differences among the compared algorithms.

Another test conducted was the sensitivity analysis on the input parameters of HMS. The sensitivity analysis showed that HMS kept its superiority even with changing the parameters values. The sensitivity analysis on another test function revealed that HMS can improve performance by changing the settings.

The reasons for the HMS algorithm's high performance could be: (1) mental search operator as it searched around a solution that enhances intensification property, (2) grouping operator because it quickly finds the promising regions of search space, (3) reduction factor as it allows the HMS algorithm to move smoothly from diversification to intensification, (4) moving operator because other solutions move toward the promising region using this operator, and (5) the type of the HMS algorithm because it's a population-based algorithm that intrinsically takes advantages of high diversification and local optima avoidance compared to the single-based metaheuristic algorithms.

Not withstanding the significant performance of the HMS algorithm, the following points should be considered for future investigations:

- Some test functions are used for evaluating the performance of the HMS algorithm. In the future, the performance of HMS on some more problems can be examined, especially real world applications such as scheduling, knapsack problem, controller design, microwave design problem, and water resource management.
- In the current work, the k-means clustering algorithm is used for grouping the solutions. Other clustering algorithms such as hierarchical clustering and distribution-based clustering algorithms can be applied as well.
- One of the most important parameters of the HMS algorithm is the number of clusters. Some clustering algorithms can find the optimum number of clusters automatically. For future research, it is recommended to use such algorithms for clustering.
- In the current work, the performance of the HMS algorithm is experimentally conducted only on standard test functions. The convergence of the HMS algorithm should be analysed theoretically by dynamic systems in the future.

Acknowledgments Authors are grateful to University of Kashan for supporting this work under grant NO. 572086.

References

1. Glover F (1989) Tabu search-part I. ORSA J Comput 1(3):190–206
2. Hwang C-R (1988) Simulated annealing: theory and applications. Acta Appl Math 12(1):108–111
3. Kirkpatrick S (1984) Optimization by simulated annealing: Quantitative studies. J Stat Phys 34(5-6):975–986
4. Mladenović N, Hansen P (1997) Variable neighborhood search. Comput Oper Res 24(11):1097–1100
5. Mitchell M, Holland JH (1993) When will a genetic algorithm outperform hill-climbing?
6. Stützle T (1998) Local search algorithms for combinatorial problems. Darmstadt University of Technology PhD Thesis
7. Lai C-C, Chang C-Y (2009) A hierarchical evolutionary algorithm for automatic medical image segmentation. Expert Syst Appl 36(1):248–259
8. Li Y, Jiao L, Shang R, Stolkin R (2015) Dynamic-context cooperative quantum-behaved particle swarm optimization based on multilevel thresholding applied to medical image segmentation. Inf Sci 294:408–422
9. Tayarani-N M-H, Yao X, Xu H (2015) Meta-Heuristic Algorithms in Car Engine Design: A Literature Survey. IEEE Trans Evol Comput 19(5):609–629
10. Yuwono M, Qin Y, Zhou J, Guo Y, Celler BG, Su SW (2016) Automatic bearing fault diagnosis using particle swarm clustering and Hidden Markov Model. Eng Appl Artif Intell 47:88–100
11. MousaviRad S, Tab FA, Mollazade K (2012) Application of imperialist competitive algorithm for feature selection: a case study on bulk rice classification. Int J Comput Appl 40(16)
12. Holland JH (1975) Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. U Michigan Press
13. Hansen N, Müller SD, Koumoutsakos P (2003) Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evol Comput 11(1):1–18
14. Rechenberg I (1994) Evolution strategy. Computational intelligence: Imitating life 1
15. Storn R, Price K (1997) Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11(4):341–359
16. Baluja S (1994) Population-based incremental learning. a method for integrating genetic search based function optimization and competitive learning. DTIC Document
17. Yao X, Liu Y, Lin G (1999) Evolutionary programming made faster. IEEE Trans Evol Comput 3(2):82–102
18. Simon D (2008) Biogeography-based optimization. IEEE Trans Evol Comput 12(6):702–713
19. Shi Y, Eberhart R (1998) modified particle swarm optimizer. In: The 1998 IEEE International Conference on Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence. IEEE, pp 69–73
20. Eberhart RC, Kennedy J (1995) A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, New York, NY, pp 39–43
21. Karaboga D (2005) An idea based on honey bee swarm for numerical optimization. Technical report-tr06, Erciyes university, engineering faculty, computer engineering department
22. Yang X-S, Deb S (2009) Cuckoo search via Lévy flights. In: World Congress on Nature & Biologically Inspired Computing, 2009. NaBIC 2009. IEEE, pp 210–214
23. Łukasik S, Żak S (2009) Firefly algorithm for continuous constrained optimization tasks. In: Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems. Springer, pp 97–106
24. Eusuff M, Lansey K, Pasha F (2006) Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization. Eng Optim 38(2):129–154
25. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
26. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67

