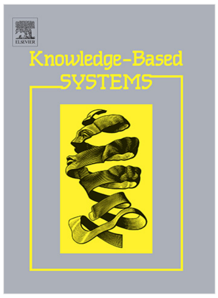
Journal Pre-proof



Learning-based elephant herding optimization algorithm for solving numerical optimization problems

Wei Li, Gai-Ge Wang, Amir H. Alavi

PII: S0950-7051(20)30117-9
DOI: https://doi.org/10.1016/j.knosys.2020.105675
Reference: KNOSYS 105675

To appear in: *Knowledge-Based Systems*

Received date : 2 July 2019
Revised date : 13 February 2020
Accepted date : 18 February 2020

Please cite this article as: W. Li, G.-G. Wang and A.H. Alavi, Learning-based elephant herding optimization algorithm for solving numerical optimization problems, *Knowledge-Based Systems* (2020), doi: https://doi.org/10.1016/j.knosys.2020.105675.

*Revised Manuscript (Clean Version)
Click here to view linked References

TITLE PAGE

Title

Learning-based elephant herding optimization algorithm for solving numerical optimization problems

1st Author

Wei Li

Department of Computer Science and Technology, Ocean University of China, 266100 Qingdao, China

E-mail: li070113wei@gmail.com

2nd & Corresponding Author

Gai-Ge Wang*

Department of Computer Science and Technology, Ocean University of China, 266100 Qingdao, China

Institute of Algorithm and Big Data Analysis, Northeast Normal University, Changchun, 130117, China

School of Computer Science and Information Technology, Northeast Normal University, Changchun, 130117, China

Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University,

Changchun, 130012, China

E-mail: gaigewang@gmail.com; gaigewang@163.com, wgg@ouc.edu.cn

3rd Author

Amir H. Alavi

Department of Civil and Environmental Engineering, University of Pittsburgh, Pittsburgh, PA 15261, USA

Department of Bioengineering, University of Pittsburgh, Pittsburgh, PA 15261, USA

Department of Computer Science and Information Engineering, Asia University, Taichung, Taiwan

E-mail: alavi@pitt.edu, a.h.alavi.iust@gmail.com

Number of Words: 20795

Number of Tables: 48

Number of Figures: 7

Number of Appendices: 1

Number of Supplementary Material: 0

Learning-based elephant herding optimization algorithm for solving numerical optimization problems

Abstract

The elephant herding optimization (EHO) is a recent swarm intelligence algorithm. This algorithm simulates the clan updating and separation behavior of elephants. The EHO method has been successfully deployed in various fields. However, a more reliable implementation of the standard EHO algorithm still requires improving the control and selection of the parameters, convergence speed, and efficiency of the optimal solutions. To cope with these issues, this study presents an improved EHO algorithm terms as IMEHO. The proposed IMEHO method uses a global velocity strategy and a novel learning strategy to update the velocity and position of the individuals. Furthermore, a new separation method is presented to keep the diversity of the population. An elitism strategy is also adopted to ensure that the fittest individuals are retained at the next generation. The influence of the parameters and strategies on the IMEHO algorithm is fully studied. The proposed method is tested on 30 benchmark functions from IEEE CEC 2014. The obtained results are compared with other eight metaheuristic algorithms and evaluated according to Friedman rank test. The results imply the superiority of the IMEHO algorithm to the standard EHO and other existing metaheuristic algorithms.

Key words: elephant herding optimization; swarm intelligence; velocity strategy; learning strategy; separation strategy; elitism strategy; benchmark function

1. Introduction

Single objective optimization is the foundation of complex optimization procedures such as multi-objective optimization and constraint optimization. As the real-world optimization problems become more complex, the traditional optimization algorithms become more uncertain for solving these problems. Metaheuristic algorithms are proved to be a solution for this challenge. Among various metaheuristic algorithms, swarm intelligence (SI) algorithms represent a class of efficient metaheuristics [1]. The concept of SI inspired by the movement of insect swarms in nature was first proposed by Gerardo and Wang [2]. Two of the most widely-used SI algorithms are particle swarm optimization (PSO) [3] and ant colony optimization (ACO) [4]. The PSO algorithm was proposed by Eberhart et al. [3] in 1995. This algorithm simulates the clustering behavior of insects, beasts, birds, and fishes. These groups find food in a cooperative way, and each member of the group continually changes its search pattern by learning from its own experience and the experience of other members. Many researches have improved the reliable performance of the PSO algorithm [5-7]. The ACO algorithm was first proposed by Dorigo et al. [4] after studying the foraging of ants. Dorigo et al. [4] found that while the behavior of ant individual is relatively simple, the whole ant colony offer an intelligent behavior. Later, Yang et al. [5] proposed an adaptive multi-mode ACO, which was extended to multi-mode optimization. Various SI algorithms have been proposed by researchers such as artificial bee colony (ABC) [6], cuckoo search (CS) [7], bat algorithm (BA) [8], firefly algorithm (FA) [9], grey wolf optimizer (GWO) [10], wolf search algorithm (WSA) [11], ant lion optimizer (ALO) [12], charged system search (CSS) [13], monarch butterfly optimization (MBO) [14], krill herd (KH) [15], earthworm optimization algorithm (EWA) [16], locust search algorithm (LS) [17], moth search algorithm (MSA) [18], multi-verse optimizer (MVO) [19], dragonfly algorithm (DA) [20], and elephant herding optimization (EHO) [21,22]. Extensive research has also been done on the implementation of these methods. For example, to solve scheduling problems, Narimani et al. [23] proposed a hybrid evolutionary algorithm based on the shuffle frog leaping algorithm and the PSO to solve multi-area economic emission dispatch problem. Sang et al. proposed a series of the SI algorithms to solve the job-shop scheduling problems [24-26]. Dynamic optimization with the SI methods has been the focus of some studies [27-29]. For test-sheet composition problems, Duan et al. [30] proposed an improved biogeography-based optimization (TS/BBO) algorithm. The experiment illustrated that the proposed TS/BBO can effectively improve composition speed and success rate [30]. In order to solve path planning for uninhabited combat aerial vehicle, Wang et al. [31] proposed a hybrid meta-heuristic DE/BBO algorithm. Yi et al. [32] improved the NSGA-III algorithm for big data [32]and large-scale optimization problems. Yi et al. [33] proposed a variant of probabilistic neural network with self-adaptive strategy for transformer fault diagnosis problems. The results proved that self-adaptive probabilistic neural networks have a more accurate prediction and better generalization performance for transformer fault diagnosis problem [33]. Naderi et al. [34] proposed an efficient PSO algorithm to solve optimal power flow problem integrated with FACTS devices. In the area of multi-objective feature selection problems, Zhang et al. [35] proposed a binary differential evolution with self-learning. For the cost-sensitive feature selection problems, Zhang et al. [36] proposed a two-archive multi-objective ABC algorithm. In addition, inspired by nature, these strong metaheuristic algorithms are applied to solve NP-hard problems such as scheduling [37-39], image [40-42], feature selection [43-45] and detection [46], path planning [47], cyber-physical social system [48], texture discrimination [49], saliency detection [50], classification [51,52], object extraction [53], economic load dispatch [54], global numerical optimization [55-59], multi-objective optimization [60-62], knapsack problem [63,64], and fault diagnosis [65,66].

Among the SI optimization algorithms, the EHO algorithm is a fairly new method [21]. In this algorithm, the herding behavior of the elephants is defined by two operators: clan updating operator and separating operator [21]. However, some of the major limitations of the EHO algorithm are as follows: (a) the position variable is adopted and the moving speed of the elephant is ignored, (b) the updating operator is fixed and limited to the clan, (c) the worst member in separation operator is replaced randomly. While this decentralized topology and individual updating methods help the algorithm not to fall into local optimization easily, they result in the lack of the ability to perform the exploitation efficiently. For these reasons, the EHO algorithm is considered more suitable for solving multimodal problems compared to unimodal problems. In order to improve the shortcomings of the standard EHO algorithm, this paper proposes an improved EHO algorithm called IMEHO. We first present a new velocity strategy that assigns each elephant a moving speed based on its current location. Then, the operators in the EHO algorithm are improved via a new learning strategy for clan updating operator and a new separation strategy for separating operator. We also adopt the elitism strategy to ensure the herd is getting better. In the IMEHO algorithm, each elephant implements updating of its velocity and position based on velocity strategy and learning strategy. Subsequently, the worst elephant is replaced by separation strategy. The performance of the algorithm is compared with the standard EHO [21], PSO [3], BA [8], evolution strategy (ES) [67], genetic algorithm (GA) [68], population-based incremental learning (PBIL) [69], chaotic cuckoo search (CCS) [70], and variable neighborhood bat (VNBA) [71] algorithms on 30 benchmark functions. In conclusion, the contributions of this paper are as follows:

1. We improved the standard EHO algorithm.
2. We introduced a global velocity strategy, which is implemented by assigning velocity to each individual.
3. We improved the updating operator and proposed a novel learning strategy that updates the velocity and position of individuals through three different scenarios.
4. We improved the separating operator and proposed a new separation strategy to improve the performance of the algorithm and keep the diversity of the population.
5. We adopt the elitism strategy to ensure the herd is getting better.
6. We tested the proposed IMEHO on 30 benchmark functions from IEEE CEC 2014 and got good results.

The rest of the paper is arranged as follows: Section 2 introduces the related work. Section 3 reviews the EHO algorithm. Section 4 introduces the IMEHO in detail. Section 5 is the comparison experiments. Finally, Section 6 draws the conclusions.

2. Related work

The current work is based on EHO and learning evolutionary algorithms. In this section, the development history of EHO is reviewed. The idea behind the learning evolutionary algorithms is also discussed.

2.1 Overview of the EHO algorithm development

The EHO algorithm was first proposed by Wang et al. [21]. Later, Wang et al. [22] improved the original EHO algorithm. It was shown that EHO performs well with less parameter control and a simpler process compared to other SI methods. Many researchers have introduced different strategies in order to improve the performance of the EHO algorithm. Gupta et al. [72] improved EHO algorithm for level control of three tank system. They used the algorithm to minimize the integral-square-error. The results showed that EHO has better performance than other methods [72]. Strumberger et al. [73] proposed a method for static drone placement using the EHO algorithm. Alihodzic et al. [74] proposed a path planning problem for drones using EHO. They adapted the EHO algorithm to path planning problem. Similarly, Meena et al. [75] proposed an improved multi-objective EHO algorithm for distribution systems. They improved the updating and separating components in the elephant group and solved a complex real-life multi-objective programming problem to maximize the overall benefit of utilities and consumers. Elhosseini et al. [76] studied the influence of the control parameters on the EHO algorithm. A more detailed parameter study of EHO algorithm was carried out using standard test bed, engineering problems and practical problems [76]. Guo et al. [77] proposed several new update strategies for EHO.

Despite these improvements, the EHO algorithm still needs improvements in the control and selection of the parameters, convergence speed and the efficiency in obtaining the optimal solution. It is worth mentioning that the EHO algorithm has two core ideas: clan updating operator and separation operator. The performance of EHO algorithm heavily relies on choice of a strategy for these cores, which is the focus of this study.

2.2 Learning evolutionary algorithms

Learning evolutionary algorithm refers to the algorithm that endows evolutionary algorithms with certain learning ability to generate offspring through continuous learning in the evolution process. This algorithm is not the same as the traditional evolutionary algorithms. It learns to change its evolutionary direction during the optimization process. The learning evolutionary algorithms have achieved good performance and are widely used in numerous fields. Chen et al. [78] proposed a PSO algorithm with incremental learning which could overcome local convergence and obtained the optimal solution at the same time. Ye et al. [79] proposed a multi-particle swarm optimization algorithm based on dynamic learning strategy. The performance of the algorithm was improved by dividing the particles into ordinary particles and communication particles with different tasks. Gao et al. [80] proposed an artificial swarm algorithm based on information learning. Anait et al. [81] proposed a learning-based ant colony clustering method. Dai et al. [82] proposed a multi-objective orthogonal evolutionary algorithm with learning automata. Zhu et al. [83] proposed a learning enhanced differential evolutionary algorithm for tracking optimal decisions in dynamic power systems. Ahmadi et al. [84] proposed GA based on dynamic Q-learning. Tomas et al. [85] combined an orthogonal learning method with the FA. Rakshit et al. [86] proposed a new ABC algorithm integrated with a random learning strategy to realize the self-adaptation of sample size. Wang et al. [87] proposed a hybrid PSO algorithm which employed an adaptive learning strategy.

In this paper, a learning strategy is proposed that is suitable for the EHO method. We also improve the two core ideas of the EHO algorithm by introducing velocity strategy, separation strategy, and applying elitism strategy to the whole algorithm. We first simulate the movement of elephants and assign a velocity to each elephant. Then, at each generation of the clan updating operator, we update all the elephants' information according to three different scenarios. In the separation operator, we take the first evaluation before performing the replace operation. A selection probability is introduced to decide whether to perform the replace operation or not. Finally, the elitism strategy is applied to assure that the population is better than the original.

3. Elephant herding optimization algorithm

In general, elephants live in groups. An elephant herd is made up of several different clans. Different elephants in the same clan live together under the leadership of a matriarch. There are only adult females and calves in the herd. An elephant does not mate with members of its own family, so as adults, the male elephants will leave the herd to find mates. Inspired by the elephant herd behavior, the EHO algorithm was proposed to solve the optimization problems. In EHO, the behavior of the elephant group is idealized into two parts: clan updating operator and separating operator. In the clan updating process, each elephant carries on the clan updating operator by its current position and the matriarch's position in the herd. As a result, for the elephant j in clan ci, the position can be updated as:

$$x_{new,ci,j} = x_{ci,j} + \alpha \times (x_{best,ci} - x_{ci,j}) \times r \,, \tag{1}$$

where $x_{new,ci,j}$ and $x_{ci,j}$ are new and old position for elephant j in clan ci, respectively. $\alpha \in [0, 1]$ is a scale factor. $x_{best,ci}$ represents the fittest position in clan ci. $r \in [0, 1]$ is a normally distributed random number.

The matriarch's position has not been updated in Eq. (1). For the fittest one, it can be updated as:

$$x_{new,ci,j} = \beta \times x_{center,ci} \,, \tag{2}$$

$$x_{center,ci} = \frac{1}{n_{ci}} \times \sum_{j=1}^{n_{ci}} x_{ci,j} \,, \tag{3}$$

where $\beta \in [0, 1]$ is a scale factor. $x_{center,ci}$ is the center position in clan ci. n_{ci} is the number of elephants in clan ci. It can be seen that the matriarch's position is updated by all the members' position in clan.

In the separation process, the male elephants leave the herd and live alone. In the EHO algorithm, it was assumed that the worst elephant in each clan is replaced by a separation operator. The process can be show in Eq. (4).

$$x_{worst,ci} = x_{\min} + (x_{\max} - x_{\min} + 1) \times r \,, \tag{4}$$

where $x_{worst,ci}$ represents the worst elephant in clan ci. The upper and lower bound of the elephant position are $x_{\max}$ and $x_{\min}$, respectively. $r \in [0, 1]$ is a normally distributed random number.

According to the description above, the EHO algorithm can be represented by Algorithm 1.

Algorithm 1: The EHO algorithm

Begin

Step 1: Initialization. Initialize the population and parameters.

Step 2: Fitness evaluation. Evaluate each elephant individual according to its position.

Step 3: While $t < T_{\max}$ (the number of maximum generations) **do**

For $c_i = 1$ to n_{ci} (the number of all clans) **do**
For $j = 1$ to n_j (the number of the elephants in one clan) **do**
Update $x_{ci,j}$ and generate $x_{new,ci,j}$ according to Eq. (1)
If $x_{ci,j} = x_{pbest,ci}$ **then**
Update $x_{ci,j}$ and generate $x_{new,ci,j}$ according to Eqs. (2) - (3)
End if
End for j
End for c_i
For $c_i = 1$ to n_{ci} (the number of all clans) **do**
Replace the worst elephant individual in clan c_i by Eq. (4).
End for c_i
End while
Step 4: Output the best solution.
End.

4. Improved Elephant Herding Optimization Algorithm

The standard EHO algorithm updates the position of each elephant using clan updating and separating operators. Three drawbacks of the standard EHO methods are as follows:

(1) The position variable is adopted, and the moving speed of the elephant is ignored. These issues affect the convergence speed of the algorithm.

(2) In the process of clan updating, for ordinary elephants, their positions are updated according to their own positions and the positions of their matriarch. The position of matriarch is updated by the middle positions of the clan. It is easy to form into local optimum for one clan of an elephant group. In elephant groups, the matriarch should lead others to explore more suitable habitats.

(3) In the separation process, the worst elephant in each clan is replaced by Eq. (4). The process simulates the departure of an adult male elephant and the birth of a calf in the herd. As a young elephant, it will be protected by other elephants, and it will have a good position. Therefore, its position should not be evaluated worse than the original.

In order to address these shortcomings, we proposed four strategies from four different perspectives. From a bionics perspective, we simulate the movement of the elephants and propose a global velocity strategy. From the learning perspective, we propose a novel learning strategy to better exchange information between elephants. From an algorithmic perspective, we improve the separation operator and propose a new separation strategy so that the algorithm will have better exploitation ability. Finally, from an evolutionary perspective, we adopt an elitism strategy to keep the elephant herd getting better.

4.1 Velocity Strategy

In IMEHO algorithm, we assign each elephant a set of velocity v_j when initializing its position. The process of initializing the position and speed of elephant can be given as follows:

$$x_j = x_{\min} + (x_{\max} - x_{\min}) \times r , \tag{5}$$

$$v_j = v_{\min} + (v_{\max} - v_{\min}) \times r , \tag{6}$$

where $v_{\max} = (x_{\max} - x_{\min}) * 0.2$, $v_{\min} = -v_{\max}$, x_j is the position of the elephant j, v_j is the speed of the elephant j. $x_{\max}$ and $x_{\min}$ are the upper and lower bound of the position, $v_{\max}$ and $v_{\min}$ are the upper and lower bound of the velocity, respectively. r is a normally distributed random number generated in the range [0, 1].

At each generation, the elephant's speed is updated by Eq. (7) as follows:

$$v_{new,ci,j} = w_i \times v_{ci,j} + c \times (x^* - x_{ci,j}) \times r, \tag{7}$$

where $v_{new,ci,j}$ and $v_{ci,j}$ are new and old speed for elephant j in clan ci. w_i is inertia weight at the ith generations and it's decreases linearly. c is the acceleration coefficient. $x_{ci,j}$ is the position of elephant j in clan ci. x^* is the learning goal of elephant $x_{ci,j}$. r is a normally distributed random number generated in the range [0, 1].

4.2 Learning Strategy

In EHO, each elephant carries on the clan updating operator by its current position and the matriarch's position in the herd. As the matriarch's position is updated by all members' information in the clan, it is easy to fall into local optimum. Based on the proposed new learning strategy, all the elephants are divided into three groups. The first kind of elephants is x_{gbest} which is the best one in the elephant herd. The second is the matriarchs of each clan called x_{pbest}. The last is the remaining elephants which is called x_{other}.

The best elephant x_{gbest} in the herd should explore outside the clan and learn from other matriarchs. Here, we assume that it updates itself according to the information of all the matriarchs. For the elephant x_{gbest} in the herd, the position can be updated as:

$$v_{new,gbest} = w_i \times v_{gbest} + \alpha \times (x_{center} - x_{gbest}), \tag{8}$$

$$x_{new,gbest} = x_{gbest} + v_{new,gbest}, \tag{9}$$

$$x_{center} = \frac{1}{n_{ci}} \times \sum_{i=1}^{n_{ci}} x_{pbest,ci}, \tag{10}$$

where $v_{new,gbest}$ and v_{gbest} are new and old speed for elephant x_{gbest}. $x_{new,gbest}$ is the new position for elephant x_{gbest}. w_i is inertia weight at the ith generation. $\alpha \in (0, 1]$ is an impact factor. x_{center} is the middle position of matriarchs and calculated by Eq. (10). n_{ci} is the number of clans.

The second kind of elephants x_{pbest} indicates the position of the matriarchs. It should lead its members to learn from the best elephant in the herd. Here, we assume that it updates itself according to the information of the elephant x_{gbest}:

$$v_{new,pbest} = w_i \times v_{pbest} + c \times (x_{gbest} - x_{pbest}) \times r, \tag{11}$$

$$x_{new,pbest} = x_{pbest} + v_{new,pbest}, \tag{12}$$

where $v_{new,pbest}$ and v_{pbest} are new and old speed for elephant x_{pbest}. c is the acceleration coefficient. $x_{new,pbest}$ is the new position for elephant x_{pbest}. r is a normally distributed random number generated in the range [0, 1].

Other elephants x_{other} will learn from their matriarch. Here, we assume that they update themselves according to their matriarch in the same clan as follows:

$$v_{new,other} = w_i \times v_{other} + c \times (x_{pbest} - x_{other}) \times r \, , \tag{13}$$

$$x_{new,other} = x_{other} + v_{new,other} \, , \tag{14}$$

where $v_{new,other}$ and v_{other} are new and old speed for elephant x_{other}, respectively. $x_{new,other}$ is new position for elephant x_{other}. r is a normally distributed random number generated in the range [0, 1].

According to the description above, the learning strategy is mainly used in the clan updating operator. A better understanding of the learning process can be obtained from Figure 1. Referring to this figure, suppose an elephant herd includes two clans, where elephant 1 is the *gbest* of the elephant herd and elephant 4 is the *pbest* of the clan 2. The specific learning process is as follows: elephant 1 learns from the information about itself and Elephant 4. Elephant 4 learn from elephant 1. Elephants 2, 3 all learn from Elephant 1. Elephants 5, 6 all learn from Elephant 4. More details of the clan updating operator is shown in Algorithm 2.

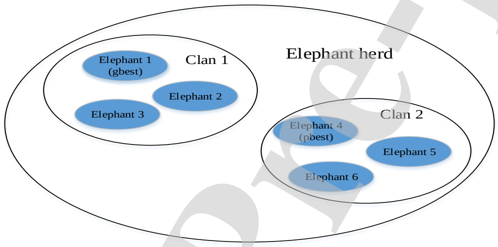


Figure 1 An example of the learning strategy

4.3 Separation Strategy

In EHO, the worst elephant in each clan will be replaced by Eq. (4). Using the separation operator in the original EHO algorithm, the evaluation of the newborn calf is ignored. To ensure the diversity of the population, we put forward a new separation strategy to improve the original algorithm. In our strategy, the newborn calf will be generated by Eqs. (5) - (6). The newborn calf will be evaluated firstly. If the fitness value is better than the original one, it can be replaced. Otherwise, there will be a probability value *pc* which will decide whether to replace the original elephant or not. Here, a judgment operation will be carried out to randomly generate a random number r. If its value is greater than *pc*, the original elephant will be replaced. Here, a small probability event is used for selection individual. It improves the performance of the algorithm and keeps the diversity of the population. The details of the separating operator can be shown in Algorithm 3. Figure 2 shows an example of the separating operator. Suppose elephant 2 is the worst individual in clan 1 and needs to be replaced by the separating operator. Then there will be 3 kinds of results, assuming they are elephants 4, 5, and 2, respectively. Elephant 4 indicates that an elephant who is superior to elephant 2 is directly replaced. Although elephant 5 indicates that the evaluation is not as good as elephant 2, the condition of probability pc is met, and this situation is also replaced. Elephant 2 indicates that no replacement occurs. That is, the individual evaluation produced is not as good as Elephant 2, and the probability pc condition is not met.

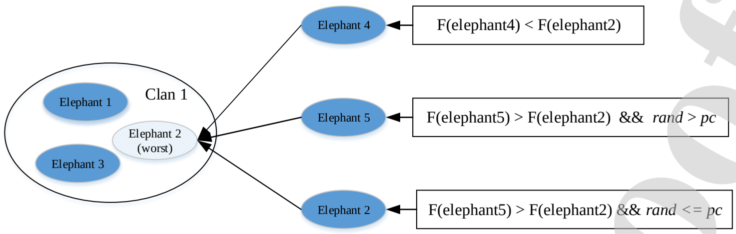


Figure 2 An example of the separation strategy

4.4 Elitism Strategy

The elitism strategy enables the fittest individuals to be retained at the next generation. The basic idea is that the individual with the highest fitness in the current population does not participate in the operation of the algorithm. However, it is used to replace the individuals with the lowest fitness in population at the next generation. The implementation of the elitism strategy can ensure that the optimal individuals will not be destroyed by other operations. That is why it is an important guarantee that the population will get better and better. In the IMEHO algorithm, we select k best individuals and save their information at each generation. At the end of each generation, the worst k individuals in the population are replaced by the k saved individuals. The elitism strategy is performed at each generation.

