



# An improved Harris Hawks optimizer combined with extremal optimization

Hai-Lin Zhang<sup>1</sup> · Min-Rong Chen<sup>1</sup> · Pei-Shan Li<sup>2</sup> · Jun-Jie Huang<sup>3</sup>

Received: 25 April 2022 / Accepted: 11 September 2022 / Published online: 25 October 2022  
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2022

## Abstract

Harris Hawks optimizer (HHO) is a new swarm intelligence optimization algorithm proposed in recent years. It seeks the optimal solution by simulating the predation strategy of Harris hawks and many previous experiments show that HHO has a good effect on solving optimization problems. However, HHO also has the shortcomings of low convergence accuracy and easy to fall into local optimum. In order to improve the performance of HHO, an improved HHO hybridized with extremal optimization (IHHO-EO) is proposed. Aiming at the defect of insufficient information utilization and excessive randomization in the exploration phase of the algorithm, the own historical optimal position of Harris hawks is introduced to better guide the individuals to search for better positions and improve the global search ability. Secondly, a nonlinear prey energy escaping factor is proposed to better balance the exploration and exploitation phases. Thirdly, refracted opposition-based learning (ROBL) with a dynamic parameter is proposed and combined with HHO, which can improve the quality of solutions and convergence speed. Finally, the exploitation ability is improved by performing EO operation which has strong local search ability. The proposed algorithm is applied to 23 classical benchmark test functions and 29 CEC2017 test functions. IHHO-EO is compared with HHO, other newly proposed optimization algorithms and some improved variants of HHO. The experimental results verify the effectiveness of the added strategies. In addition, the proposed approach is applied to solving the pressure vessel design problem. The results show that IHHO-EO has an excellent performance in terms of accuracy, reliability and statistical tests.

**Keywords** Optimization algorithm · Harris Hawks optimizer · Refracted opposition-based learning · Extremal optimization

## 1 Introduction

Optimization is the search process for the best solution to a problem, i.e. to find the maximum or minimum value of an objective function. Optimization problems exist widely in engineering design, medicine, scientific research, economic management and other fields [34]. It is of great help to deal

with optimization problems effectively in all fields. However, some traditional mathematical optimization methods, such as quasi-Newton process, conjugate gradient, strong steepest and sequential quadratic computing, are complex and highly restricted so they cannot solve various complex and non-differentiable optimization problems effectively [42, 49, 64].

Meta-heuristic algorithms are able to provide a useful and elegant solution to those optimization problems [16] due to their merits such as no limitation to specific problems, without needing gradient information and so on. Meta-heuristic algorithms can be generally divided into evolution-based methods, swarm intelligence-based methods and physics-based methods. The evolution-based algorithms are inspired by the theory of biological evolution, starting with the initial solutions and continuously obtaining higher quality solutions through evolution from generation to generation. The most famous meta-heuristic algorithm based on evolution is genetic algorithm (GA) [31] which

✉ Min-Rong Chen  
mrongchen@126.com

✉ Pei-Shan Li  
sli14@foxmail.com

<sup>1</sup> School of Computer Science, South China Normal University, Guangzhou 510631, China

<sup>2</sup> College of Cyber Security and the National Joint Engineering Research Center of Network Security Detection and Protection Technology, Jinan University, Guangzhou 510632, China

<sup>3</sup> College of Cyber Security, Jinan University, Guangzhou 510632, China

simulates Darwin's biological model of natural selection and genetic mechanism, evolving a new generation of better individuals through parental chromosome selection, crossover, and mutation. Other algorithms based on evolution include differential evolution (DE) [57], biogeography-based optimization (BBO) [56] and so on. In recent years, many new meta-heuristic optimization algorithms based on evolution have been proposed. [28] proposed an evolutionary algorithm based on ring theory (RTEA) and applied it to the combinatorial optimization problems. [27] proposed an optimization algorithm based on group theory (GTOA), which was applied to a variety of knapsack problems. [62] proposed an improved GTOA algorithm and a new operator to enhance the interpretability of group theory in discounted 0–1 knapsack problem. [29] proposed the Taper-shaped transfer functions, and on this basis, a binary differential evolution algorithm based on the Taper-shaped transfer functions (T-NBDE) was proposed to solve the knapsack problem with a single continuous variable and the uncapacitated facility location problem. Physics-based algorithms are to simulate some physical phenomena in nature, such as simulated annealing algorithm (SA) [60] which simulates the annealing phenomenon of objects, gravitational search algorithm (GSA) [52] based on universal gravitation and Newton's second law. Swarm intelligence-based algorithms are to simulate the behavior of groups in nature and seek optimal solutions through the wisdom of groups, such as particle swarm optimizer (PSO) [37], firefly algorithm (FA) [67], butterfly optimization algorithm (BOA) [3], bat algorithm (BA) [9], shuffled frog-leaping algorithm (SFLA) [41], artificial bee colony (ABC) algorithm [24], etc. These swarm intelligence optimization algorithms have been applied to various practical optimization problems. For example, [69] proposed a distributed fault estimation algorithm based on a hybrid improved biogeography based optimization (HIBBO), which introduced differential evolution into standard biogeography based optimization (BBO) to determine the moments when intermittent faults occur and disappear in discrete components. [65] proposed an adaptive competitive swarm optimization (ACSO) for intermittent fault estimation, which introduced a parameter adaptive adjustment strategy into the original competitive swarm optimization and improved the search ability of the algorithm. There are also some algorithms inspired by other ideas, such as teaching-learning-based optimization (TLBO) [51] inspired by teaching and feedback between teachers and students, sine cosine algorithm (SCA) [45] and golden sine algorithm (Gold-SA) [58] that based on mathematical ideas.

Harris Hawks Optimizer (HHO) is a swarm intelligence-based meta-heuristic algorithm proposed by [30] that simulates the behaviour of Harris hawks cooperatively foraging and surrounding prey with multiple strategies. HHO consists of two phases of exploration and exploitation, switching

between the two phases through the prey escape energy. HHO has strong competitive strengths compared with other swarm intelligence optimization algorithms [38]. On the one hand, HHO is easy to use because of its simple structure and no extra parameters except required parameters. On the other hand, the multiple strategies of HHO in the exploitation phase make the local search ability of the algorithm better. In addition, HHO simulates the continuous loss process of the prey escaping energy, and switches between exploration phase and exploitation phase based on the value of the prey escape energy, which fits the optimization process and makes the algorithm have good performance. HHO has been applied to various numerical optimization and practical problems. For example, Hussain et al. [32] applied HHO to feature selection, [21] adapted HHO to the traditional travel salesman problem, [53] applied HHO to image segmentation, [6] used HHO to cluster features, and [25] built a geohazard radar identification model using convolutional neural network and HHO. However, according to the No Free Lunch (NFL) theorem [63], no algorithm can achieve superior results on all optimization problems. HHO also suffers from low convergence accuracy, slow convergence, and unbalance in the exploration and exploitation phases. [50] pointed out that the global search ability based on the random strategy in HHO is insufficient in the exploration stage, and it is easy to fall into local optimum. In addition, the prey escape energy cannot fully reflect the actual search process, and then cannot balance the exploration and exploitation process well. [38] pointed out that when HHO optimizes complex problems, it is easy to converge with low accuracy and premature convergence. In recent years, there have been a lot of research addressing improvements to HHO. [35] combined HHO algorithm with SCA algorithm to enhance exploration capability of HHO. [8] introduced chaos strategy, topological multi-population strategy, and differential evolution strategy into the algorithm to enhance the performance of HHO. [13] used chaotic and changed the control parameters to enhance the ability of HHO. [2] discussed the influence of three selection strategies on the algorithm during the exploration phase. [18] added quasi-reflection-based learning mechanism into HHO to increase population diversity and speed of convergence. [22] used quantum particles to deal with dynamic optimization problems. In [33], opposition-based learning and chaotic local search strategy operator were combined to enhance the performance of HHO.

Self-organized criticality [5] refers to a system far from the equilibrium state. It can automatically adjust to a critical state without manual adjustment of any parameters. When a small disturbance occurs in the system, a chain reaction may occur and have a significant impact on the whole system. Extremal optimization (EO) is derived from the Bak-Sneppen (BS) [4] model, which shows self-organized criticality.

Different from most optimization algorithms acting on the group, it acts on the individual and always selects the worst component in the individual for variation, which may lead to a huge avalanche effect on the individual system and complete the improvement. EO has the characteristic of few parameters. Many articles have also applied EO to numerical optimization and practical problems. [12] proposed a swarm-based EO algorithm (PEO) to deal with single objective function optimization problems. The application of EO algorithm to multi-objective problems in [11] shows that EO also has good performance on multi-objective problems. [10] hybridized PSO and EO to make a novel algorithm. An adaptive population extremal optimization-based multivariable proportional-integral-derivative (MPID) control method was proposed in [71].

In this paper, we propose an improved HHO hybridized with EO which is named IHHO-EO. The main contributions of this work are summarized as follows:

- The Harris hawk individual's own historical optimal value is retained and applied to the exploration phase to enhance the exploration ability.
- The prey escape energy updating formula is changed to enable a better balance between the exploration phase and the exploitation phase.
- After the execution of the original HHO process, refracted opposition-based learning operation with dynamic  $k$  value is performed on all individuals. Refracted opposition-based learning is a variant of opposition-based learning that is important for improving the population quality as well as for jumping out of the local optimum. At the same time, the  $k$  value in the refracted opposition-based learning is dynamically generated to better generate the dynamic opposition-based solutions.
- Inspired by [10] that hybridized the PSO algorithm with EO, EO is added to our algorithm to enhance the local search capability.
- A large number of test functions are tested and compared with other algorithms to verify the excellent performance of the proposed algorithm IHHO-EO, and IHHO-EO is applied to the pressure vessel design problem.