27. Aarts E, Korst J (1988) Simulated annealing and Boltzmann machines
28. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: A gravitational search algorithm. Inf Sci 179(13):2232–2248
29. Eskandar H, Sadollah A, Bahreininejad A, Hamdi M (2012) Water cycle algorithm–A novel metaheuristic optimization method for solving constrained engineering optimization problems. Comput Struct 110:151–166
30. Sadollah A, Bahreininejad A, Eskandar H, Hamdi M (2013) Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems. Appl Soft Comput 13(5):2592–2612
31. Mirjalili S (2016) SCA: A sine cosine algorithm for solving optimization problems. Knowledge-Based Systems
32. Geem ZW, Kim JH, Loganathan G (2001) A new heuristic optimization algorithm: Harmony search. Simulation 76(2):60–68
33. Atashpaz-Gargari E, Lucas C (2007) Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In: 2007. CEC 2007. IEEE Congress on Evolutionary Computation. IEEE, pp 4661–4667
34. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1(1):67–82
35. Radicchi F, Baronchelli A, Amaral LA (2012) Rationality, irrationality and escalating behavior in lowest unique bid auctions. PloS one 7(1):e29910
36. Radicchi F, Baronchelli A (2012) Evolution of optimal Lévy-flight strategies in human mental searches. Phys Rev E 85(6):061121
37. Yang X-S (2010) Engineering optimization: An introduction with metaheuristic applications. Wiley
38. Yang X-S, Deb S (2010) Engineering optimisation by cuckoo search. Int J Math Modell Numer Optim 1(4):330–343
39. Hartigan JA, Wong MA (1979) Algorithm AS 136: A k-means clustering algorithm. J Royal Stat Soc Ser C (Appl Stat) 28(1):100–108
40. Digalakis JG, Margaritis KG (2001) On benchmarking functions for genetic algorithms. Int J Comput Math 77(4):481–506
41. Molga M, Smutnicki C (2005) Test functions for optimization needs. Test functions for optimization needs
42. Yang XS (2010) Appendix A: test problems in optimization. Engineering optimization:261–266
43. Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Glob Optim 39(3):459–471
44. Suganthan PN, Hansen N, Liang JJ, Deb K, Chen Y-P, Auger A, Tiwari S (2005) Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. KanGAL Report 2005005:2005
45. Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput 1(1):3–18
46. Merrikh-Bayat F (2015) The runner-root algorithm: A metaheuristic for solving unimodal and multimodal optimization problems inspired by runners and roots of plants in nature. Appl Soft Comput 33:292–303
47. Holm S (1979) A simple sequentially rejective multiple test procedure. Scandinavian Journal of Statistics:65–70
48. Liu S-H, Mernik M, HrnčIč D, Črepinšek M (2013) A parameter control method of evolutionary algorithms using exploration and exploitation measures with a practical application for fitting Sovova's mass transfer model. Appl Soft Comput 13(9):3792–3805
49. Coello CAC, Montes EM (2002) Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. Adv Eng Inf 16(3):193–203
50. Coello CAC (2002) Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. Comput Methods Appl Mech Eng 191(11):1245–1287
51. Deb K (1997) GeneAS: A robust optimal design technique for mechanical component design. In: Evolutionary Algorithms in Engineering Applications. Springer, pp 497–514
52. He Q, Wang L (2007) An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Eng Appl Artif Intell 20(1):89–99
53. Huang F-Z, Wang L, He Q (2007) An effective co-evolutionary differential evolution for constrained optimization. Appl Math Comput 186(1):340–356
54. Mezura-Montes E, Coello CAC (2008) An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. Int J Gen Syst 37(4):443–473
55. Kaveh A, Talatahari S (2010) A novel heuristic optimization method: charged system search. Acta Mech 213(3–4):267–289
56. Kannan B, Kramer SN (1994) An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. J Mech Des 116(2):405–411
57. Sandgren E (1990) Nonlinear integer and discrete programming in mechanical design optimization. J Mech Des 112(2):223–229
58. Coello Coello CA (2000) Constraint-handling using an evolutionary multiobjective optimization technique. Civ Eng Syst 17(4):319–346
59. Deb K (2000) An efficient constraint handling method for genetic algorithms. Comput Methods Appl Mech Eng 186(2):311–338
60. Lee KS, Geem ZW (2005) A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Comput Methods Appl Mech Eng 194(36):3902–3933
61. Krohling RA, Dos Santos Coelho L (2006) Coevolutionary particle swarm optimization using Gaussian distribution for solving constrained optimization problems. IEEE Trans Syst, Man, Cybern, Part B (Cybern) 36(6):1407–1416
62. Ragsdell K, Phillips D (1976) Optimal design of a class of welded structures using geometric programming. J Eng Ind 98(3):1021–1025
63. Zhang M, Luo W, Wang X (2008) Differential evolution with dynamic stochastic selection for constrained optimization. Inf Sci 178(15):3043–3074
64. Mirjalili S (2015) Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. Knowl-Based Syst 89:228–249
65. Liu H, Cai Z, Wang Y (2010) Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. Appl Soft Comput 10(2):629–640
66. RAY T, SAINI P (2001) Engineering design optimization using a swarm with an intelligent information sharing among individuals. Eng Optim 33(6):735–748
67. Gandomi AH, Yang X-S, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng Comput 29(1):17–35
68. Tsai J-F (2005) Global optimization of nonlinear fractional programming problems in engineering design. Eng Optim 37(4):399–409