4.5 Comparison between different strategies

To clarify the difference between the four strategies, we give a simple example. We assume that: f(x) = x^2 is the objective function, the population size is 4, and initialize their position to x = (4, 1, -2, -4), $x \in$ [-4, 4]. We initialize the velocity to v = (2, 1, -1, -2), $v \in$ [-2, 2]. We set all parameters to 1, including random numbers. At the same time, we assume that the first two individuals are in clan 1, and the last two individuals are in clan 2. In addition, we use S_1, S_2, S_3, S_4 to represent velocity strategy, learning strategy, separation strategy, and elitism strategy, respectively. $Step_1$, $Step_2$, $Step_3$, $Step_4$ are used to represent initialization, clan updating operator, separating operator, and elitism strategy process, respectively. Conclusions are then drawn in Table 1. As shown in Table 1, all methods have the same results after $Step_1$. In $Step_2$, because EHO-S_1 increases the speed variable and EHO-S_2 uses a learning strategy, the results after updating are different. In $Step_3$, all the methods perform the separation operation, and only EHO-S_3 is not replaced. This is because the newly generated solution does not meet the replacement conditions. In $Step_4$, only EHO-S_4 replaced the worst individual with the best individual in the previous generation. As can be seen in Table 1, each strategy has a different role. The best results are obtained using the proposed learning strategy.

Table 1 A comparison of different strategies

		EHO	EHO-S_1	EHO-S_2	EHO-S_3	EHO-S_4	IMEHO
	x	(4, 1, -2, -4)	(4, 1, -2, -4)	(4, 1, -2, -4)	(4, 1, -2, -4)	(4, 1, -2, -4)	(4, 1, -2, -4)
$Step_1$	v		(2, 1, -1, -2,)				(2, 1, -1, -2,)
	f(x)	(16, 1, 4, 16)	(16,1,4,16)	(16,1,4,16)	(16,1,4,16)	(16,1,4,16)	(16, 1, 4, 16)
$Step_2$	x	(1, 2.5, -3, -2)	**(-3, 3, -4, -4)**	**(1, 0.5, 1, -2)**	(1, 2.5, -3, -2)	(1, 2.5, -3, -2)	**(3, 0.5, -1, -4)**

	v		**(-1, 2, -2, 0)**				**(-1, -0.5, 1, 0)**
	f(x)	(1, 6.25, 9, 4)	**(9, 9, 16, 16)**	**(1, 0.25, 1, 4)**	(1, 6.25, 9, 4)	(1, 6.25, 9, 4)	**(9, 0.25, 1, 16)**
	x	(1, 2.5, 4, -2)	(-3, 3, -4, 4)	(1, 0.5, 1, 4)	**(1, 2.5, -3, -2)**	(1, 2.5, 4, -2)	(3, 0.5, -1, 4)
Step$_3$	v		(-1, 2, -2, 0)				(-1, -0.5, 1, 0)
	f(x)	(1, 6.25, 16, 4)	(9, 9, 16, 16)	(1, 0.25, 1, 16)	**(1, 6.25, 9, 4)**	(1, 6.25, 16, 4)	(9, 0.25, 1, 16)
	x	(1, 2.5, 4, -2)	(-3, 3, -4, 4)	(1, 0.5, 1, 4)	(1, 2.5, -3, -2)	**(1, 2.5, 1, -2)**	**(3, 0.5, -1, 1)**
Step$_4$	v		(-1, 2, -2, 0)				**(-1, -0.5, 1, 0)**
	f(x)	(1, 6.25, 16, 4)	(9, 9, 16, 16)	(1, 0.25, 1, 16)	(1, 6.25, 9, 4)	**(1, 6.25, 1, 4)**	**(9, 0.25, 1, 1)**
f(x*)		1	9	0.25	1	1	0.25

Algorithm 2: Clan updating operator

Begin
  For c_i = 1 to n_{ci} (the number of all clans) **do**
    For j = 1 to n_j (the number of the elephants in one clan) **do**
      If $x_{ci} = x_{gbest}$ (the best one in the herd) **then**
        Update $x_{ci,j}$, $v_{ci,j}$, and then generate $x_{new,ci,j}$ according to Eqs. (8) - (10).
      End if
      If $x_{ci} = x_{pbest}$ (the best one in the clan) **then**
        Update $x_{ci,j}$, $v_{ci,j}$, and then generate $x_{new,ci,j}$ according to Eqs. (11) - (12).
      End if
      If $x_{ci} = x_{other}$ (other ordinary elephants in the herd) **then**
        Update $x_{ci,j}$, $v_{ci,j}$, and then generate $x_{new,ci,j}$ according to Eqs. (13) - (14).
      End if
    End for j
  End for c_i
End.

Algorithm 3: The separating operator

Begin
  For c_i = 1 to n_{ci} (the number of all clans) **do**
    Generate a new elephant individual x_{new} by Eqs. (5) - (6) and evaluate it.
    If $x_{new} < x_{worst}$ **then**
      Replace the worst elephant individual by x_{new}.
    Else if $pc < rand$ **then**
      Replace the worst elephant individual by x_{new}.
    End if
  End for c_i
End.

4.6 The proposed IMEHO algorithm

The IMEHO algorithm starts with initializing the population and parameters. This process includes assigning a velocity to each individual by velocity strategy. Then, all individuals are evaluated based on their position and velocity information, and ranked by their fitness value. Then, elitism strategy is applied to save a certain number of optimal individuals. All individuals are updated by the learning and separation strategies. Finally, all individuals are evaluated and ranked according to their new position and velocity. The elitism strategy is used to replace the poorer individuals. The above process is repeated until the loop is completed. The IMEHO algorithm optimization process is shown in Algorithm 4. The IMEHO flow chart is presented in Figure 3. In Figure 3, $x_{ci,j}$ is the position of elephant j in clan ci. x_{gbest} is the position of best elephant in the herd. x_{pbest} is the position of matriarch. n_{clan} is the number of elephants in each clan. x_{new} is the position of the new individual. $x_{worst,ci}$ is the worst elephant in clan ci. *NumClan* is the number of clans in the herd. *pc* is the probability value. *r* is a normally distributed random number in the range [0, 1]. $T_{\max}$ and $F_{\max}$ are the maximum number of generations and evaluations, respectively.

Algorithm 4: The IMEHO algorithm

Begin

Step 1: Initialization. Initialize the population and parameters;

Step 2: Fitness evaluation. Evaluate all elephants according to their position and velocity;

Step 3: While $t < T_{\max}$ **do**

Sort all the elephants according to their fitness;

Save the first *k* elephants' information;

Implement Clan updating operator by Algorithm 2;

Implement Separating operator by Algorithm 3;

Evaluate according to position and velocity;

Sort all the elephants according to their newly fitness;

Replace the worst elephants by *k* saved elephants' information;

Update the generation, $t = t + 1$;

End while

Step 4: Output the best solution;

End.

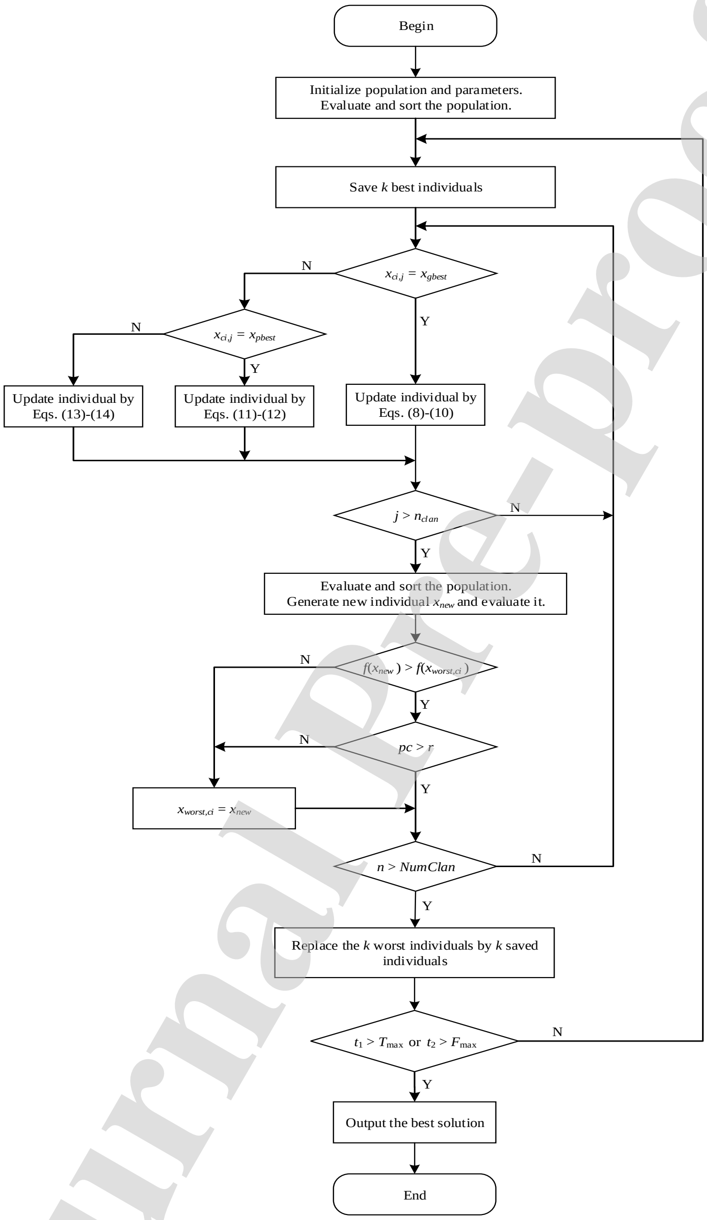


Figure 3 Flow chart of the IMEHO algorithm

5. Experiments

The IMEHO algorithm is evaluated on 30 single-objective benchmark functions from IEEE CEC 2014 [88]. These benchmark functions are divided into four groups: (1) unimodal functions (F01-F03); (2) simple multimodal functions (F04-F16); (3) hybrid functions (F17-F22); (4) composition functions (F23-F30) [89]. More details about the benchmark functions can be shown in Table 2. Table 3 shows the results of a comparative study conducted between the IMEHO and BA, EHO, ES, GA, PBIL, PSO, CCS, and VNBA algorithms.

Table 2 Summary of the CEC 2014 Test Functions

No.	Functions	$f(x^*)$	NO.	Functions	$f(x^*)$
F01	Rotated High Conditioned Elliptic Function	100	F16	Shifted and Rotated Expanded Scaffer's F6 Function	1600
F02	Rotated Bent Cigar Function	200	F17	Hybrid Function 2	1700
F03	Rotated Discus Function	300	F18	Hybrid Function 2	1800
F04	Shifted and Rotated Rosenbrock's Function	400	F19	Hybrid Function 3	1900
F05	Shifted and Rotated Ackley's Function	500	F20	Hybrid Function 4	2000
F06	hifted and Rotated Weierstrass Function	600	F21	Hybrid Function 5	2100
F07	Shifted and Rotated Griewank's Function	700	F22	Hybrid Function 6	2200
F08	Shifted Rastrigin's Function	800	F23	Composition Function 1	2300
F09	Shifted and Rotated Rastrigin's Function	900	F24	Composition Function 2	2400
F10	Shifted Schwefel's Function	1000	F25	Composition Function 3	2500
F11	Shifted and Rotated Schwefel's Function	1100	F26	Composition Function 4	2600
F12	Shifted and Rotated Katsuura Function	1200	F27	Composition Function 5	2700
F13	Shifted and Rotated HappyCat Function	1300	F28	Composition Function 6	2800
F14	Shifted and Rotated HGBat Function	1400	F29	Composition Function 7	2900
F15	Shifted and Rotated Expanded Griewank's plus Rosenbrock's Function	1500	F30	Composition Function 8	3000

Search Range: $[-100, 100]^D$

Table 3 A comparison of the IMEHO algorithms and existing optimization methods

	Advantages	Weaknesses
BA	• Fast solution. • Process is simple.	• Easy to fall in local optimality. • Slow convergence.
EHO	• Fewer number of control parameters. • No complicated operators.	• Randomly replace the worst member. • Parameters are fixed during all generations.
ES	• Easy to implement.	• Slow convergence

	• Few parameters	• Easy to fall in local optimality.
GA	• Search quickly and randomly. • Easy to be combined with other algorithms.	• Not easy to implement. • No feedback and searches slowly.
IMEHO	• Fast convergence. • High solution accuracy.	• Not easy to implement.
PBIL	• The operation is simple. • Fast and accurate problem solving	• Slow convergence. • Poor search ability.
PSO	• Search is fast and efficient. • Easy to implement.	• Poor handling of discrete problems. • Easy to fall in local optimality.
CCS	• Strong search capability. • High-quality solution.	• Not easy to implement. • Slow convergence.
VNBA	• High of the search efficiency. • Not easy to fall in local optimality.	• The code runs slowly. • Not easy to implement.

The proposed IMEHO algorithm is evaluated on both 30D and 50D benchmarks. We fixed the maximum number of generations and maximum number of function evaluations, respectively, All the problems are run independently for 30 times. In order to evaluate the IMEHO algorithm more clearly and intuitively, we selected 4 problems (F01, F04, F17, and F29) out of the above 4 groups of problems and drew the average convergence curves. More convergence curves are shown in Appendix. The parameters in IMEHO algorithm are set as follows: the number of the kept individuals k = 5% * N (N is the population size), the clan number $NumClan$ = 5, the inertia weight w decreases linearly and the range of inertia weight: $w \in [0.2, 0.9]$ as shown in Eq. (15). Inertial weight is used to balance global and local search capabilities. Larger inertia weight is more suitable for global search, while smaller inertia weight is more favorable for local search [90]. The acceleration coefficient c = 1.49445 [90]. The probability value pc = 0.05, the range of position: $x \in [-100, 100]$.

$$w = 0.9 - \frac{t}{T_{\max}} \times 0.7 , \tag{15}$$

where $T_{\max}$ is the max generations, t is current generation number.

We used the following four indicators as the criteria for evaluating each algorithm:

(1) **Mean and Std**: Each problem is run 30 times by each algorithm independently and we find their mean and standard deviation (Std) values as the results.

(2) **Best**: Each problem is run 30 times by each algorithm independently and we find their best value.

(3) **Worst**: Each problem is run 30 times by each algorithm independently and we find the worst value.

(4) **Friedman test**: Friedman test is carried out with SPSS software and the Friedman ranking of all benchmark functions is obtained. At the same time, the significance test of mean Tables data under various conditions were also conducted.

5.1 A comparison of different impact factors

There are only two newly parameters in IMEHO algorithm: impact factor α in Eq. (8) and the probability value pc in the separation operator. pc is only a small probability of the event used to select the individual separation. As long as the value of *pc* is small, it will not have a great impact. We set the value of *pc* to 0.05. However, $\alpha \in (0, 1]$ is an impact factor in clan updating operator. It is necessary to study the selection of its value. Hence, we compared the performance of the IMEHO algorithm over ten different α values. In these set of experiments, the population size N was set to 40, the maximum generations ($T_{\max}$) were set to 7500, and the dimensions were set to 30 and 50, respectively. Each method was run 30 times independently. We recorded the mean of the results as shown in Tables 4 - 7. The best values in Tables are marked in **bold**.

Table 4 shows the mean values of different α values on 30D benchmarks. As we can see from the last row, α = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, and 1 obtain their best results on 1, 2, 3, 6, 3, 3, 1, 0, 7, and 4 functions, respectively. It can be seen that the values of α are good results obtained at 0.4 and 0.8.

Table 5 shows the ranking according to the Friedman test on 30D benchmarks. α = 0.4 obtains the best rank, α = 0.6 ranks 2, and the following are 0.7, 0.5, 0.9, 0.8, 0.3, 1, 0.2, and 0.1. It means that the results obtained at α = 0.4 are generally better than other results.

Table 4 Effect of α on the mean of 30D benchmarks

	α									
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
F01	5.21E+06	3.44E+06	3.20E+06	**1.43E+06**	2.75E+06	2.60E+06	2.07E+06	1.96E+06	2.28E+06	2.03E+06
F02	8.47E+03	8.08E+03	6.07E+03	4.80E+03	5.11E+03	4.38E+03	**3.28E+03**	4.72E+03	5.01E+03	5.42E+03
F03	1.35E+03	7.83E+02	4.49E+02	4.88E+02	**4.16E+02**	5.04E+02	4.80E+02	5.53E+02	7.32E+02	7.23E+02
F04	5.37E+02	5.30E+02	5.34E+02	5.18E+02	5.34E+02	5.25E+02	5.26E+02	5.38E+02	**5.17E+02**	5.33E+02
F05	5.21E+02	**5.21E+02**	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02
F06	6.18E+02	6.16E+02	6.14E+02	6.12E+02	6.10E+02	6.11E+02	6.08E+02	6.07E+02	**6.05E+02**	6.05E+02
F07	7.00E+02	7.00E+02	7.00E+02	7.00E+02	7.00E+02	7.00E+02	7.00E+02	7.00E+02	7.00E+02	**7.00E+02**
F08	8.43E+02	8.36E+02	8.34E+02	8.35E+02	8.33E+02	**8.31E+02**	8.34E+02	8.35E+02	8.38E+02	8.35E+02
F09	9.51E+02	**9.31E+02**	9.31E+02	9.33E+02	9.34E+02	9.32E+02	9.35E+02	9.40E+02	9.40E+02	9.41E+02
F10	3.53E+03	3.49E+03	3.38E+03	3.28E+03	3.28E+03	3.48E+03	3.26E+03	3.32E+03	**3.13E+03**	3.24E+03
F11	4.10E+03	4.27E+03	4.05E+03	3.97E+03	**3.85E+03**	3.97E+03	4.20E+03	3.97E+03	3.96E+03	4.08E+03
F12	**1.20E+03**	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03
F13	1.30E+03	1.30E+03	1.30E+03	1.30E+03	1.30E+03	1.30E+03	1.30E+03	1.30E+03	**1.30E+03**	1.30E+03
F14	1.40E+03	1.40E+03	1.40E+03	1.40E+03	1.40E+03	1.40E+03	1.40E+03	1.40E+03	**1.40E+03**	1.40E+03
F15	1.52E+03	1.51E+03	1.51E+03	1.50E+03	1.50E+03	**1.50E+03**	1.50E+03	1.50E+03	1.50E+03	1.50E+03
F16	1.61E+03	1.61E+03	1.61E+03	**1.61E+03**	1.61E+03	1.61E+03	1.61E+03	1.61E+03	1.61E+03	1.61E+03
F17	1.03E+05	6.34E+04	**6.12E+04**	7.03E+04	8.01E+04	8.69E+04	7.94E+04	8.08E+04	1.09E+05	1.57E+05
F18	5.69E+03	5.23E+03	4.60E+03	**4.31E+03**	6.15E+03	5.84E+03	6.10E+03	5.99E+03	6.68E+03	7.98E+03
F19	1.91E+03	1.91E+03	1.91E+03	1.91E+03	1.91E+03	1.91E+03	1.91E+03	1.91E+03	**1.91E+03**	1.91E+03
F20	2.39E+03	2.21E+03	2.20E+03	**2.20E+03**	2.23E+03	2.23E+03	2.27E+03	2.25E+03	2.29E+03	2.25E+03
F21	3.09E+04	3.90E+04	3.27E+04	**2.85E+04**	3.44E+04	3.30E+04	2.90E+04	4.21E+04	4.66E+04	4.63E+04
F22	2.48E+03	2.44E+03	**2.40E+03**	2.41E+03	2.43E+03	2.40E+03	2.41E+03	2.40E+03	2.40E+03	2.41E+03
F23	2.62E+03	2.62E+03	2.62E+03	**2.62E+03**	2.62E+03	2.62E+03	2.62E+03	2.62E+03	2.62E+03	2.62E+03
F24	2.64E+03	2.64E+03	2.64E+03	2.64E+03	2.64E+03	2.64E+03	2.64E+03	2.63E+03	2.63E+03	**2.63E+03**
F25	2.72E+03	2.71E+03	2.71E+03	2.71E+03	2.71E+03	2.71E+03	2.71E+03	2.71E+03	2.71E+03	**2.71E+03**
F26	2.70E+03	2.70E+03	2.70E+03	2.70E+03	**2.70E+03**	2.70E+03	2.70E+03	2.70E+03	2.70E+03	2.70E+03

F27	3.31E+03	3.32E+03	3.28E+03	3.24E+03	3.19E+03	3.22E+03	3.16E+03	3.15E+03	3.14E+03	**3.13E+03**
F28	3.87E+03	3.83E+03	3.85E+03	3.85E+03	3.76E+03	**3.73E+03**	3.82E+03	3.78E+03	3.73E+03	3.76E+03
F29	1.35E+06	4.19E+03	**4.16E+03**	7.07E+05	4.19E+03	4.24E+03	4.23E+03	7.70E+05	4.26E+03	4.37E+03
F30	8.06E+03	9.51E+03	6.88E+03	7.69E+03	6.67E+03	7.02E+03	7.01E+03	6.96E+03	**6.36E+03**	6.48E+03
	1	2	3	6	3	3	1	0	7	4

Table 5 Ranking of different α values according to the Friedman test on 30D benchmarks

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Friedman rank	7.88	6.43	5.13	4.87	5	4.9	5	5.3	5.02	5.47
Final rank	9	8	6	1	3	2	3	5	4	7

Table 6 shows the mean values of different α values on 50D benchmarks. As we can see from the last row, α = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, and 1 obtain their best results on 7, 2, 2, 6, 3, 1, 2, 0, 2, and 5 functions, respectively. It also can be seen that the values of α are good results obtained at 0.1 and 0.4. Table 7 shows the ranking according to the Friedman test on 50D benchmarks. α = 0.8 obtains the best rank, α = 0.6 ranks 2, α = 0.4 ranks 3, and the following are 0.5, 0.9, 1, 0.7, 0.1,0.3, and 0.2. Although the optimal solution was not found when α = 0.8 in Table 6, the results in Table 7 indicate that the results obtained at α = 0.8 are generally superior to the others.

Table 6 Effect of α on the mean of 50D benchmarks

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
F01	**3.68E+06**	8.46E+06	9.07E+06	7.11E+06	6.72E+06	5.05E+06	5.37E+06	5.52E+06	6.85E+06	6.67E+06
F02	7.93E+03	1.02E+06	4.21E+03	7.28E+03	4.21E+03	4.85E+03	6.24E+03	4.73E+03	6.11E+03	**3.27E+03**
F03	8.07E+03	1.55E+03	1.26E+03	**1.19E+03**	1.70E+03	2.66E+03	2.22E+03	2.30E+03	2.87E+03	3.03E+03
F04	**5.34E+02**	5.96E+02	5.90E+02	5.77E+02	5.65E+02	5.63E+02	5.81E+02	5.69E+02	5.62E+02	5.67E+02
F05	5.21E+02	5.21E+02	5.21E+02	**5.21E+02**	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02
F06	6.31E+02	6.38E+02	6.32E+02	6.29E+02	6.27E+02	6.24E+02	6.22E+02	6.20E+02	6.18E+02	**6.16E+02**
F07	7.00E+02	7.00E+02	7.00E+02	7.00E+02	7.00E+02	7.00E+02	7.00E+02	7.00E+02	**7.00E+02**	7.00E+02
F08	9.06E+02	8.64E+02	**8.61E+02**	8.61E+02	8.66E+02	8.70E+02	8.72E+02	8.68E+02	8.74E+02	8.76E+02
F09	1.01E+03	**9.48E+02**	9.50E+02	9.58E+02	9.57E+02	9.66E+02	9.68E+02	9.84E+02	9.85E+02	9.79E+02
F10	**5.30E+03**	6.34E+03	6.17E+03	5.67E+03	5.86E+03	5.83E+03	5.90E+03	6.14E+03	5.44E+03	5.61E+03
F11	6.86E+03	7.40E+03	7.45E+03	7.10E+03	**6.74E+03**	7.12E+03	6.90E+03	6.79E+03	7.10E+03	6.85E+03
F12	**1.20E+03**	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03
F13	1.30E+03	1.30E+03	1.30E+03	**1.30E+03**	1.30E+03	1.30E+03	1.30E+03	1.30E+03	1.30E+03	1.30E+03
F14	1.40E+03	1.40E+03	1.40E+03	1.40E+03	**1.40E+03**	1.40E+03	1.40E+03	1.40E+03	1.40E+03	1.40E+03
F15	1.54E+03	1.55E+03	1.53E+03	1.52E+03	1.51E+03	1.51E+03	**1.51E+03**	1.51E+03	1.51E+03	1.51E+03
F16	1.62E+03	1.62E+03	1.62E+03	**1.62E+03**	1.62E+03	1.62E+03	1.62E+03	1.62E+03	1.62E+03	1.62E+03
F17	2.49E+05	3.17E+05	4.73E+05	2.93E+05	3.33E+05	**2.24E+05**	3.57E+05	2.82E+05	3.12E+05	3.70E+05
F18	3.39E+03	2.88E+03	**2.44E+03**	2.73E+03	2.81E+03	2.53E+03	2.77E+03	2.74E+03	3.46E+03	3.25E+03
F19	**1.93E+03**	1.96E+03	1.96E+03	1.96E+03	1.97E+03	1.96E+03	1.95E+03	1.96E+03	1.95E+03	1.94E+03
F20	6.02E+03	2.47E+03	2.46E+03	**2.44E+03**	2.50E+03	2.60E+03	2.72E+03	2.80E+03	2.95E+03	2.94E+03
F21	1.62E+05	**1.44E+05**	1.54E+05	1.59E+05	1.67E+05	1.56E+05	3.00E+05	1.71E+05	2.32E+05	3.82E+05
F22	2.94E+03	3.03E+03	2.91E+03	2.82E+03	2.89E+03	2.78E+03	**2.74E+03**	2.78E+03	2.75E+03	2.81E+03

F23	**2.65E+03**	2.65E+03	2.65E+03	2.65E+03	2.65E+03	2.65E+03	2.65E+03	2.65E+03	2.65E+03	2.65E+03
F24	2.69E+03	2.70E+03	2.69E+03	2.69E+03	2.69E+03	2.69E+03	2.68E+03	2.68E+03	2.68E+03	**2.68E+03**
F25	2.72E+03	2.74E+03	2.73E+03	2.73E+03	2.73E+03	2.72E+03	2.72E+03	2.72E+03	2.72E+03	**2.72E+03**
F26	2.70E+03	2.71E+03	2.70E+03	2.70E+03	**2.70E+03**	2.70E+03	2.70E+03	2.70E+03	2.70E+03	2.70E+03
F27	3.80E+03	3.99E+03	3.91E+03	3.81E+03	3.73E+03	3.69E+03	3.62E+03	3.58E+03	3.50E+03	**3.49E+03**
F28	4.67E+03	4.60E+03	4.57E+03	4.54E+03	4.59E+03	4.58E+03	4.54E+03	4.52E+03	**4.51E+03**	4.67E+03
F29	1.38E+04	5.14E+03	3.53E+06	**4.65E+03**	4.74E+03	4.99E+03	4.97E+03	4.93E+03	4.99E+03	4.95E+03
F30	**3.13E+04**	5.10E+04	4.74E+04	4.48E+04	3.99E+04	4.79E+04	4.41E+04	4.19E+04	4.20E+04	4.55E+04
	7	2	2	6	3	1	2	0	2	5

Table 7 Ranking of different α value according to the Friedman test on 50D benchmarks

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Friedman rank	6.13	7.07	6.47	4.87	5.1	4.8	5.47	4.63	5.1	5.37
Final rank	7	9	8	3	4	2	6	1	4	5

As seen in Table 5, α = 0.4 has the best performance, followed by α = 0.6, while α = 0.8 has poor performance. Referring to Table 7, α = 0.8 has the best performance for 50D benchmarks, followed by α = 0.6, and α = 0.4 ranks 3. In Table 4, α = 0.4 can find 6 optimal solutions (ranks 2), α = 0.6 can find 3 optimal solutions, and α = 0.8 cannot find the optimal solution. And in Table 6, α = 0.4 can find 6 optimal solutions (ranks 2), α = 0.6 can find 1 optimal solution and α = 0.8 cannot find the optimal solution. Based on the above statement, we believe that α = 0.4 has a good performance. Therefore, in the following experiments, we all set the value of α to 0.4.