In order to verify the performance of our proposed algorithm, extensive experiments were carried out. Firstly, IHHO-EO is compared with the original HHO algorithm, other new intelligent optimization algorithms and various HHO variants on 23 classical basic functions [68], and the superiority of IHHO-EO is verified by Friedman test [20], Quade test [15] and Wilcoxon sign rank test [14]. Subsequently, the effectiveness of the each proposed strategy is verified on high dimensions situation of the same 23 functions. Then, IHHO-EO is compared with many of the latest optimization algorithms on 29 CEC2017 functions [39].

Finally, IHHO-EO is applied to the pressure vessel design problem [54] which is a practical engineering problem with constraints. The experimental results show that the proposed IHHO-EO algorithm has better performance than the original HHO algorithm and other models in numerical optimization and practical engineering optimization problems.

The remaining content of the paper is organized as follows: Sect. 2 introduces the original HHO, refracted opposition-based learning, and the concept of EO. Section 3 presents the improved HHO hybridized with EO in detail. Section 4 performs experimental simulations and results analysis. Section 5 further applies the proposed algorithm to pressure vessel design problem. Section 6 concludes the whole paper as well as gives an outlook on future work.

## 2 Background

### 2.1 Harris Hawks optimizer (HHO)

HHO is a new swarm intelligence optimization algorithm proposed by [30]. The algorithm simulates the predation characteristics of Harris hawks. HHO is divided into two phases: exploration and exploitation, and uses the escape energy of prey as the basis for switching phases. The most important characteristic of Harris hawks is cooperative foraging and can show multiple attack patterns according to the change of environment and prey escape patterns. In Harris hawks optimization algorithm, Harris hawks individuals constitute candidate solutions, and the individual with the highest fitness in each iteration is considered as the prey. The phases of HHO are shown in Fig. 1.

#### 2.1.1 Exploration phase

When exploring the prey location, Harris hawks often randomly perch at a location and quietly search for the prey. During this phase, Harris hawks update their positions by two strategies: one is to explore based on the information about a randomly selected individual and the information about itself, while the other is to explore based on the current optimal individual and the average information of all individuals. The two strategies are determined by the parameter  $q$  and have equal chances of execution. The mathematical model is as follows:

$$X_i(t+1) = \begin{cases} X_{rand}(t) - r_1|X_{rand}(t) - 2r_2X_i(t)| & q \geq 0.5 \\ (X_{prey}(t) - X_m(t)) - r_3(LB + r_4(UB - LB)) & q < 0.5 \end{cases} \quad (1)$$

where  $X_i(t+1)$  and  $X_i(t)$  mean the position of the individual  $i$  in iteration  $t+1$  and iteration  $t$  respectively,  $q$ ,  $r_1$ ,  $r_2$ ,  $r_3$ , and  $r_4$  are random factors inside  $(0, 1)$ ,  $UB$  and  $LB$  are the upper and lower bounds of the search space,  $X_{rand}(t)$  is an

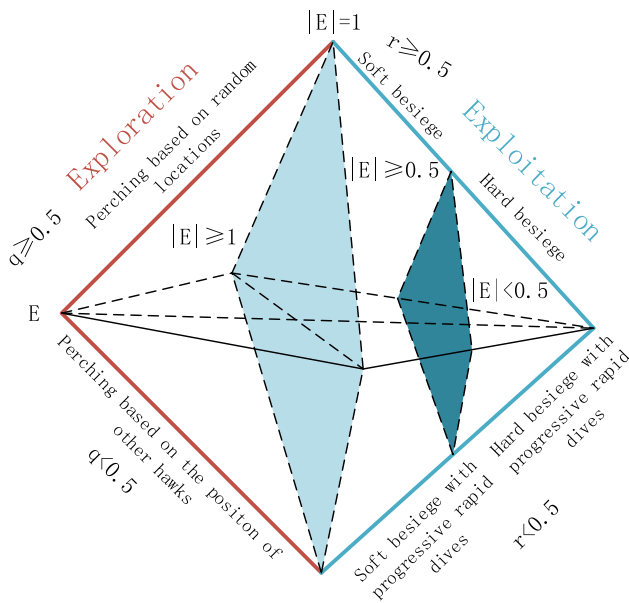


Fig. 1 The phases of HHO [30]

individual randomly selected from the population in iteration  $t$ ,  $X_{prey}(t)$  is the individual with the highest fitness in iteration  $t$ , and  $X_m(t)$  is the average position of all individuals in iteration  $t$  which is defined as below:

$$X_m(t) = \frac{1}{N} \sum_{i=1}^N X_i(t) \tag{2}$$

where  $N$  denotes the total numbers of individuals.

### 2.1.2 Transition from exploration to exploitation

In HHO, the prey escape energy  $E$  is defined to simulate the physical exertion of the prey during the escape process. When  $|E| \geq 1$ , the algorithm performs the exploration phase, reflecting the Harris hawks searching for the prey. When  $|E| < 1$ , the Harris hawks pursue the prey and enter the exploitation phase. The equation for the change of the prey escape energy  $E$  is as follows.

$$E_1 = 2 \times (1 - \frac{t}{T}) \tag{3}$$

$$E = E_2 \times E_1 \tag{4}$$

where  $E_2$  represents the initial value of the energy and ranges from  $-1$  to  $1$ , which is regenerated randomly at each iteration, and  $T$  is the maximum number of iterations of the algorithm.

### 2.1.3 Exploitation phase

During the exploitation phase, Harris hawks will besiege the prey they have discovered, forming an encirclement and waiting for an opportunity to surprise pounce. In the actual hunting process, the prey has a chance to escape, and the Harris hawks will also make different chasing actions according to the behavior of the prey. To more realistically mimic the actual siege behavior of the Harris hawks, HHO introduces four strategies. The random parameter  $r$  is defined to represent the chance that the prey can successfully escape ( $r < 0.5$ ) or not successfully escape ( $r \geq 0.5$ ) before the surprise pounce. The decision of which strategy to use is based on the parameter  $r$  and the prey escape energy  $E$  together. No matter how the prey reacts, Harris hawks will make a hard or soft encirclement to capture the prey. Along with the constant chase, the prey slowly loses energy and is eventually captured by the Harris hawks.

- Soft besiege

When  $|E| \geq 0.5$  and  $r \geq 0.5$ , the prey has enough energy to escape, and tries to escape the encirclement through some jumping behaviors, but ultimately fails. In this process, the hawks perform a high-altitude soft encirclement operation on the prey, constantly consuming the energy of the prey. After the prey is exhausted, the hawks swoop to capture the prey. The mathematical model is described as follows:

$$X_i(t + 1) = \Delta X(t) - E|JX_{prey}(t) - X_i(t)| \tag{5}$$

$$\Delta X(t) = X_{prey}(t) - X_i(t) \tag{6}$$

where  $\Delta X(t)$  is the difference between the location of the prey and the current individual in iteration  $t$ ,  $J = 2(1 - r_5)$  denotes the strength of the prey’s jump during escape, and  $r_5$  is a random number between  $(0, 1)$ .

- Hard besiege

When  $|E| < 0.5$  and  $r \geq 0.5$ , the prey is exhausted and does not have enough energy to escape, and there is no chance to escape. Harris hawks pounce on their prey in a hard-encircling fashion, the mathematical model is described as follows:

$$X_i(t + 1) = X_{prey}(t) - E|\Delta X(t)| \tag{7}$$

- Soft besiege with progressive rapid dives

When  $|E| \geq 0.5$  and  $r < 0.5$ , the prey has enough energy to escape successfully. At this point, Harris hawks will adopt a more intelligent soft-encirclement strategy, perform multiple

dives, and gradually adjust their directions and positions. Under this strategy, Harris hawks take two steps. The rule for the first step is as follows:

$$Y = X_{prey}(t) - E|JX_{prey}(t) - X_i(t)| \quad (8)$$

Then, to judge whether this dive has achieved better results. if not, Harris hawks will also suddenly make an irregular and rapid dive when approaching the prey. The mathematical model is described as follows:

$$Z = Y + S \times LF(D) \quad (9)$$

where  $D$  is the dimension of the optimization problem,  $S$  is a random vector of  $1 \times D$  dimension, and  $LF$  is the Levy flight [66]. The expression for  $LF$  is as follows:

$$LF(x) = 0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}}, \sigma = \left( \frac{\tau(1 + \beta) \times \sin(\frac{\pi\beta}{2})}{\tau(\frac{1+\beta}{2}) \times \beta \times 2^{(\frac{\beta-1}{2})}} \right)^{\frac{1}{\beta}} \quad (10)$$

where  $u, v$  is a random number between (0, 1) and  $\beta$  is a constant 1.5. Thus, the update formula for the soft besiege with progressive rapid dives of HHO is shown below:

$$X_i(t+1) = \begin{cases} Y, & \text{if } F(Y) < F(X_i(t)) \\ Z, & \text{if } F(Z) < F(X_i(t)) \end{cases} \quad (11)$$

- Hard besiege with progressive rapid dives

When  $|E| < 0.5$  and  $r < 0.5$ , the prey has a chance to escape, but does not have enough energy to do so. The hawks form a hard besiege before attacking, trying to reduce the average distance between their group and the prey, and then take a dive to kill the prey. The mathematical model is as follows:

$$\dot{Y} = X_{prey}(t) - E|JX_{prey}(t) - X_m(t)| \quad (12)$$

$$\dot{Z} = \dot{Y} + S \times LF(D) \quad (13)$$

$$X_i(t+1) = \begin{cases} \dot{Y}, & \text{if } F(\dot{Y}) < F(X_i(t)) \\ \dot{Z}, & \text{if } F(\dot{Z}) < F(X_i(t)) \end{cases} \quad (14)$$

This process is similar to soft besiege with progressive rapid dives, but it utilizes the average location information of the group. See Algorithm 1 for the pseudo-code of HHO.