Seyed Jalaleddin Mousavirad recevied the B.S. degree in Computer Engineering from Islamic Azad University, Mashhad branch, Iran, in 2009 and M.S. degree in Computer Engineering from Kurdistan University, Sanandaj, Iran, in 2012. He is currently persuing the Ph.D. degree in Computer Engineering at Kashan University, Kashan, Iran. He worked as a visiting student at Xi'an Jiaotong Liverpool University, Suzhou, China under the supervision of Professor Yuhui Shi. His research interests include evolutionary computation, multiobjective optimization, and evolutionary computation applications in other scientific fields such as image processing and pattern recognition.

Hossein Ebrahimpour-Komleh is currently an Assistant Professor at the Department of Electrical and Computer Engineering at the University of Kashan, Kashan, Iran. His main area of research includes Computer vision, Image Processing, Pattern Recognition, Biometrics, Robotics, Fractals, chaos theory and applications of Artificial Intelligence in Engineering. He received his Ph.D. degree in Computer engineering from Queensland University of technology, Brisbane, Australia in 2006. His Ph.D. research work was on the "Fractal Techniques for face recognition". From 2005 to 2007 and prior to joining the University of Kashan, he was working as a Post-doc researcher in the University of Newcastle, NSW, Australia and as a visiting scientist in CSRIO Sydney. Hossein Ebrahimpour-Komleh has B.Sc. and M.Sc. degrees both in computer engineering from Isfahan University of Technology (Isfahan, Iran) and Amirkabir University of Technology (Tehran, Iran), respectively. He has served as the editorial board member and reviewer of several journals and international and national conferences.