5.2 Comparison of different strategies

In this section, we tested our design strategy separately. The proposed velocity strategy combined with the learning strategy is termed as EHO_LS. The separation strategy and elitism strategy are represented by EHO_SS and EHO_ES, respectively. The IMEHO and EHO algorithms are both tested. We set the maximum number of generations (T_{max}) to 7500. Each strategy was run 30 times independently on each problem and the mean values were recorded. Tables 8 and 9 record the results of these strategies on 30D benchmarks. Tables 10 and 11 record the results of these strategies on 50D benchmarks. The best values in Tables are marked in **bold**.

As seen in Table 8, the proposed learning strategy has an acceptable performance. The last row shows a comparison between the current strategy and the EHO algorithm on 30 benchmark functions. The results show that IMEHO, EHO-LS, EHO-SS, and EHO-ES are superior to the standard EHO algorithm on 25, 24, 23, and 30 functions, respectively. It also can be seen from the penultimate row that EHO, IMEHO, EHO-LS, EHO-SS, and EHO-ES can find 0, 14, 11, 0, and 5 optimal results from 30 benchmark functions, respectively.

Table 9 shows the ranking of the strategies in Table 8 according to the Friedman test. It can be seen that IMEHO obtains the best rank, EHO-LS ranks 2, followed by EHO-ES, EHO-SS, and EHO. The role of the learning strategy in the EHO algorithm can also be seen in Table 10. Referring to the penultimate row, EHO, IMEHO, EHO-LS, EHO-SS, and EHO-ES can find 0, 13, 12, 0, and 5 optimal functions from 30 benchmark functions, respectively. The last row shows that IMEHO, EHO-LS, EHO-SS, and EHO-ES are better than EHO on 25, 25, 26, and 30 functions, respectively. Table 11 shows the ranking of the strategies in Table 10 according to the Friedman test. As seen, EHO-LS obtains the best rank, IMEHO ranks 2, followed by EHO-ES, EHO-SS, and EHO. Tables 8 – 11 indicate that the proposed strategies have different effects in the EHO algorithm. In particular, the proposed learning strategy and velocity strategy have made the EHO performance better. Because the learning strategy in the algorithm makes different individuals have different learning objectives, the results obtained are getting better. Secondly, although the separation strategy does not achieve optimal results, it is superior to the standard EHO algorithm in most functions. At the same time, the application of the separation strategy greatly improves the exploitation ability of the algorithm. The choice of elitism strategy which keeps the best individuals also improves the performance of the standard EHO algorithm. Finally, although the learning strategy has a good performance, the combination of all strategies is the best choice because it makes the algorithm more stable.

Table 8 Comparison of different strategies on 30D benchmarks

	EHO	IMEHO	EHO-LS	EHO-SS	EHO-ES
F01	2.75E+09	**1.43E+06**	1.48E+06	2.41E+09	4.54E+08
F02	1.02E+11	**4.80E+03**	9.00E+03	9.01E+10	3.44E+10
F03	2.35E+06	**4.88E+02**	6.43E+03	8.79E+05	5.35E+04
F04	2.45E+04	5.18E+02	**5.12E+02**	2.19E+04	4.74E+03
F05	5.21E+02	**5.21E+02**	5.21E+02	5.21E+02	5.21E+02
F06	6.48E+02	6.12E+02	**6.02E+02**	6.47E+02	6.36E+02
F07	1.74E+03	7.00E+02	**7.00E+02**	1.62E+03	9.86E+02
F08	1.28E+03	**8.35E+02**	8.46E+02	1.20E+03	1.07E+03
F09	1.36E+03	**9.33E+02**	9.45E+02	1.36E+03	1.25E+03
F10	1.01E+04	3.28E+03	**2.81E+03**	1.01E+04	7.84E+03
F11	1.09E+04	3.97E+03	**3.28E+03**	1.03E+04	8.16E+03
F12	1.21E+03	**1.20E+03**	1.21E+03	1.21E+03	1.20E+03
F13	1.31E+03	**1.30E+03**	1.30E+03	1.31E+03	1.30E+03
F14	1.76E+03	1.40E+03	**1.40E+03**	1.75E+03	1.51E+03
F15	9.91E+05	1.50E+03	**1.50E+03**	9.34E+05	2.88E+05
F16	1.61E+03	**1.61E+03**	1.61E+03	1.61E+03	1.61E+03
F17	4.50E+08	**7.03E+04**	1.10E+05	3.69E+08	7.89E+06
F18	1.04E+10	**4.31E+03**	8.75E+03	7.97E+09	3.51E+08
F19	2.73E+03	1.91E+03	**1.91E+03**	2.64E+03	2.11E+03
F20	5.41E+07	**2.20E+03**	1.71E+04	1.59E+07	1.95E+04
F21	2.33E+08	**2.85E+04**	6.76E+04	1.73E+08	1.85E+06
F22	7.64E+04	2.41E+03	**2.32E+03**	7.93E+04	3.25E+03
F23	2.52E+03	2.62E+03	2.62E+03	2.52E+03	**2.50E+03**
F24	2.60E+03	2.64E+03	2.61E+03	2.60E+03	**2.60E+03**
F25	2.70E+03	2.71E+03	2.70E+03	2.70E+03	**2.70E+03**
F26	2.80E+03	**2.70E+03**	2.70E+03	2.80E+03	2.71E+03
F27	3.16E+03	3.24E+03	3.04E+03	3.11E+03	**2.90E+03**
F28	3.06E+03	3.85E+03	3.73E+03	3.05E+03	**3.00E+03**
F29	2.46E+07	7.07E+05	**4.42E+03**	2.60E+07	3.52E+05
F30	1.81E+06	7.69E+03	**4.92E+03**	1.86E+06	5.15E+05

	0	14	11	0	5
		25	24	23	30

Table 9 Ranking of the strategies in Table 8 according to the Friedman test

	EHO	IMEHO	EHO-LS	EHO-SS	EHO-ES
Friedman rank	4.4	2.03	2.17	3.87	2.53
Final rank	4	1	2	4	3

Table 10 Comparison of different strategies on 50D benchmarks

	EHO	IMEHO	EHO-LS	EHO-SS	EHO-ES
F01	9.73E+09	7.11E+06	**6.96E+06**	9.04E+09	1.33E+09
F02	1.98E+11	7.28E+03	**5.73E+03**	1.92E+11	1.19E+11
F03	1.28E+07	**1.19E+03**	2.74E+04	4.02E+06	8.80E+04
F04	7.21E+04	5.77E+02	**5.44E+02**	6.74E+04	2.26E+04
F05	5.21E+02	**5.21E+02**	5.21E+02	5.21E+02	5.21E+02
F06	6.84E+02	6.29E+02	**6.10E+02**	6.82E+02	6.66E+02
F07	2.57E+03	7.00E+02	**7.00E+02**	2.53E+03	1.81E+03
F08	1.68E+03	**8.61E+02**	8.89E+02	1.57E+03	1.37E+03
F09	1.89E+03	**9.58E+02**	1.00E+03	1.85E+03	1.57E+03
F10	1.75E+04	5.67E+03	**5.22E+03**	1.73E+04	1.41E+04
F11	1.82E+04	7.10E+03	**6.14E+03**	1.79E+04	1.48E+04
F12	1.21E+03	**1.20E+03**	1.21E+03	1.21E+03	1.20E+03
F13	1.31E+03	**1.30E+03**	1.30E+03	1.31E+03	1.31E+03
F14	1.88E+03	**1.40E+03**	1.40E+03	1.86E+03	1.74E+03
F15	2.53E+07	1.52E+03	**1.51E+03**	2.24E+07	2.91E+06
F16	1.62E+03	**1.62E+03**	1.62E+03	1.62E+03	1.62E+03
F17	1.31E+09	2.93E+05	**1.87E+05**	1.18E+09	1.05E+08
F18	2.97E+10	2.73E+03	**2.72E+03**	2.55E+10	3.97E+09
F19	7.28E+03	1.96E+03	**1.93E+03**	5.85E+03	2.42E+03
F20	3.87E+07	**2.44E+03**	3.10E+04	1.54E+07	2.63E+04
F21	4.08E+08	**1.59E+05**	3.17E+05	3.02E+08	7.44E+06
F22	1.32E+06	**2.82E+03**	2.85E+03	6.10E+05	6.28E+03
F23	2.53E+03	2.65E+03	2.65E+03	2.53E+03	**2.50E+03**
F24	2.61E+03	2.69E+03	2.68E+03	2.60E+03	**2.60E+03**
F25	2.70E+03	2.73E+03	2.71E+03	2.70E+03	**2.70E+03**
F26	2.80E+03	**2.70E+03**	2.71E+03	2.80E+03	2.73E+03
F27	3.31E+03	3.81E+03	3.27E+03	3.28E+03	**2.90E+03**
F28	3.13E+03	4.54E+03	4.24E+03	3.13E+03	**3.00E+03**
F29	5.72E+07	**4.65E+03**	5.17E+03	5.73E+07	8.97E+05
F30	2.58E+06	4.48E+04	**2.84E+04**	2.49E+06	4.19E+04
	0	13	12	0	5
		25	25	26	30

Table 11 Ranking of the strategies in Table 10 according to the Friedman test

	EHO	IMEHO	EHO-LS	EHO-SS	EHO-ES
Friedman rank	4.53	2.13	2.07	3.77	2.5
Final rank	5	2	1	4	3

5.3 Comparison of fixed number of generations

In this section, we set the maximum number of generations ($T_{\max}$) to 7500. Each algorithm was run 30 times independently on each problem and the results were recorded. Tables 12 - 20 record the results of these algorithms on 30D benchmarks. And Tables 21 - 29 record the results of these algorithms on 50D benchmarks.

5.3.1 *D* = 30

Table 12 shows the mean values of IMEHO algorithm and other algorithms on 30D benchmarks. The last row at the bottom shows that BA, EHO, ES, GA, IMEHO, PBIL, PSO, CCS, and VNBA obtain their best results on 0, 5, 0, 4, 21, 0, 0, 0, and 0 functions, respectively. It can be seen that IMEHO performs well on F01 - F04, F06 - F09, F13 - F22, F26, F29, and F30. The penultimate row shows that IMEHO algorithm performs better than BA, EHO, ES, GA, PBIL, PSO, CCS, and VNBA on 30, 25, 30, 26, 29, 30, 29, and 29 functions, respectively.

Table 13 shows the ranking of the algorithms in Table 12 according to the Friedman test. IMEHO obtains the best rank, GA ranks 2, and the following are VNBA, EHO, CCS, PBIL, ES, BA, and PSO.

Table 14 shows the *p*-value obtained by pairwise comparison between IMEHO and other algorithms in Table 12 according to the Friedman test. As can be seen from the table, the *p*-value obtained by pairwise comparison between IMEHO and all other algorithms is less than 0.05. This indicates that IMEHO is significantly different from other algorithms.

Table 15 shows the Std values of IMEHO algorithm and other algorithms on 30D benchmarks. The last row at the bottom shows that BA, EHO, ES, GA, IMEHO, PBIL, PSO, CCS, and VNBA obtain their best results on 0, 5, 3, 3, 17, 1, 0, 1, and 0 functions, respectively. It also can be seen that IMEHO performs well on F01 - F03, F07, F09, F13 - F15, F17 - F22, F26, F29, and F30. The penultimate row shows that IMEHO algorithm performs better than BA, EHO, ES, GA, PBIL, PSO, CCS, and VNBA on 26, 19, 22, 23, 23, 25, 23, and, 22 functions, respectively.

Table 16 shows the ranking of the algorithms in Table 15 according to the Friedman test. IMEHO obtains the best rank, EHO ranks 2, and the following are GA, CCS, PBIL, VNBA, ES, PSO, and BA.

Table 17 shows the best values of IMEHO algorithm and other algorithms on 30D benchmarks. The last row at the bottom shows that BA, EHO, ES, GA, IMEHO, PBIL, PSO, CCS, and VNBA obtain their best results on 0, 3, 0, 4, 21, 0, 0, 0, and 2 functions, respectively. It also can be seen that IMEHO performs well on F01 - F04, F06 - F09, F12 - F22, F26, and F29. The penultimate row shows that IMEHO algorithm performs better than BA, EHO, ES, GA, PBIL, PSO, CCS, and VNBA on 30, 29, 30, 28, 30, 30, 30, and 28 functions, respectively.

Table 18 shows the ranking of the algorithms in Table 17 according to the Friedman test on 30D benchmarks. IMEHO obtains the best rank, GA ranks 2, and the following are VNBA, EHO, CCS, PBIL, ES, BA, and PSO.

Table 19 shows the worst values of IMEHO algorithm and other algorithms on 30D benchmarks. The last row at the bottom shows that BA, EHO, ES, GA, IMEHO, PBIL, PSO, CCS, and VNBA obtain their best results on 0, 5, 0, 3, 22, 0, 0, 0, and 0 functions, respectively. It also can be seen that IMEHO performs well on F01 - F04, F06 - F09, F11, F13 - F22, F26, F29, and F30. The penultimate row shows that IMEHO performs better than BA, EHO, ES, GA, PBIL, PSO, CCS, and VNBA on 30, 25, 30, 27, 29, 30, 29, and 30 functions, respectively.

Table 20 shows the ranking of the algorithms in Table 19 according to the Friedman test on 30D benchmarks. IMEHO obtains the best rank, GA ranks 2, and the following are EHO, VNBA, CCS, PBIL, ES, BA, and PSO.

As seen in Tables 12 - 20, the IMEHO algorithm can always find the optimal solution on most problems. In particular, the role of learning strategy allows individuals to learn from each other. In Tables 12, 17, and 19, the IMEHO algorithm can find more than 20 optimal solutions among 30 problems. It can also be seen in Table 15 that the IMEHO algorithm has more stable performance than other algorithms. On the other hand, from the perspective of Friedman rank, the IMEHO algorithm always ranks 1 compared with other algorithms. In Table 14, the *p*-value obtained by pairwise comparison between IMEHO and all other algorithms is less than 0.05 which indicates that IMEHO is significantly different from other algorithms. The convergence curves of the 4 problems (F01, F04, F17, and F29) can be shown in Figure 4. As seen, the IMEHO algorithm has faster rate of convergence than other algorithms and always gets the best solution.

Table 12 Mean values for a fixed number of generations on 30D benchmarks

	BA	EHO	ES	GA	IMEHO	PBIL	PSO	CCS	VNBA
F01	2.65E+09	4.38E+08	1.29E+09	1.89E+07	**2.38E+06**	3.43E+08	3.34E+09	1.46E+08	2.43E+08
F02	1.03E+11	3.47E+10	8.23E+10	3.44E+06	**5.69E+03**	4.08E+10	9.92E+10	2.60E+09	1.92E+10
F03	2.72E+05	5.22E+04	2.08E+05	1.18E+04	**4.41E+02**	9.22E+04	2.66E+07	2.70E+05	2.96E+04
F04	2.63E+04	4.96E+03	1.66E+04	5.28E+02	**5.24E+02**	3.83E+03	2.47E+04	7.22E+02	2.20E+03
F05	5.21E+02	5.21E+02	5.21E+02	**5.21E+02**	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02
F06	6.48E+02	6.37E+02	6.43E+02	6.25E+02	**6.12E+02**	6.38E+02	6.49E+02	6.25E+02	6.33E+02
F07	1.61E+03	9.88E+02	1.40E+03	7.02E+02	**7.00E+02**	1.04E+03	1.77E+03	7.23E+02	8.11E+02
F08	1.24E+03	1.06E+03	1.19E+03	8.44E+02	**8.33E+02**	1.10E+03	1.27E+03	1.09E+03	9.74E+02
F09	1.41E+03	1.25E+03	1.37E+03	1.07E+03	**9.32E+02**	1.27E+03	1.45E+03	1.19E+03	1.15E+03
F10	9.98E+03	7.68E+03	8.98E+03	**1.05E+03**	3.26E+03	7.76E+03	8.65E+03	9.55E+03	4.50E+03
F11	1.03E+04	8.03E+03	8.96E+03	**3.91E+03**	3.96E+03	8.20E+03	9.07E+03	9.93E+03	7.90E+03
F12	1.20E+03	1.20E+03	1.20E+03	**1.20E+03**	1.20E+03	1.20E+03	1.20E+03	1.21E+03	1.20E+03
F13	1.31E+03	1.30E+03	1.31E+03	1.30E+03	**1.30E+03**	1.30E+03	1.32E+03	1.30E+03	1.30E+03
F14	1.72E+03	1.51E+03	1.65E+03	1.40E+03	**1.40E+03**	1.50E+03	1.82E+03	1.41E+03	1.46E+03
F15	6.14E+06	3.23E+05	5.29E+06	1.52E+03	**1.50E+03**	6.86E+05	1.81E+06	1.58E+03	3.89E+03
F16	1.61E+03	1.61E+03	1.61E+03	1.61E+03	**1.61E+03**	1.61E+03	1.61E+03	1.61E+03	1.61E+03
F17	2.79E+08	9.06E+06	4.54E+07	5.30E+06	**7.86E+04**	9.75E+06	3.67E+08	1.15E+07	7.53E+06
F18	5.88E+09	3.02E+08	2.77E+09	5.47E+05	**5.10E+03**	6.16E+08	1.16E+10	1.10E+08	1.66E+08
F19	2.83E+03	2.10E+03	2.39E+03	1.95E+03	**1.91E+03**	2.09E+03	3.04E+03	1.94E+03	2.02E+03
F20	9.38E+06	1.89E+04	1.39E+06	3.52E+04	**2.21E+03**	3.79E+04	2.72E+06	1.03E+06	1.89E+04
F21	1.25E+08	2.04E+06	1.71E+07	9.58E+05	**2.93E+04**	2.53E+06	2.00E+08	5.67E+06	2.31E+06
F22	1.62E+04	3.36E+03	4.16E+03	2.89E+03	**2.41E+03**	3.22E+03	4.48E+06	3.54E+03	3.04E+03
F23	3.96E+03	**2.50E+03**	3.31E+03	2.64E+03	2.62E+03	2.90E+03	4.12E+03	2.65E+03	2.69E+03
F24	2.83E+03	**2.60E+03**	2.85E+03	2.65E+03	2.64E+03	2.80E+03	2.74E+03	2.67E+03	2.63E+03

F25	2.85E+03	**2.70E+03**	2.81E+03	2.72E+03	2.71E+03	2.74E+03	2.88E+03	2.72E+03	2.71E+03
F26	2.74E+03	2.71E+03	2.71E+03	2.71E+03	**2.70E+03**	2.70E+03	3.07E+03	2.70E+03	2.70E+03
F27	4.56E+03	**2.90E+03**	4.08E+03	3.57E+03	3.28E+03	3.78E+03	5.01E+03	3.23E+03	3.99E+03
F28	8.14E+03	**3.00E+03**	5.12E+03	4.27E+03	3.77E+03	4.14E+03	1.49E+04	4.24E+03	4.48E+03
F29	3.13E+07	3.57E+05	1.38E+07	9.28E+05	**4.11E+03**	5.71E+06	1.26E+09	1.21E+06	7.48E+06
F30	8.01E+06	5.34E+05	8.94E+05	9.01E+03	**7.08E+03**	1.52E+05	2.82E+07	7.97E+04	1.92E+05
	30	25	30	26		29	30	29	29
	0	**5**	**0**	**4**	**21**	**0**	**0**	**0**	**0**

Table 13 Ranking of the algorithms in Table 12 according to the Friedman test

	BA	EHO	ES	GA	IMEHO	PBIL	PSO	CCS	VNBA
Friedman rank	8.20	4.07	6.93	2.47	**1.40**	5.07	8.37	4.7	3.8
Final rank	8	4	7	2	**1**	6	9	5	3

Table 14 *p*-value of IMEHO compared with other algorithms in Table 12

	IMEHO-BA	IMEHO-EHO	IMEHO-ES	IMEHO-GA	IMEHO-PBIL	IMEHO-PSO	IMEHO-CCS	IMEHO-VNBA
p-value	4.32E-08	2.61E-04	4.32E-08	5.90E-05	3.19E-07	4.32E-08	3.19E-07	3.19E-07

Table 15 Std values for a fixed number of generations on 30D benchmarks

	BA	EHO	ES	GA	IMEHO	PBIL	PSO	CCS	VNBA
F01	1.06E+09	6.56E+07	2.38E+08	1.50E+07	**4.32E+06**	1.09E+08	1.02E+09	3.27E+07	5.93E+07
F02	1.98E+10	3.84E+09	1.18E+10	7.04E+04	**4.87E+03**	3.39E+09	8.08E+09	5.22E+08	4.23E+09
F03	8.45E+04	5.26E+03	5.49E+04	1.08E+04	**1.58E+02**	1.75E+04	3.79E+07	7.74E+04	1.39E+04
F04	8.13E+03	7.25E+02	4.42E+03	4.24E+01	4.77E+01	7.56E+02	5.63E+03	**4.09E+01**	3.63E+02
F05	8.60E-02	5.54E-02	**3.56E-02**	7.57E-02	5.99E-02	5.56E-02	5.81E-02	8.81E-02	5.43E-02
F06	2.24E+00	1.25E+00	1.38E+00	2.09E+00	2.72E+00	**1.16E+00**	2.37E+00	2.00E+00	2.58E+00
F07	1.58E+02	2.65E+01	9.27E+01	4.25E-01	**1.19E-01**	2.74E+01	1.12E+02	3.52E+00	1.81E+01
F08	3.16E+01	1.78E+01	2.61E+01	**7.30E+00**	9.19E+00	1.03E+01	3.39E+01	2.23E+01	1.61E+01
F09	6.89E+01	2.61E+01	3.66E+01	3.04E+01	**1.15E+01**	1.69E+01	4.92E+01	2.38E+01	2.03E+01
F10	5.59E+02	3.30E+02	2.89E+02	**1.64E+01**	5.72E+02	3.05E+02	3.20E+02	4.91E+02	3.47E+02
F11	3.85E+02	2.90E+02	**2.36E+02**	6.25E+02	5.38E+02	2.97E+02	4.36E+02	5.50E+02	3.79E+02
F12	9.23E-01	2.24E-01	3.12E-01	**1.97E-01**	5.26E-01	3.38E-01	5.46E-01	1.09E+00	3.51E-01
F13	1.27E+00	2.04E-01	6.94E-01	1.78E-01	**6.25E-02**	2.56E-01	1.58E+00	1.76E-01	3.64E-01
F14	5.83E+01	1.24E+01	3.55E+01	2.51E-01	**9.85E-02**	1.16E+01	5.76E+01	1.88E+00	1.22E+01
F15	5.44E+06	1.13E+05	2.41E+06	5.04E+00	**1.35E+00**	2.85E+05	8.28E+05	3.03E+01	1.22E+03
F16	2.01E-01	1.63E-01	**1.10E-01**	4.35E-01	7.64E-01	2.12E-01	2.09E-01	1.75E-01	3.66E-01
F17	1.75E+08	2.35E+06	2.36E+07	2.85E+06	**8.38E+04**	2.79E+06	1.61E+08	4.59E+06	3.34E+06
F18	2.30E+09	8.90E+07	7.80E+08	1.73E+05	**3.52E+03**	1.68E+08	3.68E+09	4.66E+07	1.03E+08
F19	3.14E+02	3.33E+01	1.25E+02	3.56E+01	**1.74E+00**	3.42E+01	3.88E+02	5.91E+00	3.82E+01
F20	1.21E+07	3.73E+03	1.25E+06	1.97E+04	**8.17E+01**	1.77E+04	5.05E+06	9.05E+05	6.57E+03
F21	8.51E+07	5.50E+05	6.47E+06	7.60E+05	**1.83E+04**	1.19E+06	1.18E+08	2.73E+06	1.35E+06
F22	2.27E+04	1.35E+02	3.70E+02	2.01E+02	**1.01E+02**	1.88E+02	5.08E+06	1.56E+02	1.28E+02
F23	4.68E+02	**1.49E-02**	1.39E+02	3.14E+01	4.78E-01	6.70E+01	6.03E+02	7.66E+00	2.47E+01
F24	4.58E+01	**3.17E-03**	2.48E+01	6.38E+00	6.46E+00	1.42E+01	1.03E+01	2.51E+00	2.53E+01