---

#### Algorithm 1 Pseudo-code of HHO algorithm

---

```

1: Generate the random population  $X_i(i = 1, 2, \dots, N)$  and define the maximum number of iterations  $T$ 
2: Calculate the fitness value of each hawk
3: Choose the best individual as the prey( $X_{prey}$ )
4: Initial the iteration number  $t = 0$ 
5: while  $t < T$  do
6:   for each hawk  $X_i$  do
7:     Update the initial energy  $E_2$ 
8:     Update the prey energy  $E$  using Eq.(4)
9:     Update the jump strength  $J$ 
10:    if ( $|E| \geq 1$ ) then update the position of the current hawk by exploration strategy(Eq.(1))
11:    end if
12:    if ( $|E| < 1$ ) then
13:      if ( $r \geq 0.5$  and  $|E| \geq 0.5$ ) then update the position of the current hawk by soft besiege(Eq.(5))
14:      else if ( $r \geq 0.5$  and  $|E| < 0.5$ ) then update the position of the current hawk by hard besiege(Eq.(7))
15:      else if ( $r < 0.5$  and  $|E| \geq 0.5$ ) then update the position of the current hawk by soft besiege with progressive
rapid dives(Eq.(11))
16:      else if ( $r < 0.5$  and  $|E| < 0.5$ ) then update the position of the current hawk by hard besiege with progressive
rapid dives(Eq.(14))
17:    end if
18:  end if
19: end for
20: Calculate the fitness value of each hawk and update the best individual
21:  $t = t + 1$ 
22: end while

```

---

## 2.2 Extremal optimization (EO)

EO is inspired by Bak-Sneppen model of biological evolution [7]. EO has no concept of population and operates on a single individual. The specific operation is to mutate each component of the individual one by one, and then evaluate the fitness of the mutated individual. For the new individual with the best fitness, the corresponding component of the new individual is considered to be the worst component of the original individual. Then judging whether the new individual is better than the original individual. The basic EO has no extra parameters and only a mutation operation, which has a very good performance on local search [10]. The pseudo-code of EO is shown in Algorithm 2.

---

### Algorithm 2 Pseudo-code of EO

---

- 1: Generate a random solution  $X = (x_1, x_2, \dots, x_D)$ , set the optimal solution  $X_{best} = X$  and corresponding object function value  $F(X_{best}) = F(X)$
  - 2: **for** the solution  $X_i$  **do**
  - 3:   Mutate each variable  $x_i (i \in 1, 2, \dots, D)$  in turn, and keep the remaining variables unchanged when performing operation on the current variable. The process can get  $D$  new individuals  $X_1, X_2, \dots, X_D$ .
  - 4:   Evaluate each variable's local fitness  $\lambda_i = F(X_i) - F(X_{best})$
  - 5:   Rank all  $\lambda_i (i \in 1, 2, \dots, D)$  and find the variable with lowest fitness  $x_j (j \in 1, 2, \dots, D)$ . The corresponding new individual is  $X_j$
  - 6:   **if**  $F(X_j) < F(X)$  **then** let  $X = X_j$  and  $F(X) = F(X_j)$ . Otherwise, keep  $X$  unchanged
  - 7:   **end if**
  - 8:   Update the optimal solution  $X_{best}$  and its object function value  $F(X_{best})$
  - 9: **end for**
  - 10: If not meet terminal condition, Continue with the circulation operation
  - 11: Get the optimal solution  $X_{best}$  and its object function value  $F(X_{best})$
- 

## 2.3 Refracted opposition-based learning (ROBL)

### 2.3.1 Opposition-based learning (OBL)

Opposition-based learning is proposed by [59]. The basic idea is to expand the search by considering both the current solution and its inverse solution. Simultaneous search at that solution's location and its opposite location improves the possibility of obtaining a better solution.

Suppose  $x$  is a solution in a one-dimensional space, then its opposite solution  $\tilde{x}$  is defined as follows:

$$\tilde{x} = lb + ub - x \quad (15)$$

where  $x \in R$ ,  $lb$  and  $ub$  are the lower and upper bounds of  $x$ , respectively.

Extending to the  $D$ -dimensional problem, then the opposite solution of  $x$  in dimension  $i$  is defined as follows:

$$\tilde{x}_i = lb_i + ub_i - x_i \quad (16)$$

where  $x_i \in R$ ,  $x_i \in (lb_i, ub_i)$ ,  $\forall i \in 1, 2, \dots, D$ ,  $lb_i$  and  $ub_i$  are the lower and upper bounds in  $i$ th-dimension.

Opposition-based learning is able to explore the opposite side of the current solution, making it more likely to get a high-quality solution. In recent years, opposition-based learning has been combined with various algorithms, such as GWO [70], SCA [1]. However, algorithms still can not jump out of the local optimal in the late phase despite the use of the basic OBL strategy [17].

### 2.3.2 Refracted opposition-based learning (ROBL)

In order to improve the effect of opposition-based learning, a variant of OBL based on combining the laws of refraction

of light to better generate opposition-based solutions is presented by [17]. The mathematical definition of refracted opposition-based learning is shown in Fig. 2.

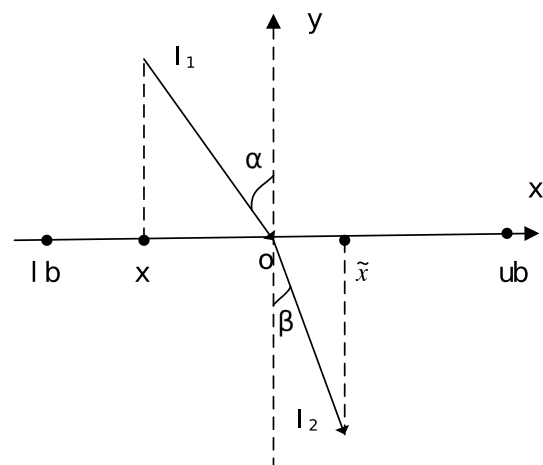


Fig. 2 Refracted opposition-based learning

In Fig. 2, the  $x$ -axis represents the dividing line and the  $y$ -axis denotes the normal line. The upper halves and lower halves of the  $x$ -axis are regarded as two different medias.  $l_1$  and  $l_2$  are the length of the incident light and the length of the refracted light, respectively.  $\alpha, \beta$  mean the angles of incidence and refraction.  $lb$  and  $ub$  denote the range of points on the  $x$ -axis. Point  $o$  is the midpoint of  $lb$  and  $ub$ . From Fig. 2, we can conclude the relational equation:

$$\sin\alpha = ((lb + ub)/2 - x)/l_1 \tag{17}$$

$$\sin\beta = (\tilde{x} - (lb + ub)/2)/l_2 \tag{18}$$

From the definition of the refractive index  $n$ , the following equation can be derived:

$$n = \frac{\sin\alpha}{\sin\beta} = \frac{l_2(lb + ub)/2 - x}{l_1(\tilde{x} - (lb + ub)/2)} \tag{19}$$

Let  $k = l_1/l_2$ , then Eq. (19) can be transformed into:

$$kn = \frac{\sin\alpha}{\sin\beta} = \frac{(lb + ub)/2 - x}{\tilde{x} - (lb + ub)/2} \tag{20}$$

After the transformation of Eq. (20), the expression of the refracted opposition-based solution  $\tilde{x}$  can be derived as follows:

$$\tilde{x} = \frac{lb + ub}{2} + \frac{lb + ub}{2kn} - \frac{x}{kn} \tag{21}$$

If  $k$  is equal to 1 and  $n$  is also equal to 1, then Eq. (21) turns into basic opposition-based learning. This shows that basic opposition-based learning is a special case of refracted opposition-based learning. ROBL can dynamically generate reverse learning solutions by adjusting the values of  $k$  and  $n$  which improve the diversity of reverse solutions and improve the possibility of the algorithm jumping out of the local optimal. It has been testified that ROBL can effectively improve the performance of the algorithm in the literature [40] and [61]. When  $n$  is set to 1 and the problem is multi-dimensional, the expression for refractive opposition-based solution is as follows:

$$\tilde{x}_{ij} = \frac{lb_j + ub_j}{2} + \frac{lb_j + ub_j}{2k} - \frac{x}{k} \tag{22}$$

where  $\tilde{x}_{ij}$  refers to the refracted opposition-based solution of individual  $i$  in the  $j$ th-dimension.  $lb_j$  and  $ub_j$  are the lower and upper bounds in  $j$ th-dimension.

### 3 The proposed IHHO-EO algorithm

#### 3.1 Introducing the individual’s own historical optimal location strategy

If  $|E| \geq 1$ , Harris hawks perform the exploration phase, searching for the prey. When the parameter  $q \geq 0.5$ , Harris hawk individuals fly to the next location based on a randomly selected companion and their own current locations. However, this way of randomly selecting a companion is too random and uses insufficient information. It is difficult to guide itself to fly to a better position and lacks exploration ability. Therefore, when  $q \geq 0.5$ , the individual’s historical optimal location is introduced and the global optimal location of the current best individual is combined to enhance exploration ability. At the same time, considering that the individual’s historical optimal position has little reference value in the early stage but will become more and more reliable with the increase of the number of iterations, the linear increasing coefficient is added to the individual’s historical optimal position. The improved exploration formula is shown below:

$$X_i(t + 1) = \begin{cases} r_1 X_i(t) + r_2 |X_{prey}(t) - \frac{t}{T} * X_{ipbest}(t)| & q \geq 0.5 \\ (X_{prey}(t) - X_m(t)) - r_3(LB + r_4(UB - LB)) & q < 0.5 \end{cases} \tag{23}$$

where  $X_{ipbest}(t)$  represents the historical optimal position of individual  $i$  in iteration  $t$ , and  $T$  is the maximum number of iterations.