F25	4.04E+01	**1.84E-04**	1.72E+01	4.44E+00	2.08E+00	6.00E+00	4.39E+01	4.44E+00	1.07E+01
F26	6.25E+01	1.35E+00	1.95E+00	2.54E+01	**6.00E-02**	2.09E-01	1.09E+02	2.06E-01	4.30E-01
F27	2.80E+02	**2.46E-03**	8.34E+01	2.05E+02	1.41E+02	2.80E+02	2.92E+02	7.92E+01	3.30E+01
F28	1.58E+03	**5.49E-03**	1.77E+02	4.29E+02	2.44E+02	1.33E+02	1.82E+03	9.46E+01	2.33E+02
F29	6.96E+07	1.77E+04	2.08E+06	2.81E+06	**2.16E+02**	3.33E+06	4.18E+08	7.03E+05	1.20E+06
F30	5.76E+06	3.82E+05	2.07E+05	2.35E+03	**1.44E+03**	5.55E+04	1.67E+07	3.37E+04	1.03E+05
	26	19	22	23		23	25	23	22
	0	**5**	**3**	**3**	**17**	**1**	**0**	**1**	**0**

Table 16 Ranking of the algorithms in Table 15 according to the Friedman test

	BA	EHO	ES	GA	IMEHO	PBIL	PSO	CCS	VNBA
Friedman rank	8.00	3.27	5.67	4.03	**2.90**	4.53	7.83	4.1	4.67
Final rank	9	2	7	3	**1**	5	8	4	6

Table 17 Best values for a fixed number of generations on 30D benchmarks

	BA	EHO	ES	GA	IMEHO	PBIL	PSO	CCS	VNBA
F01	7.90E+08	3.12E+08	7.74E+08	3.40E+06	**4.72E+04**	1.54E+08	1.86E+09	8.87E+07	1.22E+08
F02	5.61E+10	2.68E+10	5.74E+10	3.39E+06	**2.62E+02**	3.36E+10	7.11E+10	1.44E+09	9.73E+09
F03	1.47E+05	4.06E+04	1.15E+05	4.25E+02	**3.04E+02**	5.75E+04	3.79E+05	1.71E+05	1.74E+04
F04	1.39E+04	3.55E+03	8.82E+03	4.33E+02	**4.00E+02**	2.79E+03	1.09E+04	6.53E+02	1.67E+03
F05	5.21E+02	5.21E+02	5.21E+02	**5.20E+02**	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02
F06	6.42E+02	6.34E+02	6.39E+02	6.21E+02	**6.06E+02**	6.35E+02	6.44E+02	6.20E+02	6.27E+02
F07	1.27E+03	9.30E+02	1.14E+03	7.02E+02	**7.00E+02**	9.71E+02	1.56E+03	7.16E+02	7.75E+02
F08	1.19E+03	1.03E+03	1.12E+03	8.32E+02	**8.19E+02**	1.07E+03	1.20E+03	1.05E+03	9.38E+02
F09	1.28E+03	1.18E+03	1.30E+03	1.02E+03	**9.13E+02**	1.23E+03	1.35E+03	1.13E+03	1.09E+03
F10	8.47E+03	6.89E+03	8.29E+03	**1.03E+03**	2.01E+03	7.15E+03	7.74E+03	8.65E+03	3.78E+03
F11	9.41E+03	7.21E+03	8.43E+03	**2.50E+03**	2.64E+03	7.49E+03	7.98E+03	8.37E+03	6.95E+03
F12	1.20E+03	1.20E+03	1.20E+03	1.20E+03	**1.20E+03**	1.20E+03	1.20E+03	1.20E+03	1.20E+03
F13	1.31E+03	1.30E+03	1.31E+03	1.30E+03	**1.30E+03**	1.30E+03	1.31E+03	1.30E+03	1.30E+03
F14	1.60E+03	1.48E+03	1.57E+03	1.40E+03	**1.40E+03**	1.47E+03	1.72E+03	1.40E+03	1.44E+03
F15	2.27E+05	1.33E+05	1.10E+06	1.51E+03	**1.50E+03**	2.70E+05	8.34E+05	1.54E+03	2.19E+03
F16	1.61E+03	1.61E+03	1.61E+03	1.61E+03	**1.61E+03**	1.61E+03	1.61E+03	1.61E+03	1.61E+03
F17	3.23E+07	5.32E+06	2.34E+07	1.83E+06	**6.17E+03**	4.25E+06	1.09E+08	4.65E+06	1.89E+06
F18	1.37E+09	9.89E+07	1.46E+09	1.58E+05	**1.91E+03**	3.12E+08	4.03E+09	2.54E+07	1.01E+07
F19	2.24E+03	2.00E+03	2.15E+03	1.92E+03	**1.91E+03**	2.02E+03	2.33E+03	1.93E+03	1.94E+03
F20	1.29E+05	1.11E+04	3.35E+05	5.01E+03	**2.08E+03**	1.54E+04	2.40E+05	1.10E+05	7.45E+03
F21	1.52E+07	1.21E+06	8.08E+06	1.85E+05	**3.89E+03**	1.11E+06	5.17E+07	1.23E+06	5.37E+05
F22	3.93E+03	3.06E+03	3.51E+03	2.42E+03	**2.22E+03**	2.84E+03	1.65E+05	3.22E+03	2.67E+03
F23	2.98E+03	**2.50E+03**	3.05E+03	2.62E+03	2.62E+03	2.77E+03	3.37E+03	2.64E+03	2.66E+03
F24	2.74E+03	2.60E+03	2.77E+03	2.64E+03	2.62E+03	2.76E+03	2.72E+03	2.66E+03	**2.60E+03**
F25	2.78E+03	2.70E+03	2.76E+03	2.71E+03	2.71E+03	2.73E+03	2.77E+03	2.72E+03	**2.70E+03**
F26	2.71E+03	2.70E+03	2.71E+03	2.70E+03	**2.70E+03**	2.70E+03	2.75E+03	2.70E+03	2.70E+03
F27	3.79E+03	**2.90E+03**	3.74E+03	3.12E+03	3.10E+03	3.31E+03	4.60E+03	3.16E+03	3.89E+03
F28	4.98E+03	**3.00E+03**	4.72E+03	3.71E+03	3.54E+03	4.03E+03	1.08E+04	4.11E+03	4.14E+03

F29	8.84E+05	3.25E+05	9.87E+06	5.23E+03	**3.64E+03**	1.94E+06	6.52E+08	3.34E+05	5.15E+06
F30	1.10E+06	2.52E+04	3.26E+05	**5.15E+03**	5.31E+03	6.96E+04	3.30E+06	3.97E+04	9.68E+04
	30	29	30	28		30	30	30	28
	0	**3**	**0**	**4**	**21**	**0**	**0**	**0**	**2**

Table 18 Ranking of the algorithms in Table 17 according to the Friedman test

	BA	EHO	ES	GA	IMEHO	PBIL	PSO	CCS	VNBA
Friedman rank	7.63	4.23	7.27	2.13	**1.37**	5.33	8.40	4.87	3.77
Final rank	8	4	7	2	**1**	6	9	5	3

Table 19 Worst values for a fixed number of generations on 30D benchmarks

	BA	EHO	ES	GA	IMEHO	PBIL	PSO	CCS	VNBA
F01	5.02E+09	5.57E+08	1.69E+09	5.47E+07	**1.62E+07**	5.95E+08	5.27E+09	2.27E+08	3.70E+08
F02	1.56E+11	4.18E+10	1.06E+11	3.63E+06	**1.55E+04**	4.74E+10	1.14E+11	3.64E+09	2.73E+10
F03	4.97E+05	5.94E+04	3.96E+05	5.12E+04	**1.01E+03**	1.46E+05	1.60E+08	4.45E+05	9.44E+04
F04	4.78E+04	6.44E+03	2.60E+04	6.44E+02	**5.80E+02**	5.38E+03	3.30E+04	8.03E+02	3.01E+03
F05	5.21E+02	5.21E+02	5.21E+02	**5.21E+02**	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02
F06	6.51E+02	6.39E+02	6.44E+02	6.30E+02	**6.17E+02**	6.40E+02	6.53E+02	6.30E+02	6.37E+02
F07	1.90E+03	1.04E+03	1.58E+03	7.03E+02	**7.01E+02**	1.08E+03	2.01E+03	7.29E+02	8.57E+02
F08	1.32E+03	1.09E+03	1.23E+03	8.57E+02	**8.53E+02**	1.12E+03	1.32E+03	1.14E+03	1.02E+03
F09	1.55E+03	1.29E+03	1.45E+03	1.13E+03	**9.62E+02**	1.30E+03	1.52E+03	1.22E+03	1.19E+03
F10	1.09E+04	8.24E+03	9.47E+03	**1.10E+03**	4.53E+03	8.30E+03	9.41E+03	1.04E+04	5.17E+03
F11	1.08E+04	8.48E+03	9.33E+03	5.67E+03	**4.62E+03**	8.71E+03	9.88E+03	1.10E+04	8.56E+03
F12	1.21E+03	1.20E+03	1.20E+03	**1.20E+03**	1.20E+03	1.20E+03	1.20E+03	1.21E+03	1.20E+03
F13	1.31E+03	1.30E+03	1.31E+03	1.30E+03	**1.30E+03**	1.31E+03	1.32E+03	1.30E+03	1.30E+03
F14	1.82E+03	1.53E+03	1.73E+03	1.40E+03	**1.40E+03**	1.53E+03	1.93E+03	1.41E+03	1.48E+03
F15	2.36E+07	5.34E+05	1.19E+07	1.53E+03	**1.51E+03**	1.40E+06	4.65E+06	1.69E+03	7.22E+03
F16	1.61E+03	1.61E+03	1.61E+03	1.61E+03	**1.61E+03**	1.61E+03	1.61E+03	1.61E+03	1.61E+03
F17	9.12E+08	1.35E+07	1.10E+08	1.20E+07	**3.45E+05**	1.59E+07	8.41E+08	2.36E+07	1.34E+07
F18	1.04E+10	5.00E+08	3.74E+09	8.63E+05	**1.44E+04**	1.03E+09	1.75E+10	2.45E+08	3.82E+08
F19	3.70E+03	2.16E+03	2.64E+03	2.02E+03	**1.91E+03**	2.14E+03	3.85E+03	1.95E+03	2.15E+03
F20	4.04E+07	2.53E+04	6.48E+06	1.00E+05	**2.40E+03**	9.50E+04	2.63E+07	4.58E+06	2.93E+04
F21	4.45E+08	3.64E+06	3.22E+07	3.41E+06	**7.46E+04**	6.38E+06	6.74E+08	1.15E+07	6.09E+06
F22	1.13E+05	3.60E+03	5.02E+03	3.22E+03	**2.61E+03**	3.57E+03	2.44E+07	3.84E+03	3.22E+03
F23	4.87E+03	**2.50E+03**	3.59E+03	2.72E+03	2.62E+03	3.01E+03	5.51E+03	2.66E+03	2.74E+03
F24	2.92E+03	**2.60E+03**	2.90E+03	2.66E+03	2.65E+03	2.82E+03	2.76E+03	2.67E+03	2.67E+03
F25	2.93E+03	**2.70E+03**	2.83E+03	2.73E+03	2.71E+03	2.75E+03	2.98E+03	2.73E+03	2.73E+03
F26	2.96E+03	2.71E+03	2.71E+03	2.80E+03	**2.70E+03**	2.70E+03	3.32E+03	2.70E+03	2.71E+03
F27	5.04E+03	**2.90E+03**	4.16E+03	3.81E+03	3.61E+03	4.01E+03	5.94E+03	3.61E+03	4.05E+03
F28	1.09E+04	**3.00E+03**	5.52E+03	5.56E+03	4.69E+03	4.70E+03	1.78E+04	4.53E+03	4.96E+03
F29	3.74E+08	3.97E+05	1.83E+07	9.40E+06	**4.64E+03**	1.59E+07	2.27E+09	3.21E+06	8.72E+06
F30	2.94E+07	1.14E+06	1.25E+06	1.57E+04	**1.10E+04**	3.07E+05	6.87E+07	1.88E+05	6.38E+05
	30	25	30	27		29	30	29	30
	0	**5**	**0**	**3**	**22**	**0**	**0**	**0**	**0**

Table 20 Ranking of the algorithms in Table 19 according to the Friedman test

	BA	EHO	ES	GA	IMEHO	PBIL	PSO	CCS	VNBA
Friedman rank	8.27	3.90	6.77	2.70	**1.33**	5.03	8.30	4.57	4.13
Final rank	8	3	7	2	**1**	6	9	5	4

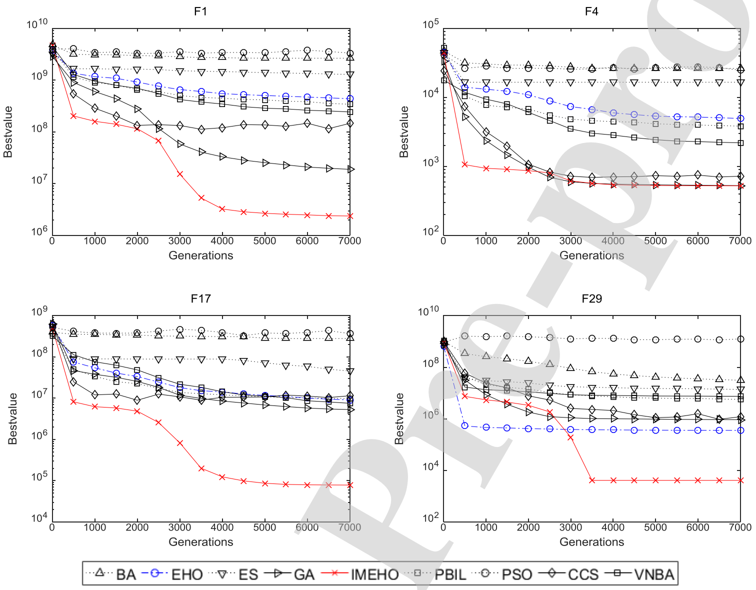


Figure 4 Convergence curve on 30D benchmarks for a fixed number of generations

5.3.2 *D* = 50

In Table 21, the mean values of IMEHO and other algorithms are shown on 50D benchmarks. The last row shows that IMEHO has better mean values on 20 functions, including F01 - F04, F06 - F09, F11, F13 - F22, and F29. For other methods, EHO and GA have better mean values on 5 and 4 functions, which are F23 - F25, F27, F28 and F5, F10, F12, F30, respectively. The penultimate row shows that IMEHO performs better than BA, EHO, ES, GA, PBIL, PSO, CCS, and VNBA on 30, 24, 30, 26, 30, 30, 29, and 30 functions, respectively. Table 22 shows the ranking of the algorithms in Table 21 according to the Friedman test. IMEHO obtains the best rank, GA ranks 2, and the following are EHO, CCS, VNBA, PBIL, ES, BA, and PSO. Table 23 shows the *p*-value obtained by pairwise comparison between IMEHO and other algorithms in Table 21 according to the Friedman test. As seen in this table, the *p*-value obtained by pairwise comparison between IMEHO and all other algorithms is less than 0.05. This indicates that IMEHO is significantly different from other algorithms.

In Table 24, the Std values of IMEHO and other algorithms are shown on 50D benchmarks. The last row shows that IMEHO has better values on 14 functions, including F01 - F03, F07 - F09, F13, F14, F17, F18, F20 - F22, and F29. For other methods, EHO has better values on 6 functions, which are F05, F23 – F25, F28, and F30. ES and GA have better values on 3 and 4 functions, which are F06, F16, F27 and F04, F10, F12, F15, respectively. CCS has better values on F19 and

F26, and VNBA has better values only on F11. The penultimate row shows that IMEHO performs better than BA, EHO, ES, GA, PBIL, PSO, CCS, and VNBA on 25, 19, 22, 23, 23, 26, 20, and 23 functions, respectively. Table 25 shows the ranking of the algorithms in Table 24 according to the Friedman test. IMEHO obtains the best rank, GA ranks 2, and the following are EHO, CCS, PBIL, VNBA, ES, PSO, and BA. In Table 26, the best values of IMEHO and other algorithms are shown on 50D benchmarks. The last row shows that IMEHO has better values on 21 functions, including F01 - F04, F06 - F09, F11, F13 - F22, F26, and F29. For other methods, EHO has better values on 4 functions, which are F23, F24, F27, and F28. GA has better values on 4 functions, which are F05, F10, F12, and F30. VNBA has better values only on F25. The penultimate row shows that IMEHO performs better than BA, EHO, ES, GA, PBIL, PSO, CCS, and VNBA on 30, 25, 30, 26, 29, 30, 30, and 28 functions, respectively. Table 27 shows the ranking of the algorithms in Table 26 according to the Friedman test. IMEHO obtains the best rank, GA ranks 2, and the following are EHO, VNBA, CCS, PBIL, ES, BA, and PSO. In Table 28, the worst values of IMEHO and other algorithms are shown on 50D benchmarks. The last row shows that IMEHO has better values on 21 functions, including F01 - F04, F06 - F09, F11, F13 - F22, F27, and F29. For other methods, EHO has better values on 5 functions, which are F23 – F25, F28, and F30. GA has better values on 3 functions, which are F05, F10, and F12. CCS has better values only on F26. The penultimate row shows that IMEHO performs better than BA, EHO, ES, GA, PBIL, PSO, CCS, and VNBA on 30, 24, 29, 27, 29, 30, 29, and 30 functions, respectively. Table 29 shows the ranking of the algorithms in Table 28 according to the Friedman test. IMEHO obtains the best rank, GA ranks 2, and the following are EHO, CCS, VNBA, PBIL, ES, BA, and PSO.

As can be observed from Tables 21 - 29, the IMEHO algorithm can always find the optimal solution on most problems. Referring to Tables 21, 26, and 28, the IMEHO algorithm can find more than 20 optimal solutions among 30 problems. This implies that the IMEHO algorithm performs well in both the mean, best and worst solution of the problem. It also can be inferred from Table 24 that the IMEHO algorithm has more stable performance than other algorithms. The proposed strategies also perform acceptable on 50D benchmarks. Mutual learning between individuals makes the algorithm more accurate. In addition to learning strategy, the application of elitism strategy ensures that the population is not worse than the original. The separation strategy also makes individuals in the population always better. On the other hand, according to the Friedman rank, the IMEHO algorithm always ranks 1 compared with other algorithms. The *p*-value obtained by pairwise comparison between IMEHO and all other algorithms is less than 0.05 which shows that the IMEHO algorithm has better performance than other algorithms on 50D benchmarks when the maximum number of generations is fixed. The convergence curves of the 4 problems (F01, F04, F17, F29) is shown in Figure 5. As seen, the IMEHO algorithm has a faster rate of convergence than other algorithms and always gets the best solution.

Table 21 Mean values for a fixed number of generations on 50D benchmarks

	BA	EHO	ES	GA	IMEHO	PBIL	PSO	CCS	VNBA
F01	7.83E+09	1.22E+09	4.30E+09	2.22E+07	**6.86E+06**	1.28E+09	1.28E+10	2.77E+08	1.32E+09
F02	2.07E+11	1.17E+11	1.89E+11	7.17E+06	**6.06E+03**	9.50E+10	2.55E+11	6.20E+09	6.54E+10
F03	4.80E+05	8.67E+04	3.71E+05	4.00E+04	**1.08E+03**	2.00E+05	3.69E+06	4.32E+05	1.48E+05
F04	7.77E+04	2.30E+04	6.56E+04	5.83E+02	**5.78E+02**	1.46E+04	7.71E+04	1.06E+03	1.31E+04
F05	5.21E+02	5.21E+02	5.21E+02	**5.21E+02**	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02

F06	6.83E+02	6.67E+02	6.72E+02	6.49E+02	**6.30E+02**	6.70E+02	6.85E+02	6.49E+02	6.66E+02
F07	2.72E+03	1.82E+03	2.54E+03	7.12E+02	**7.00E+02**	1.72E+03	4.23E+03	7.61E+02	1.26E+03
F08	1.62E+03	1.38E+03	1.58E+03	9.16E+02	**8.63E+02**	1.40E+03	1.70E+03	1.32E+03	1.23E+03
F09	1.87E+03	1.58E+03	1.88E+03	1.25E+03	**9.60E+02**	1.66E+03	1.95E+03	1.43E+03	1.47E+03
F10	1.71E+04	1.43E+04	1.60E+04	**1.48E+03**	5.76E+03	1.46E+04	1.52E+04	1.66E+04	9.85E+03
F11	1.72E+04	1.47E+04	1.50E+04	8.30E+03	**7.32E+03**	1.47E+04	1.54E+04	1.62E+04	1.49E+04
F12	1.21E+03	1.20E+03	1.20E+03	**1.20E+03**	1.20E+03	1.20E+03	1.20E+03	1.21E+03	1.20E+03
F13	1.31E+03	1.31E+03	1.31E+03	1.30E+03	**1.30E+03**	1.31E+03	1.31E+03	1.30E+03	1.31E+03
F14	1.93E+03	1.74E+03	1.91E+03	1.41E+03	**1.40E+03**	1.68E+03	2.02E+03	1.42E+03	1.53E+03
F15	4.23E+07	3.16E+06	4.65E+07	1.54E+03	**1.52E+03**	5.41E+06	8.66E+07	1.83E+03	3.24E+05
F16	1.62E+03	1.62E+03	1.62E+03	1.62E+03	**1.62E+03**	1.62E+03	1.62E+03	1.62E+03	1.62E+03
F17	1.11E+09	9.39E+07	3.11E+08	1.22E+07	**3.66E+05**	5.34E+07	2.29E+09	4.10E+07	9.90E+07
F18	2.50E+10	3.97E+09	1.29E+10	1.12E+06	**2.68E+03**	4.31E+09	3.08E+10	3.91E+08	1.85E+09
F19	6.22E+03	2.36E+03	3.80E+03	2.00E+03	**1.96E+03**	2.43E+03	1.07E+04	2.02E+03	2.25E+03
F20	1.41E+07	2.55E+04	3.34E+06	6.29E+04	**2.44E+03**	1.67E+05	1.94E+07	1.61E+06	6.98E+04
F21	2.91E+08	7.32E+06	1.27E+08	9.72E+06	**1.98E+05**	2.47E+07	3.47E+08	2.40E+07	1.16E+08
F22	7.90E+05	6.30E+03	1.29E+04	3.70E+03	**2.87E+03**	5.00E+03	6.61E+06	5.12E+03	4.70E+03
F23	5.42E+03	**2.50E+03**	4.65E+03	2.69E+03	2.65E+03	3.66E+03	6.49E+03	2.74E+03	2.99E+03
F24	3.10E+03	**2.60E+03**	3.17E+03	2.73E+03	2.69E+03	3.06E+03	3.06E+03	2.73E+03	2.75E+03
F25	3.06E+03	**2.70E+03**	3.06E+03	2.74E+03	2.73E+03	2.86E+03	3.08E+03	2.75E+03	2.73E+03
F26	2.97E+03	2.74E+03	2.72E+03	2.79E+03	2.70E+03	2.79E+03	3.34E+03	**2.70E+03**	2.81E+03
F27	6.22E+03	**2.98E+03**	5.04E+03	4.23E+03	3.81E+03	4.80E+03	7.79E+03	4.14E+03	4.90E+03
F28	1.92E+04	**3.00E+03**	9.28E+03	5.12E+03	4.46E+03	5.94E+03	2.38E+04	5.46E+03	7.32E+03
F29	6.19E+07	9.16E+05	1.07E+08	1.69E+04	**4.86E+03**	8.46E+07	7.12E+09	1.75E+07	4.95E+07
F30	4.73E+07	4.24E+04	4.27E+06	**3.60E+04**	4.53E+04	1.18E+06	2.00E+08	2.69E+05	1.70E+06
	30	24	30	26		30	30	29	30
	0	**5**	**0**	**4**	**20**	**0**	**0**	**1**	**0**