#### 3.2 Nonlinear change of prey escape energy coefficient $E_1$

In the basic HHO algorithm, the prey escape energy  $E$  controls whether the algorithm performs the exploration phase or the exploitation phase. From Eq. (4),  $E$  is determined by the initial energy  $E_2$  and a coefficient  $E_1$  related to the number of iterations  $t$ . However,  $E_1$  is linearly reduced from 2 to 0, which can not balance the exploration and exploitation

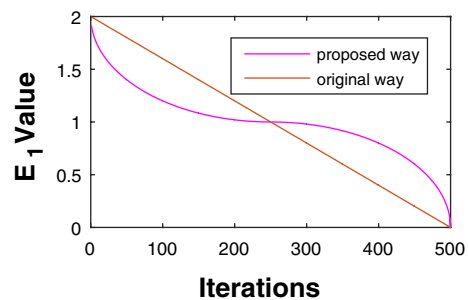


Fig. 3 The changing curve of  $E_1$

phases well. Therefore, Eq. (25) is proposed which updates  $E_1$  as follows:

$$A = \frac{t - \frac{T}{2}}{\frac{T}{2}} \quad (24)$$

$$E_1 = \begin{cases} 2 - \sqrt{1 - A^2} & t \leq \frac{T}{2} \\ \sqrt{1 - A^2} & t > \frac{T}{2} \end{cases} \quad (25)$$

Figure 3 shows a comparison between the original  $E_1$  update and the proposed  $E_1$  update.

From Fig. 3, it can be seen that  $E_1$  decreases faster in the early stage and is able to converge quickly to the optimal solution vicinity in the early stage. On the other hand, it decreases slower in the later stage and is able to better model the energy loss of the prey. Therefore it can switch adequately among the four exploitation strategies to perform a full local search.

### 3.3 ROBL with dynamic parameter $k$

Compared with the basic opposition-based learning, the refracted opposition-based learning introduced in Sect. 2.3 has two parameters  $k$  and  $n$ , which can produce opposition-based solutions more flexibly and improve the diversity of opposition-based solutions. It has a good performance on improving the quality of opposition-based solutions and helping the algorithm jump out of local optimal. When  $n$  is set to 1,  $k$  can be controlled to achieve the desired reverse effect. So if the value range of  $k$  is wide, diverse opposition-based solutions can be generated. The  $k$  value in basic ROBL is a fixed value, which can not generate dynamic opposition-based solutions well. Therefore, our proposed formula for the value of  $k$  is as follows:

$$k = \left(1 - \cos\left(2\pi * \frac{t}{T}\right)\right)^{10} \quad (26)$$

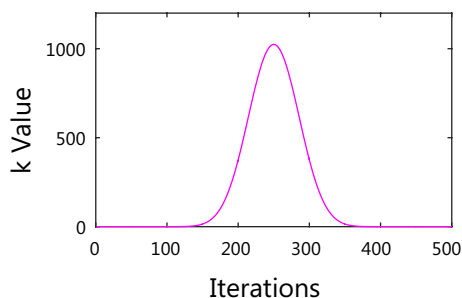


Fig. 4 The changing curve of  $k$  value

Figure 4 shows the change of  $k$  value with the number of iterations.

Combining Fig. 4 and Eq. (26), it can be concluded as follows: When the number of iterations  $t < 150$  or  $t > 350$ , the value of  $k$  is close to 0, while  $k$  obtains a larger value if  $150 < t < 350$ . Therefore, the value of  $k$  can have a relatively wide range and is highly dynamic, which can effectively improve the diversity of opposition-based solutions and increase the possibility of jumping out of the local optimum.

In view of the shortcoming that HHO is easy to fall into local optimum, the proposed ROBL with dynamic parameter  $k$  is carried out on all individuals after the basic process of HHO is performed. The newly generated opposition-based solution population is mixed with the original population in order of fitness, and the  $N$  individuals with the highest fitness in two populations are greedily selected as the new population.

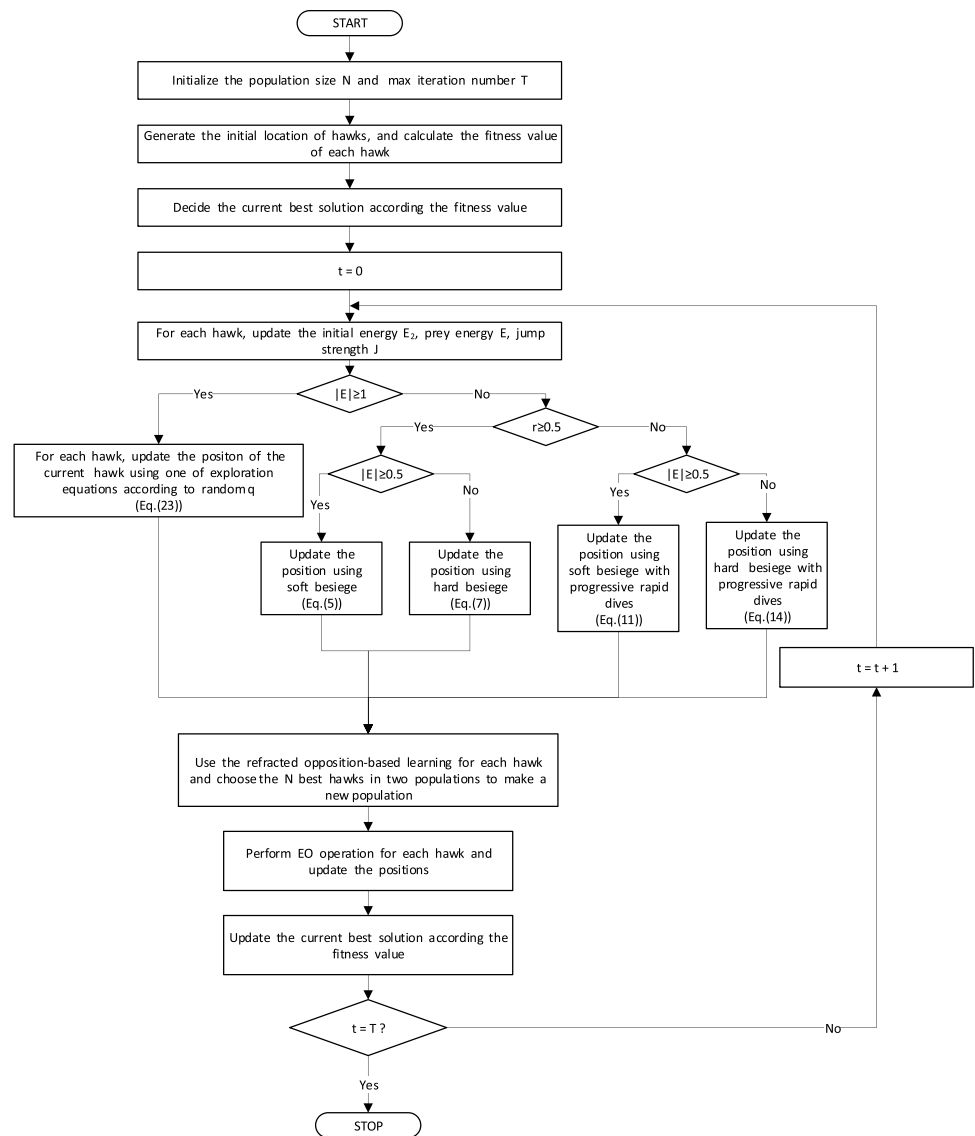
### 3.4 Combination with EO

The literature [10] has testified the strong local search ability of EO through a large number of experiments. The introduction of EO in Sect. 2.2 shows that EO acts on individuals, mutates each component in individuals, and generates new individuals with the same number of components. The new individual with the best fitness is judged with the original individual. If the new individual is superior to the original one, the original one will be replaced by the new one. Do the same for each individual.

In order to enhance the exploitation capability of HHO, EO operation is performed on all individuals after ROBL process. Since only mutation operation is performed on the elements in EO, mutation operation has a great influence on the effect of EO. The research work of [68] shows that Cauchy mutation can produce striding length for search with a higher probability than Gaussian mutation, so Cauchy mutation is more suitable for coarse tuning while Gaussian variation is more suitable for fine tuning. Hybrid Gaussian-Cauchy mutation (G-C mutation) [11] is used here. Cauchy mutation is used first, and if the newly generated variable is outside the range, a certain number of Cauchy mutation ( $TC$ ) is repeated to make the new variable fall within the range. If it still exceeds the range, a certain number of Gaussian mutation ( $TG$ ) is used to produce a smaller step size. Finally, if the range requirements are not met, the corresponding upper and lower boundaries are used. The value of  $TC$  determines the coarse-grained search time, and the value of  $TG$  determines the fine-grained search time. Therefore, the values of these two parameters should not be too large, or the search time will be greatly prolonged. [11] pointed out that the appropriate values of  $TC$  and  $TG$  were  $2 \sim 4$ . The values of  $TC$  and  $TG$  in this paper are set to 3.



**Fig. 5** Flow chart of IHHO-EO algorithm



### 3.5 The IHHO-EO algorithm

In view of the shortcomings of HHO, we propose the above four improvement strategies. The introduction of the historical optimal position of the Harris hawk individual can better guide the individual position update and enhance the exploration ability. The improved prey escape energy update formula makes the algorithm achieve a better balance between exploration and exploitation. The combination of ROBL and the proposed variation formula of parameter  $k$  increase the diversity of the solutions and the possibility of jumping out of local optimum. The hybridization with EO enhances the local exploitation capability of the algorithm. The pseudo-code of IHHO-EO is shown in Algorithm 3, and the flow chart is shown in Fig. 5.