Table 22 Ranking of the algorithms in Table 21 according to the Friedman test

	BA	EHO	ES	GA	IMEHO	PBIL	PSO	CCS	VNBA
Friedman rank	7.93	3.80	6.93	2.33	**1.37**	5.17	8.50	4.43	4.53
Final rank	8	3	7	2	**1**	6	9	4	5

Table 23 *p*-value of IMEHO compared with other algorithms in Table 21

	IMEHO-BA	IMEHO-EHO	IMEHO-ES	IMEHO-GA	IMEHO-PBIL	IMEHO-PSO	IMEHO-CCS	IMEHO-VNBA
p-value	4.32E-08	1.00E-03	4.32E-08	5.90E-05	4.32E-08	4.32E-08	3.19E-07	4.32E-08

Table 24 Std values for a fixed number of generations on 50D benchmarks

	BA	EHO	ES	GA	IMEHO	PBIL	PSO	CCS	VNBA
F01	2.27E+09	1.24E+08	6.83E+08	1.11E+07	**7.99E+06**	1.98E+08	3.23E+09	5.31E+07	3.71E+08
F02	3.20E+10	1.85E+10	2.51E+10	8.44E+05	**5.71E+03**	8.74E+09	3.12E+10	9.24E+08	7.11E+09
F03	1.68E+05	4.87E+03	5.67E+04	1.20E+04	**3.78E+02**	2.11E+04	5.40E+06	9.72E+04	4.74E+04
F04	2.20E+04	4.27E+03	1.47E+04	**5.08E+01**	5.10E+01	3.53E+03	1.11E+04	8.48E+01	1.76E+03
F05	5.92E-02	**3.03E-02**	3.09E-02	6.19E-02	3.91E-02	3.90E-02	4.65E-02	4.68E-02	3.79E-02

F06	1.70E+00	1.66E+00	**1.41E+00**	3.41E+00	3.28E+00	1.78E+00	2.68E+00	5.22E+00	2.70E+00
F07	3.05E+02	1.61E+02	2.27E+02	3.38E+00	**3.98E-01**	9.11E+01	4.52E+02	8.55E+00	7.70E+01
F08	5.36E+01	2.85E+01	3.89E+01	1.74E+01	**1.47E+01**	2.72E+01	4.08E+01	2.68E+01	3.15E+01
F09	1.08E+02	2.66E+01	6.02E+01	3.41E+01	**1.92E+01**	3.46E+01	5.40E+01	3.74E+01	2.17E+01
F10	7.93E+02	3.22E+02	6.73E+02	**1.74E+02**	9.12E+02	6.11E+02	3.65E+02	4.91E+02	7.53E+02
F11	6.42E+02	4.75E+02	3.77E+02	1.03E+03	1.22E+03	3.45E+02	4.04E+02	1.17E+03	**3.01E+02**
F12	5.06E-01	3.80E-01	3.24E-01	**2.41E-01**	8.32E-01	3.08E-01	9.06E-01	7.44E-01	3.72E-01
F13	1.17E+00	7.09E-01	6.14E-01	1.66E-01	**7.07E-02**	2.59E-01	7.29E-01	2.29E-01	2.40E-01
F14	8.33E+01	4.44E+01	5.16E+01	5.48E+00	**2.04E-01**	2.48E+01	4.94E+01	2.33E+00	1.60E+01
F15	2.61E+07	1.86E+06	1.85E+07	**8.53E+00**	1.18E+01	2.39E+06	3.98E+07	1.71E+02	1.48E+05
F16	2.75E-01	1.86E-01	**1.26E-01**	3.83E-01	1.05E+00	1.56E-01	1.30E-01	1.29E-01	3.53E-01
F17	3.17E+08	1.46E+07	4.95E+07	6.81E+06	**4.20E+05**	1.33E+07	7.25E+08	1.65E+07	3.61E+07
F18	6.63E+09	5.54E+08	3.36E+09	2.83E+05	**6.48E+02**	8.03E+08	5.96E+09	9.86E+07	6.36E+08
F19	1.62E+03	5.56E+01	4.41E+02	3.39E+01	1.59E+01	6.72E+01	2.78E+03	**9.63E+00**	8.06E+01
F20	1.91E+07	4.48E+03	2.59E+06	1.76E+04	**1.32E+02**	4.47E+04	1.42E+07	1.27E+06	3.39E+04
F21	2.08E+08	1.62E+06	4.30E+07	6.36E+06	**1.88E+05**	4.75E+06	1.30E+08	6.81E+06	5.85E+07
F22	1.00E+06	7.16E+02	5.88E+03	2.87E+02	**2.73E+02**	2.84E+02	3.87E+06	2.85E+02	2.98E+02
F23	5.86E+02	**1.73E-02**	4.18E+02	3.18E+01	2.56E+00	1.38E+02	7.90E+02	1.50E+01	7.79E+01
F24	8.80E+01	**6.28E-03**	4.83E+01	7.71E+00	4.76E+00	2.46E+01	4.44E+01	4.29E+00	2.70E+01
F25	1.01E+02	**3.67E-04**	4.61E+01	8.51E+00	7.88E+00	2.35E+01	9.89E+01	6.88E+00	3.57E+01
F26	2.11E+02	2.36E+01	4.51E-01	4.52E+01	1.84E+01	1.41E+02	2.25E+02	**1.54E-01**	5.24E+01
F27	6.66E+02	4.17E+02	**4.68E+01**	9.28E+01	1.01E+02	4.73E+01	9.74E+02	8.83E+01	4.93E+01
F28	2.55E+03	**2.22E-02**	5.67E+02	4.45E+02	3.56E+02	8.49E+02	1.86E+03	2.46E+02	8.85E+02
F29	5.15E+07	3.38E+04	1.42E+07	4.01E+03	**4.76E+02**	2.54E+07	1.21E+09	2.61E+07	4.25E+06
F30	1.91E+07	**1.50E+03**	1.01E+06	1.88E+04	1.23E+04	4.28E+05	3.27E+07	7.56E+04	5.55E+05
	25	19	22	23		23	26	20	23
	0	**6**	**3**	**4**	**14**	**0**	**0**	**2**	**1**

Table 25 Ranking of the algorithms in Table 24 according to the Friedman test

	BA	EHO	ES	GA	IMEHO	PBIL	PSO	CCS	VNBA
Friedman rank	8.03	3.70	5.67	3.50	**2.97**	4.53	7.67	4.03	4.9
Final rank	9	3	7	2	**1**	5	8	4	6

Table 26 Best values for a fixed number of generations on 50D benchmarks

	BA	EHO	ES	GA	IMEHO	PBIL	PSO	CCS	VNBA
F01	3.20E+09	9.89E+08	3.22E+09	8.74E+06	**2.92E+05**	9.08E+08	6.73E+09	1.81E+08	4.57E+08
F02	1.39E+11	9.08E+10	1.44E+11	6.01E+06	**2.05E+02**	7.92E+10	1.87E+11	4.57E+09	4.75E+10
F03	2.54E+05	7.68E+04	2.53E+05	2.19E+04	**5.87E+02**	1.50E+05	3.12E+05	2.97E+05	9.84E+04
F04	3.76E+04	1.52E+04	3.85E+04	5.11E+02	**4.92E+02**	9.66E+03	5.21E+04	8.22E+02	1.05E+04
F05	5.21E+02	5.21E+02	5.21E+02	**5.21E+02**	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02
F06	6.79E+02	6.64E+02	6.69E+02	6.43E+02	**6.24E+02**	6.66E+02	6.78E+02	6.40E+02	6.60E+02
F07	2.08E+03	1.48E+03	1.84E+03	7.07E+02	**7.00E+02**	1.58E+03	3.39E+03	7.44E+02	1.14E+03
F08	1.53E+03	1.32E+03	1.52E+03	8.82E+02	**8.39E+02**	1.33E+03	1.59E+03	1.27E+03	1.16E+03
F09	1.64E+03	1.52E+03	1.77E+03	1.17E+03	**9.27E+02**	1.57E+03	1.86E+03	1.32E+03	1.44E+03

F10	1.49E+04	1.37E+04	1.39E+04	**1.26E+03**	3.38E+03	1.30E+04	1.47E+04	1.50E+04	8.42E+03
F11	1.53E+04	1.36E+04	1.40E+04	6.48E+03	**5.49E+03**	1.41E+04	1.44E+04	1.35E+04	1.39E+04
F12	1.20E+03	1.20E+03	1.20E+03	**1.20E+03**	1.20E+03	1.20E+03	1.20E+03	1.21E+03	1.20E+03
F13	1.31E+03	1.31E+03	1.31E+03	1.30E+03	**1.30E+03**	1.31E+03	1.31E+03	1.30E+03	1.30E+03
F14	1.80E+03	1.64E+03	1.82E+03	1.40E+03	**1.40E+03**	1.62E+03	1.92E+03	1.41E+03	1.49E+03
F15	9.69E+06	7.83E+05	1.31E+07	1.53E+03	**1.51E+03**	1.83E+06	2.30E+07	1.64E+03	1.33E+05
F16	1.62E+03	1.62E+03	1.62E+03	1.62E+03	**1.62E+03**	1.62E+03	1.62E+03	1.62E+03	1.62E+03
F17	4.13E+08	5.13E+07	2.19E+08	3.99E+06	**2.33E+04**	2.75E+07	4.42E+08	1.36E+07	3.97E+07
F18	1.22E+10	3.05E+09	8.03E+09	6.04E+05	**1.91E+03**	2.55E+09	1.59E+10	1.49E+08	6.81E+08
F19	3.88E+03	2.26E+03	3.19E+03	1.94E+03	**1.92E+03**	2.34E+03	6.30E+03	2.00E+03	2.08E+03
F20	2.67E+05	1.79E+04	4.13E+05	3.54E+04	**2.26E+03**	7.34E+04	2.16E+06	1.11E+05	3.52E+04
F21	2.72E+07	4.56E+06	4.19E+07	3.07E+06	**1.65E+04**	1.45E+07	6.43E+07	1.07E+07	2.86E+07
F22	2.84E+04	5.02E+03	6.03E+03	3.08E+03	**2.37E+03**	4.38E+03	1.07E+06	4.58E+03	3.92E+03
F23	4.26E+03	**2.50E+03**	3.75E+03	2.65E+03	2.65E+03	3.23E+03	4.24E+03	2.71E+03	2.90E+03
F24	2.93E+03	**2.60E+03**	3.00E+03	2.71E+03	2.68E+03	3.00E+03	3.00E+03	2.73E+03	2.63E+03
F25	2.90E+03	2.70E+03	3.00E+03	2.73E+03	2.71E+03	2.82E+03	2.95E+03	2.74E+03	**2.70E+03**
F26	2.72E+03	2.71E+03	2.72E+03	2.70E+03	**2.70E+03**	2.71E+03	3.03E+03	2.70E+03	2.71E+03
F27	5.55E+03	**2.90E+03**	4.89E+03	4.03E+03	3.61E+03	4.71E+03	6.19E+03	3.98E+03	4.81E+03
F28	1.42E+04	**3.00E+03**	8.41E+03	4.66E+03	4.17E+03	4.93E+03	2.04E+04	4.89E+03	6.30E+03
F29	6.15E+06	8.26E+05	7.72E+07	1.25E+04	**3.96E+03**	4.87E+07	4.51E+09	5.15E+06	4.23E+07
F30	1.56E+07	3.86E+04	1.98E+06	**1.72E+04**	2.48E+04	5.72E+05	1.36E+08	1.56E+05	7.58E+05
	30	25	30	26		29	30	30	28
	0	**4**	**0**	**4**	**21**	**0**	**0**	**0**	**1**

Table 27 Ranking of the algorithms in Table 26 according to the Friedman test

	BA	EHO	ES	GA	IMEHO	PBIL	PSO	CCS	VNBA
Friedman rank	7.57	4.17	7.13	2.23	**1.40**	5.23	8.57	4.43	4.27
Final rank	8	3	7	2	**1**	6	9	5	4

Table 28 Worst values for a fixed number of generations on 50D benchmarks

	BA	EHO	ES	GA	IMEHO	PBIL	PSO	CCS	VNBA
F01	1.31E+10	1.49E+09	5.49E+09	5.70E+07	**3.39E+07**	1.70E+09	1.98E+10	4.01E+08	2.02E+09
F02	2.79E+11	1.87E+11	2.42E+11	9.02E+06	**2.16E+04**	1.14E+11	3.02E+11	7.93E+09	7.67E+10
F03	1.01E+06	9.55E+04	4.74E+05	6.61E+04	**2.24E+03**	2.35E+05	2.77E+07	6.90E+05	2.84E+05
F04	1.37E+05	3.31E+04	1.10E+05	7.17E+02	**6.99E+02**	2.66E+04	9.72E+04	1.24E+03	1.77E+04
F05	5.21E+02	5.21E+02	5.21E+02	**5.21E+02**	5.21E+02	5.21E+02	5.21E+02	5.22E+02	5.21E+02
F06	6.87E+02	6.70E+02	6.74E+02	6.56E+02	**6.38E+02**	6.74E+02	6.89E+02	6.66E+02	6.72E+02
F07	3.32E+03	2.12E+03	2.94E+03	7.24E+02	**7.02E+02**	1.88E+03	5.29E+03	7.76E+02	1.47E+03
F08	1.71E+03	1.46E+03	1.68E+03	9.60E+02	**8.93E+02**	1.45E+03	1.76E+03	1.36E+03	1.29E+03
F09	2.07E+03	1.63E+03	2.05E+03	1.33E+03	**1.02E+03**	1.72E+03	2.05E+03	1.52E+03	1.52E+03
F10	1.82E+04	1.49E+04	1.70E+04	**1.88E+03**	7.79E+03	1.59E+04	1.66E+04	1.72E+04	1.12E+04
F11	1.83E+04	1.55E+04	1.55E+04	1.09E+04	**1.02E+04**	1.53E+04	1.61E+04	1.80E+04	1.54E+04
F12	1.21E+03	1.20E+03	1.20E+03	**1.20E+03**	1.20E+03	1.20E+03	1.21E+03	1.21E+03	1.20E+03
F13	1.31E+03	1.31E+03	1.31E+03	1.30E+03	**1.30E+03**	1.31E+03	1.31E+03	1.30E+03	1.31E+03

F14	2.13E+03	1.83E+03	2.05E+03	1.43E+03	**1.40E+03**	1.73E+03	2.12E+03	1.42E+03	1.57E+03
F15	1.14E+08	7.73E+06	9.02E+07	1.57E+03	**1.56E+03**	1.33E+07	1.81E+08	2.33E+03	7.34E+05
F16	1.62E+03	1.62E+03	1.62E+03	1.62E+03	**1.62E+03**	1.62E+03	1.62E+03	1.62E+03	1.62E+03
F17	1.67E+09	1.17E+08	3.85E+08	3.18E+07	**1.53E+06**	8.74E+07	3.87E+09	7.91E+07	1.84E+08
F18	4.48E+10	5.24E+09	2.39E+10	1.69E+06	**4.08E+03**	6.02E+09	4.56E+10	6.47E+08	3.49E+09
F19	1.00E+04	2.47E+03	4.89E+03	2.13E+03	**1.98E+03**	2.62E+03	1.97E+04	2.04E+03	2.43E+03
F20	1.04E+08	3.54E+04	1.03E+07	9.89E+04	**2.78E+03**	2.99E+05	5.39E+07	6.89E+06	1.96E+05
F21	1.12E+09	1.12E+07	2.14E+08	2.95E+07	**8.43E+05**	3.78E+07	6.25E+08	3.78E+07	2.90E+08
F22	4.64E+06	7.94E+03	3.24E+04	4.16E+03	**3.55E+03**	5.50E+03	2.12E+07	5.68E+03	5.34E+03
F23	6.55E+03	**2.50E+03**	5.68E+03	2.78E+03	2.66E+03	3.91E+03	8.33E+03	2.77E+03	3.32E+03
F24	3.29E+03	**2.60E+03**	3.24E+03	2.74E+03	2.70E+03	3.10E+03	3.19E+03	2.74E+03	2.78E+03
F25	3.26E+03	**2.70E+03**	3.16E+03	2.76E+03	2.75E+03	2.90E+03	3.30E+03	2.77E+03	2.82E+03
F26	3.44E+03	2.80E+03	2.72E+03	2.82E+03	2.80E+03	3.08E+03	3.76E+03	**2.70E+03**	2.90E+03
F27	9.17E+03	5.18E+03	5.10E+03	4.42E+03	**4.00E+03**	4.89E+03	1.03E+04	4.38E+03	5.00E+03
F28	2.41E+04	**3.00E+03**	1.07E+04	6.58E+03	5.77E+03	7.90E+03	2.87E+04	5.85E+03	9.43E+03
F29	1.95E+08	9.74E+05	1.34E+08	3.34E+04	**6.01E+03**	1.62E+08	9.21E+09	1.34E+08	5.66E+07
F30	8.67E+07	**4.58E+04**	6.43E+06	9.84E+04	7.35E+04	2.64E+06	2.80E+08	4.30E+05	3.17E+06
	30	24	29	27		29	30	29	30
	0	**5**	**0**	**3**	**21**	**0**	**0**	**1**	**0**

Table 29 Ranking of the algorithms in Table 28 according to the Friedman test

	BA	EHO	ES	GA	IMEHO	PBIL	PSO	CCS	VNBA
Friedman rank	8.27	4.00	6.60	2.43	**1.37**	5.07	8.30	4.37	4.6
Final rank	8	3	7	2	**1**	6	9	4	5

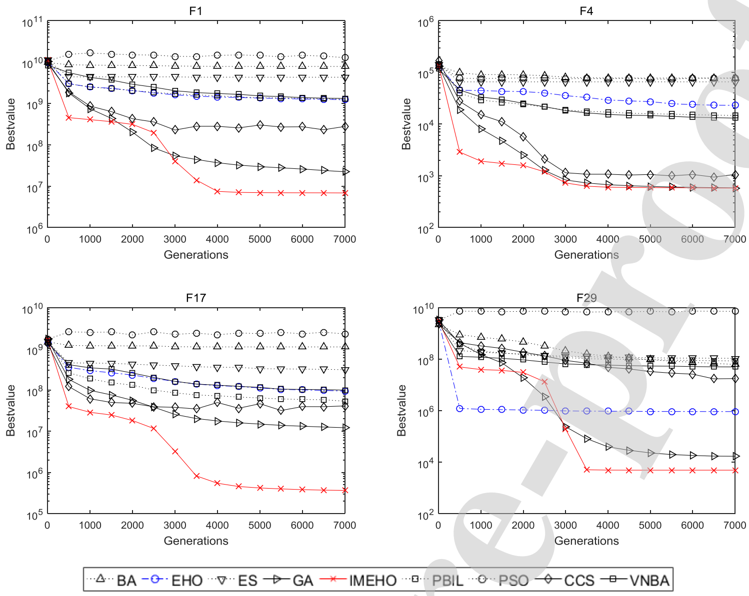


Figure 5 Convergence curve on 50D benchmarks for a fixed number of generations

5.4 Comparison of fixed number of evaluations

In this section, we fixed the maximum number of evaluations to compare IMEHO with other algorithms. We set the maximum number of evaluations (F_{max}) to 5000 * D. Each algorithm still runs 30 times independently on each problem and the results are recorded. Tables 30 - 38 record the results of these algorithms on 30D benchmarks. And Tables 39 - 47 record the results of these algorithms on 50D benchmarks.

5.4.1 *D* = 30

Table 30 shows the mean values of IMEHO algorithm and other algorithms on 30D benchmarks. It can be seen from the last row at the bottom that IMEHO obtains its best results on 20 functions, which are F01 – F04, F06, F7, F09, F11, F13 – F22, F26, and F30. In comparison, EHO performs best on 5 functions, which are F23 – F25, F27, and F28. VNBA performs best on 4 functions, which are F05, F8, F10, and F12. GA performs best only on F29. In penultimate row, IMEHO algorithm performs better than BA, EHO, ES, GA, PBIL, PSO, CCS, and VNBA on 30, 23, 28, 26, 29, 30, 30, and 25 functions, respectively.

Table 31 shows the ranking of the algorithms in Table 30 according to the Friedman test. IMEHO obtains the best rank, VNBA ranks 2, and the following are GA, EHO, CCS, ES, PBIL, BA, and PSO. Table 32 shows the *p*-value obtained by pairwise comparison between IMEHO and other algorithms in Table 30 according to the Friedman test. As can be seen from the table, the *p*-value obtained by pairwise comparison between IMEHO and all other algorithms is less than 0.05. This indicates that IMEHO is significantly different from other algorithms. Table 33 shows the Std values of IMEHO algorithm and other algorithms on 30D benchmarks. It can be seen from the last row at the bottom that IMEHO obtains its best results on 17 functions, which are F01 –

F04, F07, F09, F13 – F15, F17 – F22, F26, and F30. In comparison, EHO performs best on 7 functions, which are F05, F11, F23 – F25, F27, and F28. GA performs best only on F29. PBIL performs best on F06 and F16. VNBA performs best on 3 functions, which are F08, F10, and F12. In penultimate row, IMEHO algorithm performs better than BA, EHO, ES, GA, PBIL, PSO, CCS, and VNBA on 27, 18, 28, 26, 22, 26, 24, and 23 functions, respectively. Besides, Table 34 shows the ranking of the algorithms in Table 33 according to the Friedman test. IMEHO obtains the best rank, EHO ranks 2, and the following are VNBA, CCS, PBIL, GA, ES, BA, and PSO. Table 35 shows the best values of IMEHO algorithm and other algorithms on 30D benchmarks. It can be seen from the last row at the bottom that IMEHO obtains its best results on 21 functions, which are F01 – F04, F06, F07, F09, F11, F13 – F22, F26, F29 and F30. In comparison, EHO performs best on 5 functions, which are F23 – F25, F27, and F28. VNBA performs best on 4 functions, which are F05, F08, F10, and F12. In penultimate row, IMEHO algorithm performs better than BA, EHO, ES, GA, PBIL, PSO, CCS, and VNBA on 30, 25, 29, 28, 30, 30, 30, and 25 functions, respectively. Table 36 shows the ranking of the algorithms in Table 35 according to the Friedman test. IMEHO obtains the best rank, VNBA ranks 2, and the following are GA, EHO, ES, CCS, PBIL, BA, and PSO. Table 37 presents the worst values of IMEHO algorithm and other algorithms on 30D benchmarks. It can be seen from the last row at the bottom that IMEHO obtains its best results on 20 functions, which are F01 – F04, F06, F07, F09, F11, F13 – F22, F26, and F30. In comparison, EHO performs best on 5 functions, which are F23 – F25, F27, and F28. VNBA performs best on 4 functions, which are F05, F08, F10, and F12. GA performs best only on F29. In penultimate row, IMEHO performs better than BA, EHO, ES, GA, PBIL, PSO, CCS, and VNBA on 30, 22, 30, 26, 27, 30, 20, and 25 functions, respectively. Table 38 shows the ranking of the algorithms in Table 37 according to the Friedman test. IMEHO obtains the best rank, VNBA ranks 2, and the following are GA, EHO, CCS, PBIL, ES, BA, and PSO.

It can be seen from Tables 30 - 38, IMEHO algorithm can always find the optimal solution on most problems. As seen in Tables 30, 35, and 37, the IMEHO algorithm can find more than 20 optimal solutions among 30 problems. It also can be observed from Table 33 that the IMEHO algorithm has a more stable performance than the other algorithms. According to the Friedman rank, IMEHO always ranks 1 compared with other algorithms. As seen in Table 32, the *p*-value obtained by pairwise comparison between IMEHO and all other algorithms is less than 0.05. The convergence curves of the 4 problems (F01, F04, F17, F29) is shown in Figure 6. As seen, the IMEHO algorithm has faster rate of convergence than other algorithms and always gets the best solution.