### 3.6 Computational complexity analysis

Assume that the maximum number of individuals in the population is  $N$ , the problem dimension is  $D$ , and the number of iterations is  $T$ . In the original HHO algorithm, the computational complexity mainly consists of three parts: initialization, fitness evaluation and individual location update. The computational complexity of HHO is as follows:  $O(HHO) = O(N) + O(T \times N) + O(T \times N \times D)$

In the proposed IHHO-EO, the introduction of the individual’s own historical optimal position and the improvement of the prey escape energy update formula will not add additional complexity. The computational complexity of EO operation is  $O(T \times N \times D) + O(T \times N \times D)$ , which is the complexity of generating mutation and fitness evaluation. The computational complexity of ROBL is  $O(T \times N \times D) + O(T \times 2N)$ , where  $O(T \times N \times D)$  is the time

of refracted reverse operation for individuals, and  $O(T \times 2N)$  is the complexity of fitness evaluation.

test functions [39] are selected for evaluation. Among the 23 classic standard test functions,  $F_1 \sim F_7$  are unimodal

---

**Algorithm 3** Pseudo-code of IHHO-EO algorithm
 

---

```

1: Generate the random population  $X_i(i = 1, 2, \dots, N)$  and define the maximum number of iterations  $T$ 
2: Calculate the fitness value of each hawk
3: Choose the best individual as the prey( $X_{prey}$ )
4: Initial the number of iterations  $t = 0$ 
5: while  $t < T$  do
6:   for each hawk  $X_i$  do
7:     Update the initial energy  $E_2$ 
8:     Update the prey escape energy coefficient  $E_1$  using Eq.(25)
9:     Update the prey energy  $E$  using Eq.(4)
10:    Update the jump strength  $J$ 
11:    if ( $|E| \geq 1$ ) then update the position of the current hawk by exploration strategy(Eq.(23))
12:    end if
13:    if ( $|E| < 1$ ) then
14:      if ( $r \geq 0.5$  and  $|E| \geq 0.5$ ) then update the position of the current hawk by soft besiege(Eq.(5))
15:      else if ( $r \geq 0.5$  and  $|E| < 0.5$ ) then update the position of the current hawk by hard besiege(Eq.(7))
16:      else if ( $r < 0.5$  and  $|E| \geq 0.5$ ) then update the position of the current hawk by soft besiege with
    progressive rapid dives(Eq.(11))
17:      else if ( $r < 0.5$  and  $|E| < 0.5$ ) then update the position of the current hawk by hard besiege with
    progressive rapid dives(Eq.(14))
18:      end if
19:    end if
20:  end for
21:  for each hawk  $X_i$  do
22:    Calculate the refracted opposition-based solution of the current hawk(Eq.(26))
23:  end for
24:  Select the best  $N$  hawks in original population and refracted opposition-based population to make a new
  population
25:  Perform EO operation for each hawk
26:  Calculate the fitness value of each hawk and Update the best individual
27:   $t = t + 1$ 
28: end while

```

---

In summary, the complexity of IHHO-EO is  $O(N) + O(3 \times T \times N) + O(4 \times T \times N \times D)$ . Although the complexity of IHHO-EO has increased slightly more than the original HHO, the complexity of which is  $O(N) + O(T \times N) + O(T \times N \times D)$ , the performance of the proposed algorithm is excellent.

## 4 Experimental results and analysis

### 4.1 Experimental setup

To fully illustrate the performance of IHHO-EO, the experiments are divided into 4 parts, as shown in Table 1. In order to verify the performance of the proposed IHHO-EO algorithm in solving numerical optimization problems, 23 classical standard test functions [68] and 29 CEC2017

functions to test the exploitation performance of the algorithm,  $F_8 \sim F_{13}$  are multimodal functions to test the exploration performance and the ability of the algorithm to jump out of the local optimal solution.  $F_{14} \sim F_{23}$  are multimodal functions with fixed low dimensions to test the stability of the algorithm. The details of the 23 classic standard test functions are shown in Tables 2, 3, 4. Among the 29 CEC2017 test functions ( $f_1 \sim f_{30}$ ), there are unimodal, multimodal and composite functions, which are more complex and are used to test the ability of the algorithm in complex numerical optimization. The details of CEC2017 test functions are shown in Table 5. The bold data in all tables represent the best mean result achieved on this function among all the algorithms being compared.

In Exp. 1, IHHO-EO is compared in 23 classical standard test functions with some new intelligent optimization algorithms proposed in recent years and some famous classical

**Table 1** Summary of experimental conditions

Experiment	Test problem	Dimension	Algorithm	Population size	Maximum number of iterations	Number of runs
Exp.1	$F_1 \sim F_{13}$	30	IHHO-EO	30	500	30
		2~6	HHO [30]			
	$F_{14} \sim F_{23}$	WOA [47]				
		SSA [46]				
		GWO [48]				
		BOA [3]				
		MFO [43]				
		GA [31]				
		DE [57]				
		BBO [56]				
		PSO [37]				
Exp.2	$F_1 \sim F_{13}$	30	IHHO-EO	30	500	30
		2~6	HHO-SCA [35]			
	$F_{14} \sim F_{23}$	QRHHO [18]				
THHO [2]						
Exp.3	$F_1 \sim F_{13}$	50	IHHO-EO	30	500	30
		2~6	IHHO1			
	$F_{14} \sim F_{23}$	IHHO2				
Exp.4		$f_1 \sim f_{30}$	50	IHHO-EO	30	1500
	SCHHO [32]					
	GOA [55]					
	DA [44]					
	SCA [45]					
HHO [30]						

**Table 2** The description of unimodal functions  $F_1 \sim F_7$  functions

Function	Dimension	Range	$f_{min}$
$F_1(x) = \sum_{i=1}^n x_i^2$	30,50	[- 100,100]	0
$F_2(x) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $	30,50	[- 10,10]	0
$F_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	30,50	[- 100,100]	0
$F_4(x) = \max_i \{  x_i , 1 \leq i \leq n \}$	30,50	[- 100,100]	0
$F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30,50	[- 30,30]	0
$F_6(x) = \sum_{i=1}^n ([x_i + 0.5])^2$	30,50	[- 100,100]	0
$F_7(x) = \sum_{i=1}^n ix_i^4 + random [0, 1]$	30,50	[-1.28,1.28]	0

algorithms. The comparison algorithms include original HHO, Whale Optimization Algorithm (WOA) [47], Salp Swarm Algorithm (SSA) [46], Grey Wolf Optimizer (GWO) [48], Butterfly Optimization Algorithm [3], Moth-Flame Optimization Algorithm (MFO) [43], Genetic Algorithm (GA) [31], Differential Evolution (DE) [57], Biogeography-Based Optimization (BBO) [56], Particle Swarm Optimizer

(PSO) [37]. The number of population for each algorithm was set to 30 and each algorithm was run 30 times with 500 iterations.

Exp.2 compares the IHHO-EO with three improved variants of HHO, including HHO-SCA [35], QRHHO [18], and THHO [2].

Exp.3 conducts on the classical standard test functions in 50 dimensions to explore the effectiveness of the added strategies.

Exp.4 conducts on CEC2017 test functions and set each test function to 50 dimensions. Some intelligent optimization algorithms (Grasshopper Optimisation Algorithm (GOA) [55], Dragonfly Algorithm (DA) [44], Sine Cosine Algorithm [45], HHO) and a variant of HHO (SCHHO [32]) were compared. The number of individuals of each algorithm was set to 30, and 30 runs were performed with 1500 iterations each time.

The parameters settings of the algorithms used in the experiment are shown in Table 6. The parameter setting has a certain impact on the performance of the algorithm.

**Table 3** The description of multimodal functions  $F_8 \sim F_{13}$  functions

Function	Dimension	Range	$f_{min}$
$F_8(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	30,50	[- 500,500]	-418.9829x n
$F_9(x) = \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$	30,50	[-5.12,5.12]	0
$F_{10}(x) = -20\exp(-0.2\sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	30,50	[- 32,32]	0
$F_{11}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	30,50	[- 600,600]	0
$F_{12}(x) = \frac{\pi}{n} \{ 10\sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2 \}$ + $\sum_{i=1}^n u(x_i, 10, 100, 4)$	30,50	[- 50,50]	0
$y_i = 1 + \frac{x_i+1}{4}, u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$			
$F_{13}(x) = 0.1 \{ \sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \}$ + $\sum_{i=1}^n u(x_i, 5, 100, 4)$	30,50	[- 50,50]	0

**Table 4** The description of fixed-dimension multimodal functions  $F_{14} \sim F_{23}$  functions

Function	Dimension	Range	$f_{min}$
$F_{14}(x) = (\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6})^{-1}$	2	[-65.536,65.536]	0.998
$F_{15}(x) = \sum_{i=1}^{11} [a_i - \frac{x_i(b_i^2 + b_j x_2)}{b_i^2 + b_j x_3 + x_4}]^2$	4	[- 5,5]	0.0030
$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[- 5,5]	-1.036
$F_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos x_1 + 10$	2	$x_1 \in [- 5,10], x_2 \in [0,15]$	0.39789
$F_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]$ $\times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	[- 2,2]	3
$F_{19}(x) = - \sum_{i=1}^4 c_i \exp(- \sum_{j=1}^3 a_{ij}(x_j - p_{ij})^2)$	3	[0,1]	-3.8628
$F_{20}(x) = - \sum_{i=1}^4 c_i \exp(- \sum_{j=1}^6 a_{ij}(x_j - p_{ij})^2)$	6	[0,1]	-3.322
$F_{21}(x) = - \sum_{i=1}^5 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	-10.1532
$F_{22}(x) = - \sum_{i=1}^7 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	-10.4029
$F_{23}(x) = - \sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	-10.5364

Therefore, the values of the parameters of the compared algorithms were taken directly from their original papers for fair comparison. All experiments were performed on a 2.60GHz computer with i5-3320 M CPU, and 8 GB RAM.

### 4.2 Comparison with advanced meta-heuristic algorithms and famous classical algorithms

This part of the experiment compares IHHO-EO with HHO, WOA, SSA, GWO, BOA, MFO, GA, DE, BBO, PSO which are some of the swarm intelligence optimization algorithms proposed in recent years and some famous classical algorithms. Tables 7, 8, 9, 10, 11 show the experimental results.

#### 4.2.1 Comparison with advanced meta-heuristic algorithms

In terms of mean, standard deviation, best and worst results, IHHO-EO has a significant advantage over all these algorithms or the results are essentially equivalent, which is obvious on functions  $F_1 \sim F_4$ . On the fixed dimensional functions  $F_{14} \sim F_{23}$ , only the IHHO-EO algorithm was able to find the optimal value. Compared with the original HHO, IHHO-EO has similar performance on  $F_5, F_{12}$  and  $F_{13}$ , but slightly lower in terms of mean.

To further illustrate the difference between IHHO-EO and other algorithms, Wilcoxon signed rank test [14] was

**Table 5** The description of CEC2017 benchmark functions

Function	Description	Type	Optimum
$f_1$	Shifted and Rotated Bent Cigar Function	Unimodal	100
$f_3$	Shifted and Rotated Zakharov function	Unimodal	300
$f_4$	Shifted and Rotated Rosenbrock's function	Multimodal	400
$f_5$	Shifted and Rotated Rastrigin's function	Multimodal	500
$f_6$	Shifted and Rotated Expanded Scaffer's F6 function	Multimodal	600
$f_7$	Shifted and Rotated Lunacek Bi-Rastrigin function	Multimodal	700
$f_8$	Shifted and Rotated Non-Continuous Rastrigin's function	Multimodal	800
$f_9$	Shifted and Rotated Lévy function	Multimodal	900
$f_{10}$	Shifted and Rotated Schwefel's function	Multimodal	1000
$f_{11}$	Hybrid Function 1 (N = 3)	Hybrid	1100
$f_{12}$	Hybrid Function 2 (N = 3)	Hybrid	1200
$f_{13}$	Hybrid Function 3 (N = 3)	Hybrid	1300
$f_{14}$	Hybrid Function 4 (N = 4)	Hybrid	1400
$f_{15}$	Hybrid Function 5 (N = 4)	Hybrid	1500
$f_{16}$	Hybrid Function 6 (N = 4)	Hybrid	1600
$f_{17}$	Hybrid Function 6 (N = 5)	Hybrid	1700
$f_{18}$	Hybrid Function 6 (N = 5)	Hybrid	1800
$f_{19}$	Hybrid Function 6 (N = 5)	Hybrid	1900
$f_{20}$	Hybrid Function 6 (N = 6)	Hybrid	2000
$f_{21}$	Composition Function 1 (N = 3)	Composition	2100
$f_{22}$	Composition Function 2 (N = 3)	Composition	2200
$f_{23}$	Composition Function 3 (N = 4)	Composition	2300
$f_{24}$	Composition Function 4 (N = 4)	Composition	2400
$f_{25}$	Composition Function 5 (N = 5)	Composition	2500
$f_{26}$	Composition Function 6 (N = 5)	Composition	2600
$f_{27}$	Composition Function 7 (N = 6)	Composition	2700
$f_{28}$	Composition Function 8 (N = 6)	Composition	2800
$f_{29}$	Composition Function 9 (N = 3)	Composition	2900
$f_{30}$	Composition Function 10 (N = 3)	Composition	3000

performed. If the  $p$ -value is less than 0.05, IHHO-EO can be considered significantly different from the compared algorithm on this function, and *NaN* indicates that there is essentially no difference between the two algorithms. It can be seen from Table 10 that IHHO-EO and the compared algorithms have obvious effect differences on most functions. The symbol “+/ $\approx$ /-” indicates that IHHO-EO achieves better, similar, and worse results than other algorithms on the mean for 23 functions, and it is obvious that IHHO-EO has better results than the other algorithms for most functions.

#### 4.2.2 Convergence analysis

The convergence diagram of experiments was analyzed to further illustrate the performance of IHHO-EO. Figure 6 shows the convergence of each algorithm on some functions. On most functions, IHHO-EO algorithm has faster convergence speed and higher convergence accuracy. For function  $F_1$ , IHHO-EO achieved the optimal value 0 in the

first half of iterations, while the other algorithms couldn't find the optimal value at the end of iterations. In functions  $F_6$ ,  $F_7$  and  $F_8$ , IHHO-EO not only can achieve better results, but also has faster convergence speed, which reflects the effectiveness of the proposed strategies. In function  $F_{10}$ , IHHO-EO algorithm has a very good convergence speed compared with other algorithms and finds the optimal value of the function first. In function  $F_{13}$ , although the final result of IHHO-EO is slightly lower than that of the original HHO algorithm, the initial convergence speed is still higher than HHO. IHHO-EO still maintains good convergence speed and accuracy in function  $F_{14}$  with fixed dimension.

#### 4.2.3 Comparison with famous classical algorithms

As can be seen from Table 11, compared with GA, DE, PSO and BBO, IHHO-EO achieves the smallest mean and standard deviation on the 23 classical standard test functions, indicating the good performance of IHHO-EO. GA

**Table 6** Parameter settings

Algorithm	Parameter	Value
WOA	Convergence constant $a$	[0,2]
SSA	Coefficient $c_1$	$[2/e, 2]$
GWO	Convergence constant $a$	[0,2]
BOA	Sensory modality $c$	0.01
	Power exponent dependent $a$	0.1
MFO	Convergence constant $a$	$[-2, -1]$
GOA	Convergence constant $c_{max}$	1
	Convergence constant $c_{min}$	0.0004
DA	Inertia weight $w$	0.7
	Separation weight $s$	0.1
	Alignment weight $a$	0.1
	Cohesion weight $c$	0.7
	Food factor $f$	1
	Enemy factor $e$	1
SCA	Convergence constant $\alpha$	2
SCHHO	Convergence constant $\alpha$	2
DE	Scaling factor	0.5
	Crossover probability	0.5
PSO	Inertia factor	0.3
	$c_1$	1
	$c_2$	1
BBO	Habitat modification probability	1
	Immigration probability limits	[0,1]
	Step size	1
	Max immigration (I) and Max emigration (E)	1
	Mutation probability	0.005

and BBO also find the theoretical optimal value on function  $F_{14}$  and  $F_{18}$ .

### 4.3 Comparison with three variants of HHO

Since the HHO algorithm was proposed, many improved HHO variants have been proposed and achieved good results. In this experiment, three HHO modified variants (HHO-SCA, QRHHO, THHO) proposed in recent two years are selected for comparison. Table 12 shows the comparison of the results in 23 classical standard test functions. It can be seen that IHHO-EO obtains the best or comparable mean values for most of the functions. The standard deviation of IHHO-EO is also small, indicating good stability. Friedman test [20] and Quade test [15] are conducted on the results, as shown in Table 13. The results show that IHHO-EO ranked higher than other algorithms, indicating that IHHO-EO has better performance than other algorithms.

### 4.4 Impact of the added strategies on the optimizer

This experimental section explores the effectiveness of the various added strategies of the proposed IHHO-EO. Tests are performed on the 50 dimensions of  $F_1 \sim F_{13}$  and on the  $F_{14} \sim F_{23}$  with fixed dimensions. IHHO1 is used to represent the addition of the own historical optimal position strategy and the nonlinear prey energy escaping factor strategy, IHHO2 to represent the addition of only the refracted opposition-based learning strategy, HHO-EO to represent the addition of only the EO strategy, and IHHO-EO to represent the addition of all strategies. As shown in Tables 14, 15, the original HHO, IHHO1, IHHO2, HHO-EO, and IHHO-EO are all able to find optimal values for functions  $F_9, F_{10}, F_{16}$ , and  $F_{18}$ , but the original HHO does not achieve as well as they do for the remaining functions, indicating the effectiveness of the added strategies. IHHO1 obtains better results than HHO algorithm on functions  $F_1 \sim F_4, F_7, F_8, F_{19}$  and  $F_{21}$ . The improved effect of IHHO2 on functions  $F_1 \sim F_4$  and  $F_{21} \sim F_{23}$  is obvious while HHO-EO achieved best results on functions  $F_8, F_{12}, F_{13}$  and  $F_{15}$ . IHHO-EO combines these strategies and achieves good results on most functions.

### 4.5 Comparison with advanced meta-heuristic algorithms and one variant of HHO on CEC2017 functions

In order to test the performance of IHHO-EO on complex function optimization problems, IHHO-EO was applied to solving more complex CEC2017 test functions. The results of IHHO-EO in comparison with GOA, DA, SCA, HHO and a modified HHO variant (SCHHO), are shown in Table 16. The mean in Table 16 is the difference between the actual run result and the optimal value of the function. Compared with GOA, DA, SCA and HHO, IHHO-EO has better performance with respect to the mean. GOA has a smaller mean value than IHHO-EO on function  $f_7$ , while it does not perform as well as IHHO-EO on the remaining functions. IHHO-EO achieves better average results than DA, SCA, and HHO on all functions. Our algorithm outperforms SCHHO on 22 functions, and is slightly inferior to SCHHO on the remaining 7 functions.