Table 30 Mean values for a fixed number of evaluations on 30D benchmarks

	BA	EHO	ES	GA	IMEHO	PBIL	PSO	CCS	VNBA
F01	2.55E+09	4.98E+08	4.40E+08	3.51E+07	**2.43E+06**	3.67E+08	3.18E+09	1.22E+08	1.22E+08
F02	1.01E+11	3.94E+10	1.95E+10	6.82E+06	**6.10E+03**	4.22E+10	1.03E+11	1.99E+09	2.84E+08
F03	2.53E+05	5.81E+04	1.05E+05	2.12E+04	**9.33E+02**	1.14E+05	1.10E+07	2.50E+05	2.66E+04
F04	2.61E+04	5.62E+03	2.28E+03	5.45E+02	**5.28E+02**	4.36E+03	2.64E+04	7.26E+02	6.35E+02
F05	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02	**5.20E+02**
F06	6.47E+02	6.37E+02	6.32E+02	6.31E+02	**6.13E+02**	6.39E+02	6.49E+02	6.35E+02	6.22E+02
F07	1.66E+03	1.02E+03	8.79E+02	7.09E+02	**7.00E+02**	1.06E+03	1.80E+03	7.18E+02	7.04E+02
F08	1.23E+03	1.08E+03	9.78E+02	9.13E+02	8.35E+02	1.10E+03	1.26E+03	1.08E+03	**8.23E+02**
F09	1.40E+03	1.25E+03	1.13E+03	1.14E+03	**9.30E+02**	1.28E+03	1.45E+03	1.19E+03	1.04E+03

F10	9.90E+03	7.93E+03	4.26E+03	1.56E+03	3.58E+03	7.92E+03	8.76E+03	9.19E+03	**1.21E+03**
F11	1.01E+04	8.21E+03	7.64E+03	5.01E+03	**4.02E+03**	8.36E+03	9.60E+03	1.00E+04	5.01E+03
F12	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.21E+03	**1.20E+03**
F13	1.31E+03	1.30E+03	1.30E+03	1.30E+03	**1.30E+03**	1.30E+03	1.31E+03	1.30E+03	1.30E+03
F14	1.74E+03	1.52E+03	1.45E+03	1.43E+03	**1.40E+03**	1.50E+03	1.81E+03	1.40E+03	1.40E+03
F15	5.29E+06	2.98E+05	8.75E+04	1.53E+03	**1.51E+03**	8.13E+05	1.97E+06	1.56E+03	1.54E+03
F16	1.61E+03	1.61E+03	1.61E+03	1.61E+03	**1.61E+03**	1.61E+03	1.61E+03	1.61E+03	1.61E+03
F17	2.10E+08	1.19E+07	3.78E+07	8.30E+06	**1.97E+05**	1.07E+07	4.15E+08	8.95E+06	1.48E+07
F18	7.95E+09	3.95E+08	8.69E+08	5.20E+05	**5.01E+03**	7.67E+08	1.31E+10	9.49E+07	1.25E+07
F19	2.63E+03	2.15E+03	2.10E+03	1.98E+03	**1.91E+03**	2.11E+03	2.97E+03	1.94E+03	1.96E+03
F20	4.08E+06	2.35E+04	1.82E+05	4.41E+04	**2.33E+03**	4.31E+04	6.51E+06	7.27E+05	3.74E+04
F21	1.20E+08	2.82E+06	1.42E+07	1.38E+06	**5.55E+04**	2.94E+06	2.01E+08	4.43E+06	3.71E+06
F22	5.54E+04	3.37E+03	3.42E+03	2.88E+03	**2.45E+03**	3.32E+03	3.73E+06	3.44E+03	2.94E+03
F23	3.80E+03	**2.50E+03**	2.78E+03	2.67E+03	2.62E+03	2.93E+03	4.05E+03	2.65E+03	2.66E+03
F24	2.85E+03	**2.60E+03**	2.71E+03	2.68E+03	2.64E+03	2.80E+03	2.75E+03	2.66E+03	2.63E+03
F25	2.87E+03	**2.70E+03**	2.75E+03	2.73E+03	2.71E+03	2.75E+03	2.89E+03	2.72E+03	2.72E+03
F26	2.76E+03	2.71E+03	2.74E+03	2.71E+03	**2.70E+03**	2.70E+03	3.03E+03	2.70E+03	2.70E+03
F27	4.62E+03	**2.90E+03**	3.87E+03	3.75E+03	3.23E+03	3.73E+03	5.13E+03	3.39E+03	3.78E+03
F28	9.05E+03	**3.00E+03**	5.09E+03	4.37E+03	3.81E+03	4.21E+03	1.53E+04	4.24E+03	4.50E+03
F29	3.18E+07	3.73E+05	2.49E+07	**8.24E+03**	7.44E+05	5.50E+06	1.20E+09	1.85E+06	3.14E+06
F30	8.92E+06	5.58E+05	3.88E+05	1.23E+04	**7.86E+03**	1.86E+05	2.55E+07	9.04E+04	4.97E+04
	30	23	28	26		29	30	30	25
	0	**5**	**0**	**1**	**20**	**0**	**0**	**0**	**4**

Table 31 Ranking of the algorithms in Table 30 according to the Friedman test

	BA	EHO	ES	GA	IMEHO	PBIL	PSO	CCS	VNBA
Friedman rank	8.13	4.60	5.40	3.10	**1.63**	5.63	8.57	4.97	2.97
Final rank	8	4	6	3	**1**	7	9	5	2

Table 32 *p*-value of IMEHO compared with other algorithms in Table 30

	IMEHO-BA	IMEHO-EHO	IMEHO-ES	IMEHO-GA	IMEHO-PBIL	IMEHO-PSO	IMEHO-CCS	IMEHO-VNBA
p-value	4.32E-08	3.49E-03	2.00E-06	5.90E-05	3.19E-07	4.32E-08	4.32E-08	2.61E-04

Table 33 Std values for a fixed number of evaluations on 30D benchmarks

	BA	EHO	ES	GA	IMEHO	PBIL	PSO	CCS	VNBA
F01	7.75E+08	1.08E+08	1.73E+08	1.71E+07	**3.22E+06**	1.04E+08	9.44E+08	3.62E+07	4.46E+07
F02	1.88E+10	6.39E+09	6.59E+09	2.59E+06	**6.60E+03**	4.59E+09	7.10E+09	2.87E+08	1.25E+08
F03	7.76E+04	5.14E+03	8.41E+04	1.24E+04	**6.34E+02**	1.43E+04	1.43E+07	7.81E+04	1.66E+04
F04	9.43E+03	9.75E+02	9.01E+02	4.00E+01	**3.57E+01**	8.48E+02	4.28E+03	3.63E+01	7.48E+01
F05	6.35E-02	**4.83E-02**	1.33E-01	7.87E-02	5.91E-02	5.04E-02	6.88E-02	9.93E-02	5.48E-02
F06	2.12E+00	1.51E+00	2.57E+00	3.04E+00	2.14E+00	**1.13E+00**	2.17E+00	2.85E+00	2.26E+00
F07	1.85E+02	3.76E+01	4.16E+01	2.34E+00	**2.08E-02**	2.97E+01	1.39E+02	3.21E+00	1.19E+00
F08	3.83E+01	1.61E+01	2.99E+01	1.75E+01	9.22E+00	1.80E+01	2.73E+01	2.00E+01	**3.44E+00**
F09	3.95E+01	2.53E+01	3.13E+01	3.20E+01	**1.10E+01**	2.39E+01	4.29E+01	1.88E+01	2.72E+01

F10	5.64E+02	2.94E+02	6.21E+02	3.86E+02	5.45E+02	3.58E+02	3.57E+02	4.50E+02	**7.90E+01**
F11	4.33E+02	**2.91E+02**	7.84E+02	7.99E+02	5.99E+02	3.61E+02	5.71E+02	4.24E+02	5.95E+02
F12	7.01E-01	2.58E-01	6.21E-01	3.30E-01	5.52E-01	3.94E-01	4.29E-01	8.25E-01	**1.38E-01**
F13	1.05E+00	1.76E-01	5.51E-01	6.26E-01	**4.92E-02**	2.60E-01	1.71E+00	2.01E-01	1.09E-01
F14	6.53E+01	8.68E+00	2.04E+01	1.26E+01	**4.24E-02**	9.94E+00	4.99E+01	1.37E+00	1.65E-01
F15	3.43E+06	9.53E+04	1.02E+05	9.06E+00	**2.25E+00**	3.13E+05	9.06E+05	2.22E+01	2.48E+01
F16	3.26E-01	2.03E-01	4.07E-01	5.14E-01	8.62E-01	**1.73E-01**	2.83E-01	2.10E-01	5.13E-01
F17	1.24E+08	3.42E+06	2.26E+07	5.10E+06	**2.17E+05**	4.05E+06	1.88E+08	3.86E+06	6.08E+06
F18	2.99E+09	1.21E+08	4.40E+08	1.94E+05	**3.92E+03**	2.58E+08	4.20E+09	2.97E+07	6.94E+06
F19	2.64E+02	3.96E+01	5.81E+01	3.60E+01	**2.15E+00**	2.33E+01	4.02E+02	4.66E+00	4.09E+01
F20	5.92E+06	4.55E+03	2.28E+05	2.16E+04	**1.30E+02**	1.67E+04	1.36E+07	6.55E+05	2.15E+04
F21	8.02E+07	1.02E+06	9.32E+06	1.12E+06	**4.32E+04**	9.26E+05	1.01E+08	2.11E+06	1.79E+06
F22	1.19E+05	1.64E+02	2.69E+02	2.51E+02	**1.19E+02**	1.69E+02	4.17E+06	1.77E+02	1.49E+02
F23	4.86E+02	**1.52E-02**	6.61E+01	3.57E+01	4.43E-01	5.19E+01	5.34E+02	7.47E+00	2.91E+01
F24	5.59E+01	**4.62E-03**	1.36E+01	1.12E+01	6.30E+00	1.38E+01	7.42E+00	1.89E+00	1.96E+01
F25	5.21E+01	**1.58E-04**	1.15E+01	5.77E+00	2.76E+00	9.59E+00	6.47E+01	4.98E+00	5.13E+00
F26	8.68E+01	1.32E+00	5.66E+01	3.22E+01	**7.81E-02**	2.33E-01	1.15E+02	1.61E-01	1.15E-01
F27	2.09E+02	**3.46E-03**	7.46E+01	1.62E+02	1.13E+02	2.96E+02	3.82E+02	1.33E+02	6.38E+01
F28	1.40E+03	**4.99E-03**	5.79E+02	2.43E+02	2.14E+02	1.21E+02	2.08E+03	1.19E+02	3.54E+02
F29	4.32E+07	2.17E+04	1.78E+07	**2.28E+03**	4.05E+06	3.23E+06	3.17E+08	3.21E+06	4.15E+06
F30	5.01E+06	5.46E+05	2.42E+05	3.12E+03	**2.15E+03**	6.05E+04	1.61E+07	3.18E+04	2.02E+04
	27	18	28	26		22	26	24	23
	0	**7**	**0**	**1**	**17**	**2**	**0**	**0**	**3**

Table 34 Ranking of the algorithms in Table 33 according to the Friedman test

	BA	EHO	ES	GA	IMEHO	PBIL	PSO	CCS	VNBA
Friedman rank	7.63	3.17	6.83	4.70	**2.53**	4.27	7.73	4.23	3.9
Final rank	8	2	7	6	**1**	5	9	4	3

Table 35 Best values for a fixed number of evaluations on 30D benchmarks

	BA	EHO	ES	GA	IMEHO	PBIL	PSO	CCS	VNBA
F01	1.37E+09	2.70E+08	1.71E+08	1.15E+07	**1.16E+05**	2.15E+08	1.32E+09	5.87E+07	2.08E+07
F02	6.79E+10	3.01E+10	8.68E+09	3.39E+06	**2.00E+02**	3.40E+10	8.49E+10	1.34E+09	1.37E+08
F03	9.14E+04	4.62E+04	2.97E+04	2.31E+03	**3.00E+02**	8.99E+04	1.50E+05	1.02E+05	3.71E+03
F04	9.86E+03	3.91E+03	8.26E+02	4.79E+02	**4.59E+02**	2.74E+03	1.67E+04	6.62E+02	5.33E+02
F05	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02	**5.20E+02**
F06	6.41E+02	6.33E+02	6.26E+02	6.23E+02	**6.09E+02**	6.36E+02	6.42E+02	6.28E+02	6.18E+02
F07	1.35E+03	9.48E+02	7.99E+02	7.06E+02	**7.00E+02**	1.00E+03	1.55E+03	7.13E+02	7.02E+02
F08	1.16E+03	1.05E+03	9.11E+02	8.78E+02	8.18E+02	1.08E+03	1.21E+03	1.03E+03	**8.16E+02**
F09	1.33E+03	1.18E+03	1.07E+03	1.08E+03	**9.10E+02**	1.23E+03	1.38E+03	1.15E+03	9.97E+02
F10	8.91E+03	7.21E+03	3.08E+03	1.11E+03	2.59E+03	7.25E+03	8.20E+03	8.21E+03	**1.08E+03**
F11	9.36E+03	7.33E+03	5.61E+03	3.40E+03	**2.32E+03**	7.61E+03	8.57E+03	9.10E+03	3.85E+03
F12	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	**1.20E+03**
F13	1.31E+03	1.30E+03	1.30E+03	1.30E+03	**1.30E+03**	1.30E+03	1.31E+03	1.30E+03	1.30E+03

F14	1.63E+03	1.50E+03	1.42E+03	1.41E+03	**1.40E+03**	1.49E+03	1.71E+03	1.40E+03	1.40E+03
F15	4.95E+05	1.19E+05	1.09E+04	1.52E+03	**1.50E+03**	2.79E+05	1.07E+06	1.53E+03	1.52E+03
F16	1.61E+03	1.61E+03	1.61E+03	1.61E+03	**1.61E+03**	1.61E+03	1.61E+03	1.61E+03	1.61E+03
F17	2.71E+07	5.44E+06	6.02E+06	1.84E+06	**2.95E+04**	4.29E+06	1.68E+08	2.96E+06	6.16E+06
F18	2.85E+09	1.81E+08	1.69E+08	1.58E+05	**1.86E+03**	2.56E+08	5.22E+09	4.28E+07	9.37E+05
F19	2.16E+03	2.08E+03	1.99E+03	1.92E+03	**1.91E+03**	2.06E+03	2.25E+03	1.93E+03	1.91E+03
F20	2.12E+05	1.54E+04	2.53E+04	1.11E+04	**2.10E+03**	1.59E+04	2.14E+05	7.23E+04	9.08E+03
F21	2.04E+07	1.19E+06	1.70E+06	2.85E+05	**9.36E+03**	8.83E+05	5.37E+07	1.42E+06	6.88E+05
F22	3.84E+03	2.97E+03	2.73E+03	2.42E+03	**2.24E+03**	2.89E+03	1.69E+05	3.15E+03	2.66E+03
F23	2.92E+03	**2.50E+03**	2.68E+03	2.62E+03	2.62E+03	2.82E+03	3.37E+03	2.64E+03	2.62E+03
F24	2.74E+03	**2.60E+03**	2.69E+03	2.66E+03	2.63E+03	2.76E+03	2.73E+03	2.66E+03	2.60E+03
F25	2.78E+03	**2.70E+03**	2.73E+03	2.72E+03	2.71E+03	2.73E+03	2.80E+03	2.72E+03	2.71E+03
F26	2.71E+03	2.71E+03	2.70E+03	2.70E+03	**2.70E+03**	2.70E+03	2.75E+03	2.70E+03	2.70E+03
F27	4.21E+03	**2.90E+03**	3.75E+03	3.19E+03	3.10E+03	3.30E+03	4.60E+03	3.15E+03	3.65E+03
F28	6.05E+03	**3.00E+03**	4.32E+03	4.00E+03	3.62E+03	4.05E+03	1.09E+04	4.09E+03	3.88E+03
F29	4.71E+05	3.28E+05	3.90E+06	5.56E+03	**3.90E+03**	1.29E+06	7.16E+08	2.05E+05	3.28E+04
F30	1.79E+06	2.63E+04	8.70E+04	7.85E+03	**4.47E+03**	9.36E+04	7.85E+06	4.38E+04	1.33E+04
	30	25	29	28		30	30	30	25
	0	**5**	**0**	**0**	**21**	**0**	**0**	**0**	**4**

Table 36 Ranking of the algorithms in Table 35 according to the Friedman test

	BA	EHO	ES	GA	IMEHO	PBIL	PSO	CCS	VNBA
Friedman rank	7.97	5.00	5.10	2.83	**1.43**	6.17	8.63	5.2	2.67
Final rank	8	4	5	3	**1**	7	9	6	2

Table 37 Worst values for a fixed number of evaluations on 30D benchmarks

	BA	EHO	ES	GA	IMEHO	PBIL	PSO	CCS	VNBA
F01	4.90E+09	7.57E+08	8.64E+08	7.06E+07	**1.33E+07**	6.84E+08	5.33E+09	2.10E+08	2.28E+08
F02	1.39E+11	5.52E+10	3.56E+10	1.27E+07	**2.63E+04**	5.19E+10	1.14E+11	2.53E+09	6.88E+08
F03	4.59E+05	6.72E+04	3.77E+05	5.69E+04	**2.89E+03**	1.47E+05	7.32E+07	3.83E+05	6.70E+04
F04	4.36E+04	7.55E+03	4.68E+03	6.45E+02	**5.78E+02**	6.73E+03	3.32E+04	8.09E+02	8.45E+02
F05	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02	**5.21E+02**
F06	6.50E+02	6.41E+02	6.37E+02	6.36E+02	**6.16E+02**	6.41E+02	6.52E+02	6.39E+02	6.28E+02
F07	2.10E+03	1.09E+03	9.79E+02	7.15E+02	**7.00E+02**	1.11E+03	2.10E+03	7.24E+02	7.06E+02
F08	1.33E+03	1.12E+03	1.04E+03	9.48E+02	8.56E+02	1.15E+03	1.33E+03	1.12E+03	**8.30E+02**
F09	1.49E+03	1.28E+03	1.20E+03	1.21E+03	**9.54E+02**	1.35E+03	1.53E+03	1.23E+03	1.12E+03
F10	1.11E+04	8.45E+03	5.59E+03	2.66E+03	4.77E+03	8.60E+03	9.79E+03	1.01E+04	**1.39E+03**
F11	1.09E+04	8.58E+03	9.31E+03	6.09E+03	**5.07E+03**	9.04E+03	1.10E+04	1.08E+04	6.25E+03
F12	1.21E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.21E+03	**1.20E+03**
F13	1.31E+03	1.31E+03	1.30E+03	1.30E+03	**1.30E+03**	1.31E+03	1.32E+03	1.30E+03	1.30E+03
F14	1.85E+03	1.54E+03	1.52E+03	1.46E+03	**1.40E+03**	1.52E+03	1.91E+03	1.41E+03	1.40E+03
F15	1.38E+07	5.28E+05	4.72E+05	1.56E+03	**1.51E+03**	1.54E+06	4.90E+06	1.62E+03	1.62E+03
F16	1.61E+03	1.61E+03	1.61E+03	1.61E+03	**1.61E+03**	1.61E+03	1.61E+03	1.61E+03	1.61E+03
F17	4.71E+08	1.92E+07	1.08E+08	1.96E+07	**9.72E+05**	1.96E+07	8.95E+08	1.97E+07	3.25E+07

F18	1.46E+10	6.70E+08	2.10E+09	9.67E+05	**1.50E+04**	1.34E+09	1.92E+10	1.54E+08	2.56E+07
F19	3.42E+03	2.25E+03	2.22E+03	2.04E+03	**1.92E+03**	2.14E+03	4.00E+03	1.95E+03	2.07E+03
F20	3.07E+07	3.25E+04	9.40E+05	9.69E+04	**2.60E+03**	8.28E+04	6.64E+07	3.02E+06	7.70E+04
F21	3.00E+08	5.49E+06	4.64E+07	5.89E+06	**2.26E+05**	4.87E+06	4.52E+08	1.13E+07	8.37E+06
F22	6.03E+05	3.67E+03	3.76E+03	3.47E+03	**2.65E+03**	3.60E+03	1.96E+07	3.83E+03	3.29E+03
F23	4.88E+03	**2.50E+03**	2.93E+03	2.74E+03	2.62E+03	3.04E+03	5.71E+03	2.66E+03	2.72E+03
F24	2.95E+03	**2.60E+03**	2.74E+03	2.71E+03	2.65E+03	2.83E+03	2.76E+03	2.67E+03	2.66E+03
F25	3.00E+03	**2.70E+03**	2.77E+03	2.74E+03	2.71E+03	2.77E+03	3.08E+03	2.74E+03	2.74E+03
F26	3.04E+03	2.71E+03	2.87E+03	2.81E+03	**2.70E+03**	2.70E+03	3.26E+03	2.70E+03	2.70E+03
F27	5.08E+03	**2.90E+03**	4.02E+03	3.94E+03	3.41E+03	4.04E+03	6.54E+03	3.67E+03	3.89E+03
F28	1.21E+04	**3.00E+03**	6.67E+03	4.95E+03	4.58E+03	4.52E+03	1.90E+04	4.77E+03	5.36E+03
F29	1.72E+08	4.06E+05	9.10E+07	**1.71E+04**	2.22E+07	1.30E+07	1.96E+09	1.80E+07	9.46E+06
F30	2.08E+07	1.47E+06	1.07E+06	2.03E+04	**1.29E+04**	3.77E+05	7.59E+07	1.68E+05	1.04E+05
	30	22	30	26		27	30	29	25
	0	**5**	**0**	**1**	**20**	**0**	**0**	**0**	**4**

Table 38 Ranking of the algorithms in Table 37 according to the Friedman test

	BA	EHO	ES	GA	IMEHO	PBIL	PSO	CCS	VNBA
Friedman rank	8.20	4.30	5.70	3.20	1.70	5.43	8.53	5	2.93
Final rank	8	4	7	3	1	6	9	5	2

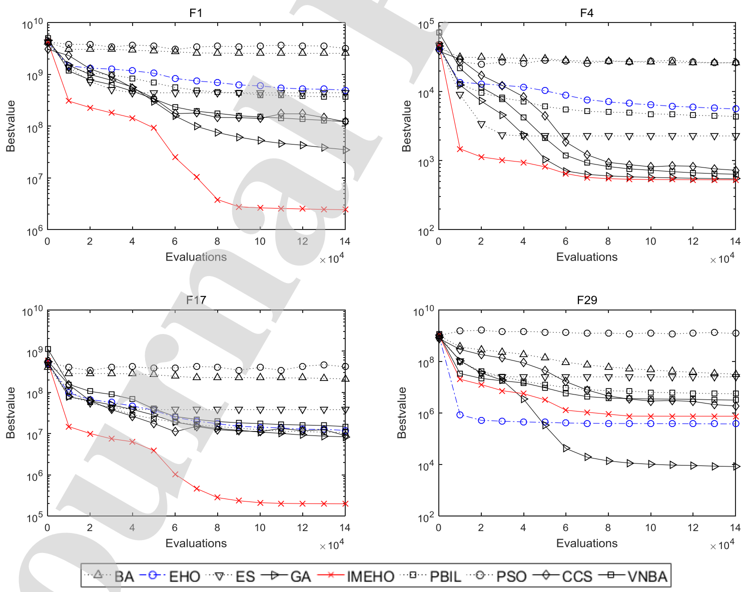


Figure 6 Convergence curve on 30D benchmarks values for a fixed number of evaluations

5.4.2 *D* =50

Table 39 shows the mean values of IMEHO and other algorithms on 50D benchmarks. It can be observed from the last row of this table that the values of IMEHO on 18 functions (F01 - F04, F06, F07, F09, F11, F13 – F15, F17 - F22, and F29) are notably smaller than others. This means that the performance of the proposed IMEHO is acceptable. In addition, EHO performs well on 5 functions (F23 – F25, F27, and F28). VNBA performs well on 5 functions (F05, F08, F10, F12, and F16). GA performs well only on F30. And CCS performs well on F26. The penultimate row shows that IMEHO performs better than BA, EHO, ES, GA, PBIL, PSO, CCS, and VNBA on 30, 23, 28, 26, 29, 30, 29, and 24 functions, respectively. Table 40 shows the ranking of the algorithms in Table 39 according to the Friedman test. IMEHO obtains the best rank, VNBA ranks 2, and the following are GA, EHO, CCS, ES, PBIL, BA, and PSO. In Table 41, the *p*-value obtained by pairwise comparison between IMEHO and other algorithms in Table 39 according to the Friedman test. As can be seen from the table, the *p*-value obtained by pairwise comparison between IMEHO and all other algorithms is less than 0.05. This indicates that IMEHO is significantly different from other algorithms. In Table 42, the Std values of IMEHO and other algorithms are shown on 50D benchmarks. It can be observed from the last row that the values of IMEHO on 14 functions (F01 - F04, F07, F09, F13 - F15, F17, F18, F20, F21, and F29) are much smaller than others, which means that the performance of our proposed IMEHO is steady in convergence process. In addition, EHO performs well on 7 functions (F16, F23 – F25, F27, F28, and F30). PBIL performs well on 4 functions (F5, F6, F11, and F22). CCS performs well on F19 and F26. VNBA performs well on F08, F10, and F12. In penultimate row, it can be seen that IMEHO performs better than BA, EHO, ES, GA, PBIL, PSO, CCS, and VNBA on 25, 18, 28, 22, 22, 24, 22, and 21 functions, respectively.

Table 43 shows the ranking of the algorithms in Table 42 according to the Friedman test. IMEHO obtains the best rank, VNBA ranks 2, and the following are EHO, GA, CCS, PBIL, ES, PSO, and BA. In Table 44, the best values of IMEHO and other algorithms are shown on 50D benchmarks. It can be observed from the last row that the values of IMEHO on 19 functions (F01 - F03, F06, F07, F09, F11, F13 - F22, F26, and F29) are much smaller than others, which means that the performance of our proposed IMEHO is good. In addition, EHO performs well on 5 functions (F23 – F25, F27, and F28). GA performs well on F04 and F30. VNBA performs well on 4 functions (F05, F08, F10, and F12). In penultimate row, it can be seen that IMEHO performs better than BA, EHO, ES, GA, PBIL, PSO, CCS, and VNBA on 30, 25, 29, 25, 30, 30, 30, and 25 functions, respectively. Table 45 shows the ranking of the algorithms in Table 44 according to the Friedman test. IMEHO obtains the best rank, VNBA ranks 2, and the following are GA, EHO, ES, CCS, PBIL, BA, and PSO. In Table 46, the worst values of IMEHO and other algorithms are shown on 50D benchmarks. It can be observed from the last row that the values of IMEHO on 16 functions (F01 - F03, F06, F07, F09, F13 – F15, F17 – F22, and F29) are much smaller than others. In addition, EHO performs well on 6 functions (F23 – F25, F27, F28, and F30). GA performs well on F04. CCS performs well on F26. And VNBA performs well on 6 functions (F05, F08, F10 – F12, and F16). The penultimate row shows that IMEHO performs better than BA, EHO, ES, GA, PBIL, PSO, CCS, and VNBA on 30, 22, 30, 25, 30, 30, 29, and 23 functions, respectively. Table 47 shows the ranking of the algorithms in Table 46 according to the Friedman test. IMEHO obtains the best rank, VNBA ranks 2, and the following are GA, EHO, CCS, ES, PBIL, BA, and PSO. It can be seen in Tables 39 – 47 that the IMEHO algorithm can always find the optimal solution on most problems. Although IMEHO does not perform as well as before on this set of problems, it is still better than other algorithms. It can be seen in Table 42 that the IMEHO algorithm has more stable performance than other algorithms. According to the Friedman rank, the IMEHO algorithm always ranks 1 compared with other algorithms. In addition, in Table 41, all *p*-values are less than 0.05, which indicates that IMEHO is significantly different from other algorithms. These results suggest that IMEHO algorithm has better performance than other algorithms on 50D benchmarks when the maximum number of evaluations is fixed. The convergence curves of the 4 problems (F01, F04, F17, F29) is given in Figure 7. As seen, the IMEHO algorithm has faster rate of convergence than other algorithms and always gets the best solution.

Table 39 Mean values for a fixed number of evaluations on 50D benchmarks

	BA	EHO	ES	GA	IMEHO	PBIL	PSO	CCS	VNBA
F01	8.09E+09	1.19E+09	7.82E+08	2.29E+07	**7.58E+06**	1.39E+09	1.49E+10	2.03E+08	1.29E+08
F02	2.16E+11	1.24E+11	4.61E+10	8.83E+06	**9.64E+04**	1.01E+11	2.48E+11	4.46E+09	5.69E+08
F03	4.68E+05	8.86E+04	2.47E+05	4.15E+04	**2.05E+03**	2.10E+05	5.05E+06	4.75E+05	4.80E+04
F04	7.53E+04	2.46E+04	7.48E+03	5.83E+02	**5.81E+02**	1.49E+04	7.93E+04	9.39E+02	7.79E+02
F05	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02	**5.21E+02**
F06	6.83E+02	6.67E+02	6.60E+02	6.54E+02	**6.28E+02**	6.70E+02	6.85E+02	6.54E+02	6.40E+02
F07	2.73E+03	1.90E+03	1.15E+03	7.24E+02	**7.00E+02**	1.73E+03	4.13E+03	7.45E+02	7.06E+02
F08	1.63E+03	1.37E+03	1.13E+03	9.83E+02	8.74E+02	1.39E+03	1.68E+03	1.29E+03	**8.37E+02**
F09	1.88E+03	1.59E+03	1.37E+03	1.31E+03	**9.58E+02**	1.66E+03	1.95E+03	1.43E+03	1.16E+03
F10	1.69E+04	1.43E+04	7.54E+03	2.82E+03	5.75E+03	1.45E+04	1.53E+04	1.63E+04	**1.41E+03**
F11	1.73E+04	1.46E+04	1.39E+04	9.26E+03	**7.71E+03**	1.49E+04	1.55E+04	1.68E+04	8.04E+03
F12	1.21E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.21E+03	**1.20E+03**
F13	1.31E+03	1.31E+03	1.30E+03	1.30E+03	**1.30E+03**	1.31E+03	1.31E+03	1.30E+03	1.30E+03
F14	1.91E+03	1.74E+03	1.53E+03	1.44E+03	**1.40E+03**	1.68E+03	2.02E+03	1.41E+03	1.40E+03
F15	4.60E+07	3.44E+06	6.07E+05	1.54E+03	**1.52E+03**	5.35E+06	8.47E+07	1.72E+03	1.60E+03
F16	1.62E+03	1.62E+03	1.62E+03	1.62E+03	1.62E+03	1.62E+03	1.62E+03	1.62E+03	**1.62E+03**
F17	1.10E+09	9.93E+07	1.19E+08	1.31E+07	**3.46E+05**	6.64E+07	2.45E+09	3.09E+07	3.85E+07
F18	2.66E+10	4.37E+09	3.41E+09	1.01E+06	**2.61E+03**	4.68E+09	2.89E+10	2.57E+08	4.99E+07
F19	6.85E+03	2.41E+03	2.29E+03	2.01E+03	**1.96E+03**	2.44E+03	9.90E+03	2.00E+03	1.99E+03
F20	1.42E+07	2.68E+04	4.02E+05	7.97E+04	**2.64E+03**	1.85E+05	2.27E+07	1.30E+06	5.61E+04
F21	3.81E+08	7.69E+06	5.27E+07	1.04E+07	**1.66E+05**	2.54E+07	3.59E+08	1.51E+07	2.03E+07
F22	7.05E+05	6.52E+03	5.41E+03	3.66E+03	**2.96E+03**	5.07E+03	8.59E+06	4.95E+03	4.11E+03
F23	5.18E+03	**2.50E+03**	3.15E+03	2.69E+03	2.65E+03	3.72E+03	6.25E+03	2.72E+03	2.70E+03
F24	3.10E+03	**2.60E+03**	2.85E+03	2.76E+03	2.69E+03	3.06E+03	3.07E+03	2.73E+03	2.69E+03
F25	3.08E+03	**2.70E+03**	2.82E+03	2.74E+03	2.73E+03	2.86E+03	3.10E+03	2.75E+03	2.73E+03
F26	2.90E+03	2.74E+03	2.84E+03	2.85E+03	2.70E+03	2.75E+03	3.48E+03	**2.70E+03**	2.78E+03
F27	6.30E+03	**2.90E+03**	4.52E+03	4.38E+03	3.80E+03	4.80E+03	7.68E+03	4.26E+03	4.27E+03
F28	1.91E+04	**3.00E+03**	7.78E+03	5.30E+03	4.68E+03	5.97E+03	2.45E+04	5.60E+03	5.29E+03
F29	7.16E+07	9.32E+05	1.63E+08	1.63E+06	**4.68E+03**	8.32E+07	6.46E+09	5.91E+07	3.18E+07
F30	4.39E+07	4.25E+04	1.53E+06	**3.44E+04**	4.77E+04	1.03E+06	1.84E+08	2.81E+05	7.22E+04
	30	23	28	26		29	30	29	24
	0	**5**	**0**	**1**	**18**	**0**	**0**	**1**	**5**

Table 40 Ranking of the algorithms in Table 39 according to the Friedman test

	BA	EHO	ES	GA	IMEHO	PBIL	PSO	CCS	VNBA
Friedman rank	8.03	4.50	5.37	3.07	**1.70**	6.10	8.60	4.93	2.7
Final rank	8	4	6	3	**1**	7	9	5	2

Table 41 *p*-value of IMEHO compared with other algorithms in Table 39

	IMEHO-BA	IMEHO-EHO	IMEHO-ES	IMEHO-GA	IMEHO-PBIL	IMEHO-PSO	IMEHO-CCS	IMEHO-VNBA
p-value	4.32E-08	3.49E-03	2.00E-06	5.90E-05	3.19E-07	4.32E-08	3.19E-07	1.02E-03

Table 42 Std values for a fixed number of evaluations on 50D benchmarks

	BA	EHO	ES	GA	IMEHO	PBIL	PSO	CCS	VNBA
F01	2.77E+09	1.51E+08	2.78E+08	9.38E+06	**7.65E+06**	2.04E+08	3.89E+09	4.12E+07	4.37E+07
F02	4.07E+10	2.17E+10	1.12E+10	1.78E+06	**3.74E+05**	8.99E+09	2.07E+10	7.29E+08	1.69E+08
F03	1.23E+05	5.84E+03	6.32E+04	1.27E+04	**1.08E+03**	2.01E+04	8.34E+06	9.64E+04	1.27E+04
F04	1.55E+04	4.68E+03	2.63E+03	4.74E+01	**3.95E+01**	2.26E+03	1.09E+04	7.81E+01	9.34E+01
F05	3.68E-02	3.53E-02	1.30E-01	6.08E-02	4.39E-02	**3.31E-02**	3.39E-02	7.71E-02	5.76E-02
F06	2.67E+00	2.49E+00	5.23E+00	3.23E+00	3.37E+00	**1.59E+00**	2.42E+00	7.55E+00	4.48E+00
F07	2.72E+02	2.19E+02	1.10E+02	8.94E+00	**5.54E-02**	1.12E+02	5.36E+02	6.59E+00	1.55E+00
F08	5.17E+01	2.80E+01	4.04E+01	3.13E+01	1.94E+01	3.36E+01	4.07E+01	4.05E+01	**5.12E+00**
F09	7.89E+01	3.33E+01	4.44E+01	3.90E+01	**2.23E+01**	3.38E+01	6.66E+01	3.39E+01	3.38E+01
F10	6.10E+02	5.09E+02	7.92E+02	4.90E+02	7.75E+02	4.53E+02	5.54E+02	5.89E+02	**1.21E+02**
F11	4.38E+02	4.22E+02	1.27E+03	9.09E+02	1.07E+03	**3.36E+02**	3.73E+02	6.40E+02	6.43E+02
F12	7.06E-01	3.18E-01	9.76E-01	2.11E-01	6.47E-01	3.92E-01	6.18E-01	8.52E-01	**1.67E-01**
F13	1.13E+00	5.60E-01	5.72E-01	5.36E-01	**8.88E-02**	3.15E-01	5.94E-01	1.64E-01	1.25E-01
F14	7.70E+01	4.59E+01	2.87E+01	1.03E+01	**2.22E-01**	2.33E+01	7.06E+01	2.72E+00	3.00E-01
F15	3.09E+07	2.49E+06	4.47E+05	1.02E+01	**9.75E+00**	2.05E+06	3.35E+07	1.22E+02	1.04E+02
F16	2.89E-01	**1.93E-01**	6.84E-01	4.60E-01	1.14E+00	2.02E-01	2.21E-01	2.90E-01	4.16E-01
F17	3.78E+08	2.22E+07	5.64E+07	5.96E+06	**3.89E+05**	1.67E+07	8.69E+08	1.10E+07	1.76E+07
F18	7.14E+09	4.25E+08	1.74E+09	2.23E+05	**5.54E+02**	9.45E+08	6.68E+09	5.34E+07	2.98E+07
F19	1.50E+03	7.19E+01	1.45E+02	3.54E+01	2.21E+01	7.52E+01	1.88E+03	**6.87E+00**	1.72E+01
F20	1.91E+07	4.59E+03	3.07E+05	3.66E+04	**3.31E+02**	5.14E+04	2.88E+07	8.50E+05	1.55E+04
F21	2.29E+08	1.55E+06	2.97E+07	5.93E+06	**7.94E+04**	7.71E+06	1.46E+08	5.58E+06	6.83E+06
F22	1.23E+06	7.99E+02	2.31E+03	3.64E+02	2.40E+02	**2.25E+02**	5.11E+06	2.98E+02	3.69E+02
F23	5.11E+02	**1.99E-02**	1.83E+02	2.46E+01	2.52E+00	1.23E+02	9.56E+02	1.25E+01	2.40E+01
F24	9.34E+01	**9.65E-03**	2.35E+01	1.04E+01	5.95E+00	2.30E+01	4.50E+01	4.78E+00	3.75E+00
F25	8.63E+01	**3.84E-04**	2.40E+01	9.21E+00	5.90E+00	2.36E+01	5.49E+01	8.46E+00	5.80E+00
F26	2.16E+02	2.42E+01	8.35E+01	9.29E+01	1.82E+01	1.12E+02	2.15E+02	**2.02E-01**	4.52E+01
F27	4.90E+02	**2.81E-03**	1.01E+02	1.03E+02	1.10E+02	6.08E+01	8.71E+02	9.94E+01	9.70E+01
F28	2.47E+03	**2.14E-02**	7.66E+02	5.32E+02	5.32E+02	8.84E+02	2.68E+03	3.08E+02	6.04E+02
F29	7.17E+07	2.11E+04	7.96E+07	8.83E+06	**4.80E+02**	2.50E+07	1.47E+09	7.67E+07	1.43E+07
F30	1.59E+07	**9.30E+02**	6.95E+05	1.34E+04	1.38E+04	3.31E+05	4.70E+07	9.68E+04	3.32E+04
	25	18	28	22		22	24	22	21
	0	**7**	**0**	**0**	**14**	**4**	**0**	**2**	**3**

Table 43 Ranking of the algorithms in Table 42 according to the Friedman test

	BA	EHO	ES	GA	IMEHO	PBIL	PSO	CCS	VNBA
Friedman rank	7.70	3.60	6.87	4.03	**2.93**	4.47	7.43	4.43	3.53
Final rank	9	3	7	4	**1**	6	8	5	2

Table 44 Best values for a fixed number of evaluations on 50D benchmarks

	BA	EHO	ES	GA	IMEHO	PBIL	PSO	CCS	VNBA
F01	2.56E+09	8.75E+08	3.36E+08	9.43E+06	**7.85E+05**	9.03E+08	8.72E+09	1.28E+08	4.33E+07
F02	1.36E+11	9.16E+10	2.77E+10	6.51E+06	**2.00E+02**	8.67E+10	2.05E+11	2.87E+09	3.30E+08
F03	2.95E+05	7.88E+04	1.33E+05	1.64E+04	**5.32E+02**	1.76E+05	2.30E+05	3.45E+05	2.19E+04
F04	5.21E+04	1.90E+04	3.30E+03	**5.12E+02**	5.20E+02	1.04E+04	5.89E+04	7.94E+02	6.11E+02
F05	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02	**5.20E+02**
F06	6.75E+02	6.62E+02	6.50E+02	6.48E+02	**6.22E+02**	6.67E+02	6.80E+02	6.40E+02	6.33E+02
F07	2.32E+03	1.56E+03	9.44E+02	7.10E+02	**7.00E+02**	1.47E+03	2.97E+03	7.29E+02	7.04E+02
F08	1.54E+03	1.32E+03	1.07E+03	9.34E+02	8.31E+02	1.29E+03	1.60E+03	1.21E+03	**8.26E+02**
F09	1.74E+03	1.53E+03	1.29E+03	1.23E+03	**9.22E+02**	1.59E+03	1.75E+03	1.35E+03	1.10E+03
F10	1.57E+04	1.29E+04	5.45E+03	2.01E+03	4.29E+03	1.33E+04	1.43E+04	1.51E+04	**1.22E+03**
F11	1.64E+04	1.37E+04	1.10E+04	7.69E+03	**5.71E+03**	1.43E+04	1.44E+04	1.52E+04	6.35E+03
F12	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	**1.20E+03**
F13	1.31E+03	1.31E+03	1.30E+03	1.30E+03	**1.30E+03**	1.31E+03	1.31E+03	1.30E+03	1.30E+03
F14	1.75E+03	1.67E+03	1.49E+03	1.42E+03	**1.40E+03**	1.63E+03	1.92E+03	1.40E+03	1.40E+03
F15	5.97E+06	9.16E+05	8.04E+04	1.52E+03	**1.51E+03**	2.51E+06	2.24E+07	1.57E+03	1.55E+03
F16	1.62E+03	1.62E+03	1.62E+03	1.62E+03	**1.62E+03**	1.62E+03	1.62E+03	1.62E+03	1.62E+03
F17	4.04E+08	5.01E+07	3.77E+07	3.24E+06	**4.22E+04**	3.99E+07	1.07E+09	1.31E+07	1.57E+07
F18	1.53E+10	3.35E+09	7.31E+08	5.37E+05	**2.02E+03**	2.21E+09	1.59E+10	1.70E+08	1.20E+07
F19	4.40E+03	2.31E+03	2.07E+03	1.95E+03	**1.91E+03**	2.29E+03	6.40E+03	1.99E+03	1.94E+03
F20	1.94E+05	1.78E+04	8.96E+04	2.15E+04	**2.26E+03**	9.59E+04	4.13E+06	9.65E+04	3.34E+04
F21	6.61E+07	4.51E+06	2.02E+07	2.35E+06	**4.03E+04**	8.61E+06	7.00E+07	7.29E+06	8.58E+06
F22	1.73E+04	4.98E+03	4.12E+03	3.04E+03	**2.37E+03**	4.48E+03	1.60E+06	4.49E+03	3.12E+03
F23	4.09E+03	**2.50E+03**	2.90E+03	2.65E+03	2.65E+03	3.43E+03	4.93E+03	2.70E+03	2.67E+03
F24	2.94E+03	**2.60E+03**	2.80E+03	2.75E+03	2.68E+03	2.99E+03	3.00E+03	2.71E+03	2.68E+03
F25	2.90E+03	**2.70E+03**	2.77E+03	2.72E+03	2.72E+03	2.78E+03	2.98E+03	2.74E+03	2.72E+03
F26	2.72E+03	2.71E+03	2.71E+03	2.70E+03	**2.70E+03**	2.71E+03	3.12E+03	2.70E+03	2.70E+03
F27	5.49E+03	**2.90E+03**	4.36E+03	4.09E+03	3.58E+03	4.67E+03	6.25E+03	4.09E+03	4.05E+03
F28	1.40E+04	**3.00E+03**	6.38E+03	4.72E+03	4.20E+03	5.15E+03	1.89E+04	5.08E+03	4.58E+03
F29	6.15E+06	8.91E+05	4.98E+07	1.43E+04	**3.86E+03**	4.50E+07	2.97E+09	2.65E+06	1.68E+05
F30	1.82E+07	4.08E+04	3.81E+05	**2.00E+04**	3.11E+04	5.07E+05	9.22E+07	1.29E+05	2.91E+04
	30	25	29	25		30	30	30	25
	0	**5**	**0**	**2**	**19**	**0**	**0**	**0**	**4**

Table 45 Ranking of the algorithms in Table 44 according to the Friedman test

	BA	EHO	ES	GA	IMEHO	PBIL	PSO	CCS	VNBA
Friedman rank	8.03	4.97	5.10	2.80	**1.53**	6.30	8.57	5.13	2.57
Final rank	8	4	5	3	**1**	7	9	6	2

Table 46 Worst values for a fixed number of evaluations on 50D benchmarks

	BA	EHO	ES	GA	IMEHO	PBIL	PSO	CCS	VNBA
F01	1.29E+10	1.55E+09	1.48E+09	4.76E+07	**2.71E+07**	1.85E+09	2.48E+10	2.93E+08	2.40E+08
F02	3.36E+11	1.79E+11	7.29E+10	1.28E+07	**1.89E+06**	1.27E+11	3.00E+11	5.99E+09	9.53E+08
F03	8.64E+05	9.95E+04	3.69E+05	7.02E+04	**4.97E+03**	2.56E+05	4.35E+07	6.57E+05	7.99E+04
F04	1.22E+05	3.91E+04	1.54E+04	**6.89E+02**	7.14E+02	2.06E+04	1.03E+05	1.09E+03	1.01E+03
F05	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.21E+02	5.22E+02	**5.21E+02**
F06	6.86E+02	6.71E+02	6.68E+02	6.61E+02	**6.36E+02**	6.73E+02	6.89E+02	6.65E+02	6.49E+02
F07	3.36E+03	2.27E+03	1.48E+03	7.49E+02	**7.00E+02**	1.91E+03	4.98E+03	7.54E+02	7.10E+02
F08	1.76E+03	1.42E+03	1.22E+03	1.09E+03	9.12E+02	1.46E+03	1.75E+03	1.38E+03	**8.45E+02**
F09	2.05E+03	1.65E+03	1.44E+03	1.39E+03	**1.03E+03**	1.73E+03	2.06E+03	1.48E+03	1.22E+03
F10	1.80E+04	1.50E+04	8.80E+03	3.91E+03	7.41E+03	1.54E+04	1.75E+04	1.74E+04	**1.70E+03**
F11	1.82E+04	1.58E+04	1.61E+04	1.08E+04	1.07E+04	1.57E+04	1.61E+04	1.79E+04	**9.10E+03**
F12	1.21E+03	1.20E+03	1.21E+03	1.20E+03	1.20E+03	1.20E+03	1.21E+03	1.21E+03	**1.20E+03**
F13	1.31E+03	1.31E+03	1.31E+03	1.30E+03	**1.30E+03**	1.31E+03	1.31E+03	1.30E+03	1.30E+03
F14	2.07E+03	1.82E+03	1.60E+03	1.47E+03	**1.40E+03**	1.73E+03	2.14E+03	1.42E+03	1.40E+03
F15	1.17E+08	1.15E+07	1.60E+06	1.57E+03	**1.55E+03**	1.06E+07	1.51E+08	2.16E+03	2.05E+03
F16	1.62E+03	1.62E+03	1.62E+03	1.62E+03	1.62E+03	1.62E+03	1.62E+03	1.62E+03	**1.62E+03**
F17	2.09E+09	1.36E+08	2.23E+08	2.77E+07	**1.63E+06**	1.01E+08	5.14E+09	5.66E+07	8.53E+07
F18	4.80E+10	4.97E+09	9.94E+09	1.33E+06	**3.92E+03**	6.30E+09	4.13E+10	3.37E+08	1.36E+08
F19	9.85E+03	2.57E+03	2.78E+03	2.07E+03	**1.99E+03**	2.63E+03	1.37E+04	2.02E+03	2.01E+03
F20	9.16E+07	3.60E+04	1.13E+06	1.91E+05	**3.72E+03**	3.21E+05	1.51E+08	3.37E+06	8.87E+04
F21	1.13E+09	1.22E+07	1.67E+08	2.23E+07	**3.63E+05**	4.18E+07	7.09E+08	2.81E+07	3.58E+07
F22	6.45E+06	8.52E+03	1.63E+04	4.41E+03	**3.46E+03**	5.48E+03	2.08E+07	5.53E+03	4.79E+03
F23	6.51E+03	**2.50E+03**	3.68E+03	2.75E+03	2.66E+03	3.96E+03	9.63E+03	2.75E+03	2.76E+03
F24	3.30E+03	**2.60E+03**	2.90E+03	2.79E+03	2.71E+03	3.10E+03	3.20E+03	2.73E+03	2.70E+03
F25	3.27E+03	**2.70E+03**	2.87E+03	2.77E+03	2.74E+03	2.90E+03	3.20E+03	2.77E+03	2.75E+03
F26	3.24E+03	2.80E+03	3.05E+03	3.05E+03	2.80E+03	3.08E+03	3.85E+03	**2.70E+03**	2.81E+03
F27	7.40E+03	**2.90E+03**	4.71E+03	4.62E+03	4.00E+03	4.90E+03	9.71E+03	4.49E+03	4.44E+03
F28	2.22E+04	**3.00E+03**	9.47E+03	6.88E+03	6.34E+03	8.28E+03	2.84E+04	6.36E+03	7.11E+03
F29	3.53E+08	9.67E+05	4.17E+08	4.84E+07	**5.79E+03**	1.74E+08	8.81E+09	2.36E+08	4.14E+07
F30	7.92E+07	**4.44E+04**	3.20E+06	8.15E+04	8.65E+04	1.78E+06	2.66E+08	6.24E+05	1.45E+05
	30	22	30	25		30	30	29	23
	0	**6**	**0**	**1**	**16**	**0**	**0**	**1**	**6**

Table 47 Ranking of the algorithms in Table 46 according to the Friedman test

	BA	EHO	ES	GA	IMEHO	PBIL	PSO	CCS	VNBA
Friedman rank	8.27	4.23	5.87	3.03	**1.70**	5.97	8.37	4.9	2.67
Final rank	8	4	6	3	**1**	7	9	5	2

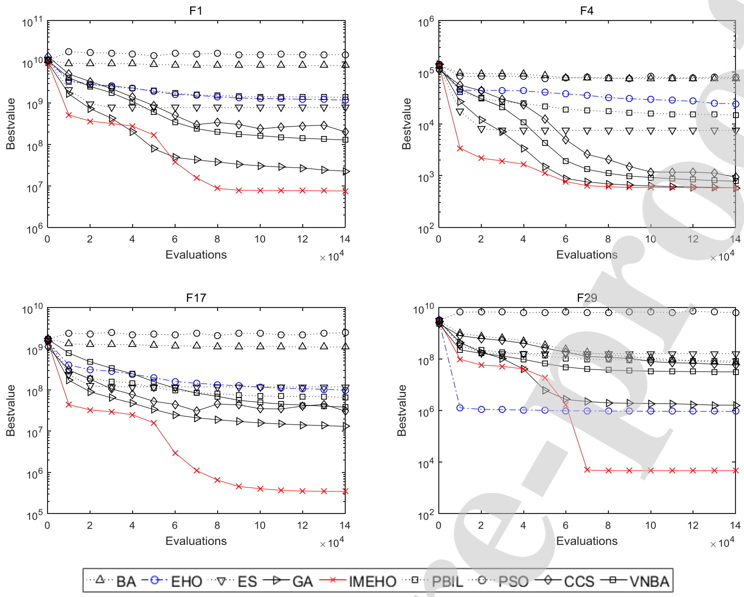


Figure 7 Convergence curve on 50D benchmarks values for a fixed number of evaluations

5.5 Discussion

For further analysis of the IMEHO algorithm performance, the corresponding mean values given in Tables 12, 21 30, and 39 are selected and presented in Table 48. The best values are marked in **bold**. The column of Group represents the grouping of all problems. The column of *T*-30D, *T*-50D, *F*-30D, *F*-50D represent the mean values returned by the IMEHO algorithm in Tables 12, 21, 30, and 39. The column of Rate represents the ratio between the optimal solutions obtained and all the problems in this group. As seen in Table 48, IMEHO performs well in solving unimodal functions, simple multimodal functions and hybrid functions. For the first group of unimodal functions and the third group of hybrid functions, IMEHO can always find the optimal solution. For the second group of simple multimodal functions, the algorithm can find the optimal solution on most problems, and the ratio of finding the optimal solution is 69%. On the other hand, the IMEHO algorithm does not seem to be suitable for solving composition functions, and the ratio is only 22%. It seems that the IMEHO performs best in solving unimodal and hybrid functions. Referring to the second group of simple multimodal functions, IMEHO can find the optimal solution on most problems. This implies that the algorithm has better performance in solving such problems.