## 5 IHHO-EO for pressure vessel design problem

In this section, IHHO-EO algorithm is further applied to the pressure vessel design [36], which is a typical engineering design problem. The problem contains four variables: inner radius ( $R$ ), the thickness of the shell ( $T_s$ ), the thickness of the head ( $T_h$ ) and the length of the cylindrical section without the head ( $L$ ). The objective of the problem is to minimize the

**Table 7** Results of unimodal benchmark functions ( $F_1 \sim F_7$ )

Function	Optimum	Metric	IHHO-EO	HHO	WOA	SSA	GWO	BOA	MFO
$F_1$	0	Mean	0	9.9487E-96	8.3153E-74	1.8455E-07	1.0026E-27	1.2998E-11	1338.5681
		Best	0	9.02E-111	8.0691E-86	2.6276E-08	5.301E-29	1.1576E-11	0.30677
		Worst	0	2.1148E-94	2.2119E-72	6.5591E-07	8.5605E-27	1.4286E-11	10005.9983
		Std	0	4.0364E-95	4.0357E-73	1.9952E-07	1.6758E-27	7.9566E-13	3456.4151
$F_2$	0	Mean	0	3.4574E-50	2.4116E-50	2.5407	9.1272E-17	4.6628E-09	40.1198
		Best	0	3.7506E-58	1.1989E-57	0.18905	9.6841E-18	1.298E-09	0.20263
		Worst	0	1.0077E-48	3.3999E-49	6.673	3.504E-16	6.0453E-09	110.0005
		Std	0	1.8382E-49	7.4391E-50	1.9534	7.6105E-17	1.242E-09	22.1887
$F_3$	0	Mean	0	3.5633E-72	33729.8751	1607.6295	1.7906E-05	1.2426E-11	21826.0701
		Best	0	3.4011E-100	6519.3242	378.9169	1.5808E-08	1.1196E-11	2387.2473
		Worst	0	1.0687E-70	66126.1739	4984.7933	0.00025642	1.4259E-11	57081.4939
		Std	0	1.9511E-71	14608.5768	989.9224	5.0355E-05	7.7769E-13	13295.1782
$F_4$	0	Mean	0	1.1454E-48	54.661	10.5345	6.3561E-07	6.1329E-09	70.8847
		Best	0	1.9299E-56	0.067443	4.989	1.3012E-07	5.2444E-09	53.3842
		Worst	0	2.4476E-47	85.2955	19.6842	2.4273E-06	7.1507E-09	81.5999
		Std	0	4.704E-48	26.8689	3.7344	4.7553E-07	4.247E-10	8.3753
$F_5$	0	Mean	0.028631	<b>0.010662</b>	28.2326	263.5157	27.0458	28.9469	2686823.4845
		Best	8.3697E-05	5.7042E-07	27.3309	21.7871	25.7248	28.903	217.105
		Worst	0.14611	0.072171	28.782	1561.0287	28.5501	28.9907	80123135.6865
		Std	0.032677	0.015577	0.42258	375.2412	0.68358	0.021129	14625423.3654
$F_6$	0	Mean	9.427E-05	0.00020269	0.69196	<b>2.9047E-07</b>	0.79164	5.837	1331.5302
		Best	1.9533E-10	1.1085E-06	0.32498	2.5249E-08	0.24993	4.6841	0.59235
		Worst	0.0013527	0.0011871	1.2622	4.8125E-06	1.4772	6.9097	20000.8853
		Std	0.00028983	0.00027241	0.23031	8.783E-07	0.36379	0.58422	4326.8312
$F_7$	0	Mean	<b>9.6075E-05</b>	0.00019867	0.0017061	0.18043	0.002024	0.0016012	2.9596
		Best	4.4792E-06	2.0688E-05	4.9244E-05	0.068782	0.00064546	0.00052631	0.093355
		Worst	0.00035016	0.00086011	0.0091121	0.35208	0.0050645	0.00379	21.6952
		Std	9.6785E-05	0.00018566	0.0021431	0.071463	0.0010301	0.00061379	4.7218

**Table 8** Results of multimodal benchmark functions( $F_8 \sim F_{13}$ )

Function	Optimum	Metric	IHHO-EO	HHO	WOA	SSA	GWO	BOA	MFO
$F_8$	-418.9829xD	Mean	-12569.3743	-12568.6992	-10919.7653	-7572.4535	-6333.3733	-3754.8758	-8365.7394
		Best	-12569.4855	-12569.4866	-12569.486	-9663.4247	-8708.7933	-4350.1699	-9828.8947
		Worst	-12569.2085	-12564.6801	-6941.2238	-6166.0082	-3770.4069	-3106.7849	-6607.205
		Std	0.066997	1.217	1816.8252	773.7814	1036.79	338.6855	1003.5812
$F_9$	0	Mean	0	0	0	55.8834	3.2719	18.1619	162.6352
		Best	0	0	0	21.8891	0	0	68.6983
		Worst	0	0	0	107.4551	12.2179	205.222	233.1748
		Std	0	0	0	18.9142	4.0665	55.6764	48.3901
$F_{10}$	0	Mean	8.8818E-16	8.8818E-16	4.3225E-15	2.797	9.5154E-14	5.8769E-09	14.9811
		Best	8.8818E-16	8.8818E-16	8.8818E-16	1.1551	7.1942E-14	4.3031E-09	0.863
		Worst	8.8818E-16	8.8818E-16	7.9936E-15	4.9895	1.3589E-13	6.5854E-09	19.9632
		Std	0	0	2.1847E-15	0.86078	1.6709E-14	5.1675E-10	7.0343
$F_{11}$	0	Mean	0	0	0.0065209	0.020095	0.0032987	5.2603E-12	21.923
		Best	0	0	0	0.00033445	0	1.3893E-12	0.29495
		Worst	0	0	0.19563	0.040865	0.021443	1.0529E-11	180.7825
		Std	0	0	0.035716	0.011754	0.0063222	2.5443E-12	45.3461
$F_{12}$	0	Mean	1.0136E-05	8.7754E-06	0.05003	6.3901	0.043423	0.63375	8534122.8226
		Best	1.0491E-07	7.7358E-08	0.011401	1.3153	0.013505	0.35792	1.7546
		Worst	8.3082E-05	5.9613E-05	0.085254	18.8941	0.086178	0.83815	256000019.6478
		Std	2.0648E-05	1.2599E-05	0.020311	3.3038	0.018228	0.1424	46738846.3604
$F_{13}$	0	Mean	7.9115E-05	4.6855E-05	0.88642	18.3322	0.67763	2.8654	13668813.1374
		Best	4.9018E-06	7.0649E-07	0.28705	0.0070184	0.29054	2.385	6.38
		Worst	0.00035495	0.00033658	1.4581	54.0221	1.0898	2.9996	410063510.8766
		Std	7.4781E-05	8.4305E-05	0.3275	14.8161	0.20726	0.19373	74867006.0784



**Table 9** Results of fixed-dimension multimodal benchmark functions( $F_{14} \sim F_{23}$ )

Function	Metric	Optimum	IHHO-EO	HHO	WOA	SSA	GWO	BOA	MFO
$F_{14}$	Mean	0.998	<b>0.998</b>	1.3612	3.1219	1.3291	4.3632	1.3563	2.8728
	Best	0.998	0.998	0.998	0.998	0.998	0.998	0.998	0.998
	Worst	0.998	0.998	5.9288	10.7632	2.9821	12.6705	2.9834	9.8039
	Std		2.4801E-15	0.95204	3.553	0.60211	4.3162	0.56465	2.3126
$F_{15}$	Mean	0.00030	<b>0.00033615</b>	0.00034558	0.0010482	0.0041423	0.0037795	0.00040607	0.0024693
	Best		0.00030762	0.00030868	0.00031334	0.00043692	0.00030751	0.00031873	0.0005945
	Worst		0.0005132	0.00043594	0.012772	0.020367	0.020363	0.00062515	0.020363
	Std		4.5518E-05	3.4114E-05	0.002244	0.0073839	0.0075482	7.7744E-05	0.0050571
$F_{16}$	Mean	-1.036	<b>-1.0316</b>	<b>-1.0316</b>	<b>-1.0316</b>	<b>-1.0316</b>	<b>-1.0316</b>	-1.0315	<b>-1.0316</b>
	Best		-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316
	Worst		-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0311	-1.0316
	Std		1.0508E-10	9.1587E-09	1.3772E-09	4.1042E-14	3.0285E-08	0.00014732	6.7752E-16
$F_{17}$	Mean	0.39789	<b>0.39789</b>	0.3979	0.39791	<b>0.39789</b>	0.39797	0.39991	<b>0.39789</b>
	Best		0.39789	0.39789	0.39789	0.39789	0.39789	0.39798	0.39789
	Worst		0.39789	0.39802	0.39809	0.39789	0.39902	0.40594	0.39789
	Std		4.0947E-10	2.1698E-05	4.4184E-05	4.6918E-14	0.0002686	0.0022226	0
$F_{18}$	Mean	3	<b>3</b>	<b>3</b>	4.8068	<b>3</b>	<b>3</b>	3.0537	<b>3</b>
	Best		3	3	3	3	3	3.0023	3
	Worst		3	3	30.1161	3	3.0001	3.3327	3
	Std		7.6861E-13	6.6727E-07	6.8656	1.3219E-12	4.1147E-05	0.078446	2.202E-15
$F_{19}$	Mean	-3.8628	<b>-3.8628</b>	-3.8592	-3.8546	<b>-3.8628</b>	-3.8611	-3.8315	<b>-3.8628</b>
	Best		-3.8628	-3.8628	-3.8628	-3.8628	-3.8628	-3.8622	-3.8628
	Worst		-3.8628	-3.8347	-3.8232	-3.8628	-3.8628	-3.8549	-3.8628
	Std		3.1006E-11	0.0053535	0.0084548	5.3291E-12	0.0027714	0.045626	2.7101E-15
$F_{20}$	Mean	-3.32	-3.2863	-3.1033	-3.2025	-3.2051	-3.2933	-2.912	-3.2303
	Best		-3.32	-3.315	-3.32	-3.322	-3.322	-3.1414	-3.322
	Worst		-3.2031	-2.8691	-2.422	-3.1719	-3.1369	-1.9689	-3.1376
	Std		0.055415	0.11826	0.18895	0.040591	0.059532	0.26129	0.058556
$F_{21}$	Mean	-10.1532	<b>-10.1532</b>	-5.2122	-7.909	-7.7317	-9.1438	-5.1888	-7.1349
	Best		-10.1532	-9.988	-10.1507	-10.1532	-10.1529	-8.5392	-10.1532
	Worst		-10.1532	-4.9919	-2.6152	-2.6305	-2.6304	-4.5846	-2.6305
	Std		9.1365E-08	0.90209	2.7732	3.3143	2.3442	0.83542	3.1888
$F_{22}$	Mean	-10.4029	<b>-10.4029</b>	-5.2432	-6.2929	-8.1031	-10.4008	-5.8777	-7.4877