The topology of the EHO is relatively decentralized because it is treated on a clan basis. The proposed learning strategy makes the topology of IMEHO more compact, especially for good individuals. They can exchange information better, which makes it easier for algorithms to find better solutions. On the other hand, the application of elitism strategy and separation strategy have led to a better evolutionary direction for the population. For composition functions, IMEHO performs poorly. According to the "no free lunch" theorem [91]. "Any improvement in performance on one type of problem is offset by the performance of another type of problem."

Adjusting IMEHO to get better composition function performance comes at a price, and cost is a slow convergence of other problems. Therefore, one may not expect to get the best performance on all functions because the proposed IMEHO focuses on improving the performance of EHO on most functions.

Table 48 The statistical analysis of the IMEHO algorithm

Group	Problem	*T*-30D	*T*-50D	*F*-30D	*F*-50D	Rate
unimodal functions	F01	**2.38E+06**	**6.86E+06**	**2.43E+06**	**7.58E+06**	100%
	F02	**5.69E+03**	**6.06E+03**	**6.10E+03**	**9.64E+04**	
	F03	**4.41E+02**	**1.08E+03**	**9.33E+02**	**2.05E+03**	
simple multimodal functions	F04	**5.24E+02**	**5.78E+02**	**5.28E+02**	**5.81E+02**	69%
	F05	5.21E+02	5.21E+02	5.21E+02	5.21E+02	
	F06	**6.12E+02**	**6.30E+02**	**6.13E+02**	**6.28E+02**	
	F07	**7.00E+02**	**7.00E+02**	**7.00E+02**	**7.00E+02**	
	F08	**8.33E+02**	**8.63E+02**	8.35E+02	8.74E+02	
	F09	**9.32E+02**	**9.60E+02**	**9.30E+02**	**9.58E+02**	
	F10	3.26E+03	5.76E+03	3.58E+03	5.75E+03	
	F11	3.96E+03	**7.32E+03**	4.02E+03	**7.71E+03**	
	F12	1.20E+03	1.20E+03	1.20E+03	1.20E+03	
	F13	**1.30E+03**	**1.30E+03**	**1.30E+03**	**1.30E+03**	
	F14	**1.40E+03**	**1.40E+03**	**1.40E+03**	**1.40E+03**	
	F15	**1.50E+03**	**1.52E+03**	**1.51E+03**	**1.52E+03**	
	F16	**1.61E+03**	**1.62E+03**	**1.61E+03**	1.62E+03	
hybrid functions	F17	**7.86E+04**	**3.66E+05**	**1.97E+05**	**3.46E+05**	100%
	F18	**5.10E+03**	**2.68E+03**	**5.01E+03**	**2.61E+03**	
	F19	**1.91E+03**	**1.96E+03**	**1.91E+03**	**1.96E+03**	
	F20	**2.21E+03**	**2.44E+03**	**2.33E+03**	**2.64E+03**	
	F21	**2.93E+04**	**1.98E+05**	**5.55E+04**	**1.66E+05**	
	F22	**2.41E+03**	**2.87E+03**	**2.45E+03**	**2.96E+03**	
Composition functions	F23	2.62E+03	2.65E+03	2.62E+03	2.65E+03	22%
	F24	2.64E+03	2.69E+03	2.64E+03	2.69E+03	
	F25	2.71E+03	2.73E+03	2.71E+03	2.73E+03	
	F26	**2.70E+03**	2.70E+03	**2.70E+03**	2.70E+03	
	F27	3.28E+03	3.81E+03	3.23E+03	3.80E+03	
	F28	3.77E+03	4.46E+03	3.81E+03	4.68E+03	
	F29	**4.11E+03**	**4.86E+03**	7.44E+05	**4.68E+03**	
	F30	**7.08E+03**	4.53E+04	**7.86E+03**	4.77E+04	

6. Conclusions

In this paper, several strategies are proposed to improve the performance of the standard EHO algorithm. The movement of elephants is simulated, and a velocity strategy is developed to assign an initial velocity to each elephant. A new learning strategy is offered for the clan updating operator, in which each elephant in the herd has a learning goal through three different situations. Moreover, a new separation strategy is established that adds an evaluation process. We also applied the elitism strategy to the entire herd. The influence of the parameters and strategies on the proposed IMEHO algorithm is studied to ensure its better performance. Finally, a comprehensive comparative study is carried out to benchmark the performance of the IMEHO algorithm against a number of robust optimization algorithms. The results proved that the proposed IMEHO algorithm has a better performance than the standard EHO and other algorithms. Despite the acceptable performance of the IMEHO algorithm, some concerns need to be addressed in future research. First, the influence of other parameters such as the size of acceleration coefficient on the overall performance of the algorithm should be studied. The IMEHO algorithm was evaluated on the single objective benchmark functions. Therefore, further research should be done on the suitability of the IMEHO algorithm for solving multi-objective problems. Future research can also focus on deploying the IMEHO algorithm to solve some other problems such as Optimal power flow (OPF) problem [34] and economic emission dispatch problem [23].

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 41576011, No. U1706218, No. 41706010, and No. 61503165,).

References

[1] E. Tuba, R. Capor-Hrosik, A. Alihodzic, R. Jovanovic, M. Tuba, Chaotic elephant herding optimization algorithm, 2018 IEEE 16th World Symposium on Applied Machine Intelligence and Informatics (SAMI), IEEE, 2018, pp. 213-216.

[2] G. Beni, J. Wang, Swarm intelligence in cellular robotic systems, Robots and biological systems: towards a new bionics?, Springer, 1993, pp. 703-712.

[3] J. Kennedy, R. Eberhart, Particle swarm optimization (PSO), Proc. IEEE International Conference on Neural Networks, Perth, Australia, 1995, pp. 1942-1948.

[4] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents, IEEE Trans Syst Man Cybern B Cybern 1996, 26(1) 29-41.

[5] Q. Yang, W.-N. Chen, Z. Yu, T. Gu, Y. Li, H. Zhang, J. Zhang, Adaptive multimodal continuous ant colony optimization, IEEE Trans Evolut Comput 2017, 21(2) 191-205.

[6] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J Global Optim 2007, 39(3) 459-471.

[7] X.-S. Yang, S. Deb, Cuckoo search via Lévy flights, 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), IEEE, 2009, pp. 210-214.

[8] X.-S. Yang, A new metaheuristic bat-inspired algorithm, Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), Springer, 2010, pp. 65-74.

[9] X.-S. Yang, Firefly algorithm, stochastic test functions and design optimisation, Int J Bio-Inspir Comput 2010, 2(2) 78-84.

[10] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv Eng Softw 2014, 69 46-61.

[11] R. Tang, S. Fong, X.-S. Yang, S. Deb, Wolf search algorithm with ephemeral memory, Seventh International Conference on Digital Information Management (ICDIM 2012),

IEEE, 2012, pp. 165-172.
[12] S. Mirjalili, The ant lion optimizer, Adv Eng Softw 2015, 83 80-98.
[13] A. Kaveh, S. Talatahari, A novel heuristic optimization method: charged system search, Acta Mech 2010, 213(3-4) 267-289.
[14] G.-G. Wang, S. Deb, Z. Cui, Monarch butterfly optimization, Neural Comput Appl 2019, 31(7) 1995-2014.
[15] A.H. Gandomi, A.H. Alavi, Krill herd: a new bio-inspired optimization algorithm, Commun Nonlinear Sci Numer Simulat 2012, 17(12) 4831-4845.
[16] G.-G. Wang, S. Deb, L.D.S. Coelho, Earthworm optimization algorithm: a bio-inspired metaheuristic algorithm for global optimization problems, Int J Bio-Inspir Comput 2018, 12(1) 1-22.
[17] E. Cuevas, A. González, D. Zaldívar, M. Pérez-Cisneros, An optimisation algorithm based on the behaviour of locust swarms, Int J Bio-Inspir Comput 2015, 7(6) 402-407.
[18] G.-G. Wang, Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems, Memetic Computing 2018, 10(2) 151-164.
[19] S. Mirjalili, S.M. Mirjalili, A. Hatamlou, Multi-verse optimizer: a nature-inspired algorithm for global optimization, Neural Comput Appl 2016, 27(2) 495-513.
[20] S. Mirjalili, Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, Neural Comput Appl 2016, 27(4) 1053-1073.
[21] G.-G. Wang, S. Deb, L.d.S. Coelho, Elephant herding optimization, 2015 3rd International Symposium on Computational and Business Intelligence (ISCBI), IEEE, 2015, pp. 1-5.
[22] G.-G. Wang, S. Deb, X.-Z. Gao, L.D.S. Coelho, A new metaheuristic optimisation algorithm motivated by elephant herding behaviour, Int J Bio-Inspir Comput 2016, 8(6) 394-409.
[23] H. Narimani, S.-E. Razavi, A. Azizivahed, E. Naderi, M. Fathi, M.H. Ataei, M.R. Narimani, A multi-objective framework for multi-area economic emission dispatch, Energy 2018, 154 126-142.
[24] H.-Y. Sang, Q.-K. Pan, P.-Y. Duan, J.-Q. Li, An effective discrete invasive weed optimization algorithm for lot-streaming flowshop scheduling problems, J Intell Manuf 2018, 29(6) 1337-1349.
[25] H.-Y. Sang, P.-Y. Duan, J.-Q. Li, An effective invasive weed optimization algorithm for scheduling semiconductor final testing problem, Swarm Evol Comput 2018, 38 42-53.
[26] H.-Y. Sang, Q.-K. Pan, J.-Q. Li, P. Wang, Y.-Y. Han, K.-Z. Gao, P. Duan, Effective invasive weed optimization algorithms for distributed assembly permutation flowshop problem with total flowtime criterion, Swarm Evol Comput 2019, 44 64-73.
[27] W. Luo, R. Yi, B. Yang, P. Xu, Surrogate-assisted evolutionary framework for data-driven dynamic optimization, IEEE Trans Emerg Top Comput Intell 2018, 3(2) 137-150.
[28] C. Bu, W. Luo, L. Yue, Continuous dynamic constrained optimization with ensemble of locating and tracking feasible regions strategies, IEEE Trans Evolut Comput 2016, 21(1) 14-33.
[29] W. Luo, J. Sun, C. Bu, H. Liang, Species-based particle swarm optimizer enhanced by memory for dynamic optimization, Appl Soft Compt 2016, 47 130-140.

[30] H. Duan, W. Zhao, G. Wang, X. Feng, Test-sheet composition using analytic hierarchy process and hybrid metaheuristic algorithm TS/BBO, Math Probl Eng 2012, 2012.

[31] G. Wang, L. Guo, H. Duan, L. Liu, H. Wang, M. Shao, Path planning for uninhabited combat aerial vehicle using hybrid meta-heuristic DE/BBO algorithm, Advanced Science, Engineering and Medicine 2012, 4(6) 550-564.

[32] J.-H. Yi, S. Deb, J. Dong, A.H. Alavi, G.-G. Wang, An improved NSGA-III Algorithm with adaptive mutation operator for big data optimization problems, Future Gener Comp Sy 2018, 88 571-585.

[33] J.-H. Yi, J. Wang, G.-G. Wang, Improved probabilistic neural networks with self-adaptive strategies for transformer fault diagnosis problem, Adv Mech Eng 2016, 8(1) 1687814015624832.

[34] E. Naderi, M. Pourakbari-Kasmaei, H. Abdi, An efficient particle swarm optimization algorithm to solve optimal power flow problem integrated with FACTS devices, Appl Soft Compt 2019, 80 243-262.

[35] Y. Zhang, D.-w. Gong, X.-z. Gao, T. Tian, X.-y. Sun, Binary differential evolution with self-learning for multi-objective feature selection, Inf Sci 2020, 507 67-85.

[36] Y. Zhang, S. Cheng, Y. Shi, D.-w. Gong, X. Zhao, Cost-sensitive feature selection using two-archive multi-objective artificial bee colony algorithm, Expert Syst Appl 2019, 137 46-58.

[37] J.-Q. Li, Q.-K. Pan, M.F. Tasgetiren, A discrete artificial bee colony algorithm for the multi-objective flexible job-shop scheduling problem with maintenance activities, Appl Math Model 2014, 38(3) 1111-1132.

[38] J.-q. Li, Q.-k. Pan, Y.-C. Liang, An effective hybrid tabu search algorithm for multi-objective flexible job-shop scheduling problems, Comput Ind Eng 2010, 59(4) 647-662.

[39] Y.-Y. Han, D. Gong, X. Sun, A discrete artificial bee colony algorithm incorporating differential evolution for the flow-shop scheduling problem with blocking, Eng Optimiz 2015, 47(7) 927-946.

[40] M. Li, D. Xiao, Y. Zhang, H. Nan, Reversible data hiding in encrypted images using cross division and additive homomorphism, Signal Processing: Image Communication 2015, 39 234-248.

[41] M. Li, Y. Guo, J. Huang, Y. Li, Cryptanalysis of a chaotic image encryption scheme based on permutation-diffusion structure, Signal processing: Image communication 2018, 62 164-172.

[42] H. Fan, M. Li, D. Liu, E. Zhang, Cryptanalysis of a colour image encryption using chaotic APFM nonlinear adaptive filter, Signal Process 2018, 143 28-41.

[43] Y. Zhang, D. Gong, Y. Hu, W. Zhang, Feature selection algorithm based on bare bones particle swarm optimization, Neurocomputing 2015, 148 150-157.

[44] Y. Zhang, X.-f. Song, D.-w. Gong, A return-cost-based binary firefly algorithm for feature selection, Inf Sci 2017, 418 561-574.

[45] W. Mao, J. He, J. Tang, Y. Li, Predicting remaining useful life of rolling bearings based on deep feature representation and long short-term memory neural network, Adv Mech Eng 2018, 10(12) 1-18.

[46] M. Jian, K.-M. Lam, J. Dong, Facial-feature detection and localization based on a

hierarchical scheme, Inf Sci 2014, 262 1-14.
[47] G.-G. Wang, H.E. Chu, S. Mirjalili, Three-dimensional path planning for UCAV using an improved bat algorithm, Aerosp Sci Technol 2016, 49 231-238.
[48] G.-G. Wang, X. Cai, Z. Cui, G. Min, J. Chen, High performance computing for cyber physical social systems by using evolutionary multi-objective optimization algorithm, IEEE Transactions on Emerging Topics in Computing 2017, in press. doi: 10.1109/TETC.2017.2703784.
[49] M. Jian, K.-M. Lam, J. Dong, Illumination-insensitive texture discrimination based on illumination compensation and enhancement, Inf Sci 2014, 269 60-72.
[50] M. Jian, K.-M. Lam, J. Dong, L. Shen, Visual-patch-attention-aware saliency detection, IEEE Trans Cybern 2014, 45(8) 1575-1586.
[51] G.-G. Wang, M. Lu, Y.-Q. Dong, X.-J. Zhao, Self-adaptive extreme learning machine, Neural Comput Appl 2016, 27(2) 291-303.
[52] W. Mao, Y. Zheng, X. Mu, J. Zhao, Uncertainty evaluation and model selection of extreme learning machine based on Riemannian metric, Neural Comput Appl 2014, 24(7-8) 1613-1625.
[53] G. Liu, J. Zou, Level set evolution with sparsity constraint for object extraction, IET Image Processing 2018, 12(8) 1413-1422.
[54] R.P. Parouha, K.N. Das, Economic load dispatch using memory based differential evolution, Int J Bio-Inspir Comput 2018, 11(3) 159-170.
[55] G.-G. Wang, A.H. Gandomi, X. Zhao, H.C.E. Chu, Hybridizing harmony search algorithm with cuckoo search for global numerical optimization, Soft Comput 2016, 20(1) 273-285.
[56] X. Zhao, A perturbed particle swarm algorithm for numerical optimization, Appl Soft Compt 2010, 10(1) 119-124.
[57] G. Wu, X. Shen, H. Li, H. Chen, A. Lin, P.N. Suganthan, Ensemble of differential evolution variants, Inf Sci 2018, 423 172-186.
[58] G.-G. Wang, L. Guo, A.H. Gandomi, G.-S. Hao, H. Wang, Chaotic krill herd algorithm, Inf Sci 2014, 274 17-34.
[59] H. Wang, J.-H. Yi, An improved optimization method based on krill herd and artificial bee colony with information exchange, Memetic Computing 2018, 10(2) 177-198.
[60] J. Sun, Z. Miao, D. Gong, X.-J. Zeng, J. Li, G. Wang, Interval multiobjective optimization with memetic algorithms, IEEE Trans Cybern 2019.
[61] G.-G. Wang, Y. Tan, Improving metaheuristic algorithms with information feedback models, IEEE Trans Cybern 2017, 49(2) 542-555.
[62] Z.-M. Gu, G.-G. Wang, Improving NSGA-III algorithms with information feedback models for large-scale many-objective optimization, Future Gener Comp Sy 2020, 107 49-69.
[63] Y.-H. Feng, G.-G. Wang, Binary moth search algorithm for discounted {0-1} knapsack problem, IEEE Access 2018, 6 10708-10719.
[64] M. Abdel-Basset, Y. Zhou, An elite opposition-flower pollination algorithm for a 0-1 knapsack problem, Int J Bio-Inspir Comput 2018, 11(1) 46-53.
[65] W. Mao, J. He, Y. Li, Y. Yan, Bearing fault diagnosis with auto-encoder extreme learning machine: A comparative study, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 2017, 231(8) 1560-1578.

[66] W. Mao, W. Feng, X. Liang, A novel deep output kernel learning method for bearing fault structural diagnosis, Mech Syst Signal Pr 2019, 117 293-318.
[67] J. Knowles, D. Corne, The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation, Congress on Evolutionary Computation (CEC99), Vol. 1, 1999, pp. 98-105.
[68] D. Whitley, A genetic algorithm tutorial, Stat Comput 1994, 4(2) 65-85.
[69] S. Baluja, Population-based incremental learning. a method for integrating genetic search based function optimization and competitive learning, Technical report CMU-CS-94-163 Carnegie Mellon University (1994).
[70] G.-G. Wang, S. Deb, A.H. Gandomi, Z. Zhang, A.H. Alavi, Chaotic cuckoo search, Soft Comput 2016, 20(9) 3349-3362.
[71] G.-G. Wang, M. Lu, X.-J. Zhao, An improved bat algorithm with variable neighborhood search for global optimization, 2016 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2016, pp. 1773-1778.
[72] S. Gupta, V. Singh, S. Singh, T. Prakash, N. Rathore, Elephant herding optimization based PID controller tuning, Int J Adv Technol Eng Exploration 2016, 3(24) 194.
[73] I. Strumberger, N. Bacanin, S. Tomic, M. Beko, M. Tuba, Static drone placement by elephant herding optimization algorithm, 2017 25th Telecommunication Forum (Telfor), IEEE, 2017, pp. 808-811.
[74] A. Alihodzic, E. Tuba, R. Capor-Hrosik, E. Dolicanin, M. Tuba, Unmanned aerial vehicle path planning problem by adjusted elephant herding optimization, 2017 25th Telecommunication Forum (Telfor), IEEE, 2017, pp. 804-807.
[75] N.K. Meena, S. Parashar, A. Swarnkar, N. Gupta, K.R. Niazi, Improved elephant herding optimization for multiobjective DER accommodation in distribution systems, Ind Inform 2018, 14(3) 1029-1039.
[76] M.A. Elhosseini, R.A. El Sehiemy, Y.I. Rashwan, X. Gao, On the performance improvement of elephant herding optimization algorithm, Knowl-Based Syst 2019, 166 58-70.
[77] J. Li, L. Guo, Y. Li, C. Liu, Enhancing elephant herding optimization with novel individual updating strategies for large-scale optimization problems, Mathematics 2019, 7(5) 395.
[78] J. Chen, H. Xu, H. Lv, Crossover Particle Swarm Optimization with Incremental Learning, J Inf Comput. 10(6) 1759-1766.
[79] W. Ye, W. Feng, S. Fan, A novel multi-swarm particle swarm optimization with dynamic learning strategy, Appl Soft Compt 2017, 61 832-843.
[80] W.-F. Gao, L.-L. Huang, S.-Y. Liu, C. Dai, Artificial bee colony algorithm based on information learning, IEEE Trans Cybern 2015, 45(12) 2827-2839.
[81] B. Anari, J. Akbari Torkestani, A.M. Rahmani, A learning automata- based clustering algorithm using ant swarm intelligence, Expert Syst 2018, 35(6) e12310.
[82] W. Chung, J. Kim, H. Lee, E. Kim, General dimensional multiple-output support vector regressions and their multiple kernel learning, IEEE Trans Cybern 2015, 45(11) 2572-2584.
[83] T. Zhu, Y. Hao, W. Luo, H. Ning, Learning enhanced differential evolution for tracking optimal decisions in dynamic power systems, Appl Soft Compt 2018, 67 812-821.

[84] E. Ahmadi, B. Goldengorin, G.A. Süer, H. Mosadegh, A hybrid method of 2-TSP and novel learning-based GA for job sequencing and tool switching problem, Appl Soft Compt 2018, 65 214-229.

[85] K. Tomas, P. Michal, V. Adam, S. Roman, Orthogonal learning firefly algorithm, International Conference on Hybrid Artificial Intelligence Systems, Springer, 2018, pp. 315-326.

[86] P. Rakshit, A. Konar, Realization of learning induced self-adaptive sampling in noisy optimization, Appl Soft Compt 2018, 69 288-315.

[87] F. Wang, H. Zhang, K. Li, Z. Lin, J. Yang, X.-L. Shen, A hybrid particle swarm optimization algorithm using adaptive learning strategy, Inf Sci 2018, 436 162-177.

[88] J. Liang, B. Qu, P. Suganthan, Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore 2013.

[89] X. Chen, H. Tianfield, C. Mei, W. Du, G. Liu, Biogeography-based learning particle swarm optimization, Soft Comput 2017, 21(24) 7519-7541.

[90] J.J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE Trans Evolut Comput 2006, 10(3) 281-295.

[91] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans Evolut Comput 1997, 1(1) 67-82.

***conflict of Interest Statement**

Dear Editor,

All the authors declare that there is no conflict of interests regarding the publication of this article.

Once again, thank you for your help in processing our manuscript.

Yours sincerely,

Wei Li
Department of Computer Science and Technology, Ocean University of China, 266100 Qingdao, China

Gai-Ge Wang
Department of Computer Science and Technology, Ocean University of China, 266100 Qingdao, China

Amir H. Alavi
Department of Civil and Environmental Engineering, University of Pittsburgh, Pittsburgh, PA 15261, USA
Department of Bioengineering, University of Pittsburgh, Pittsburgh, PA 15261, USA
Department of Computer Science and Information Engineering, Asia University, Taichung, Taiwan

*Credit Author Statement

February 13, 2020

Dear Professor

It is a great pleasure for us to submit a research paper entitled:

"*Learning-based elephant herding optimization algorithm for solving numerical optimization problems*"

to ***Knowledge-Based Systems*** for possible publication. This article has not been published previously and it is not under consideration for publication elsewhere. This publication is approved by all authors and if accepted, it will not be published elsewhere in the same form, in English or in any other language, without the written consent of the copyright-holder.

Credit Author Statement

Wei Li: Conceptualization, Methodology, Software, Visualization, Investigation;
Gai-Ge Wang: Supervision, Validation, Data curation, Reviewing and Editing, Writing-Original draft preparation;
Amir H. Alavi: Visualization, Reviewing and Editing.

Best regards,

Gai-Ge Wang et al.