Table 9 (continued)

Function	Optimum	Metric	IHHO-EO	HHO	WOA	SSA	GWO	BOA	MFO
$F_{23}$	Best		-10.4029	-9.886	-10.3907	-10.4029	-10.4024	-9.8753	-10.4029
	Worst		-10.4029	-5.068	-1.8369	-2.7519	-10.3982	-3.6274	-2.7519
	Std		9.2175E-08	0.8769	2.5675	3.1493	0.0010365	1.6242	3.4431
	Mean	-10.5364	-5.5125	-7.0953	-9.3264	-10.3557	-6.213	-7.647	
	Best		-10.5364	-10.5202	-10.5186	-10.5364	-9.4224	-10.5364	
	Worst		-10.5364	-2.4061	-2.4081	-2.4273	-5.1746	-4.208	-1.6766
		Std	2.4566E-08	1.7439	3.0565	2.5065	0.97857	1.5299	3.6595

cost, including material cost, forming cost and welding cost. A diagram of the problem is shown in Fig. 7. The pressure vessel design problem can be abstracted as a single objective minimization problem containing four variables and four constraints. The mathematical model of the problem is as follows.

$$\begin{aligned}
 \text{Consider } \vec{x} &= [x_1 \ x_2 \ x_3 \ x_4] = [T_s \ T_h \ R \ H] \\
 \text{Minimize } f(\vec{x}) &= 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 \\
 &\quad + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \\
 \text{Subject to } g_1(\vec{x}) &= -x_1 + 0.0193x_3 \leq 0 \\
 g_2(\vec{x}) &= -x_2 + 0.00954x_3 \leq 0 \\
 g_3(\vec{x}) &= -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0 \\
 g_4(\vec{x}) &= x_4 - 240 \leq 0 \\
 \text{Variable range } &0 \leq x_1, x_2 \leq 99 \\
 &10 \leq x_3, x_4 \leq 200
 \end{aligned} \tag{27}$$

IHHO-EO is compared with IEHHO [50], HHO, GWO [48], GA [47], HPSO [26], BA [19] and WOA [47] algorithms on this problem, and the optimal results obtained by them are shown in Table 17. It can be seen from the table that the proposed IHHO-EO algorithm is superior to all compared algorithms in solving this practical engineering optimization problem with constraints and has excellent performance.

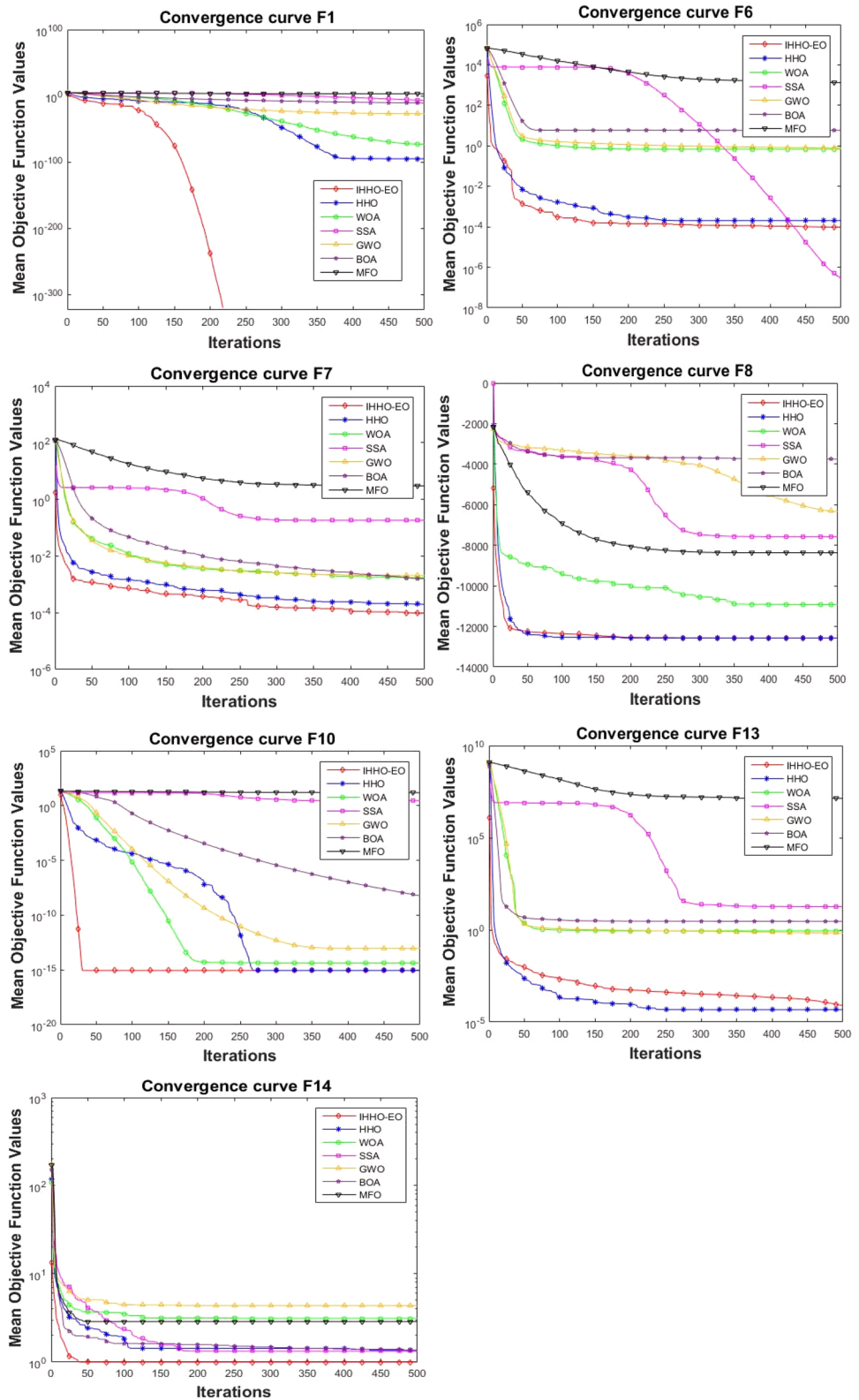
## 6 Conclusion and future work

In this paper, four improved strategies were introduced to the original HHO and a novel algorithm, named IHHO-EO, was proposed. First of all, the historical optimal location of the individual is added to solve the defect of over-randomization and using insufficient information in the exploration stage of HHO when updating the location. The better location that has been searched is used to enhance the guidance and reduce the possibility of blind search. Second, the linear update of prey escape energy is improved to allow for a better balance between the exploration and exploitation phases. Refracted opposition-based learning is introduced and a dynamic approach to the value of  $k$  is proposed to produce dynamic opposition-based solutions on a larger scale, improving the quality of the population and increasing the likelihood of jumping out of the local optimum. Finally, EO operation is combined to enhance the local exploitation capability of the algorithm. To illustrate the performance of IHHO-EO, extensive experiments are conducted. Exp.1 tested IHHO-EO on 23 classical benchmark functions in 30 dimensions and compared it with various optimization algorithms such as

**Table 10** Results of Wilcoxon signed rank test

Function	HHO	WOA	SSA	GWO	BOA	MFO
$F_1$	1.2118E-12	1.2118E-12	1.2118E-12	1.2118E-12	1.2118E-12	1.2118E-12
$F_2$	1.2118E-12	1.2118E-12	1.2118E-12	1.2118E-12	1.2118E-12	1.2118E-12
$F_3$	1.2118E-12	1.2118E-12	1.2118E-12	1.2118E-12	1.2118E-12	1.2118E-12
$F_4$	1.2118E-12	1.2118E-12	1.2118E-12	1.2118E-12	1.2118E-12	1.2118E-12
$F_5$	0.00031821	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11
$F_6$	1.2493E-05	3.0199E-11	0.00065486	3.0199E-11	3.0199E-11	3.0199E-11
$F_7$	0.0058282	3.1967E-09	3.0199E-11	3.4971E-09	3.0199E-11	3.0199E-11
$F_8$	0.093341	3.8249E-09	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11
$F_9$	NaN	NaN	1.2118E-12	4.5092E-12	1.6995E-08	1.2118E-12
$F_{10}$	NaN	8.8114E-10	1.2118E-12	1.0768E-12	1.2118E-12	1.2118E-12
$F_{11}$	NaN	0.33371	1.2118E-12	0.0055843	1.2118E-12	1.2118E-12
$F_{12}$	0.14532	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11
$F_{13}$	0.00095207	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11	3.0199E-11
$F_{14}$	2.8484E-11	2.8484E-11	0.014923	2.8484E-11	2.8484E-11	0.18338
$F_{15}$	0.061452	3.9648E-08	4.0772E-11	0.00238	3.2555E-07	3.018E-11
$F_{16}$	2.683E-07	2.7391E-08	0.0003714	2.8809E-11	2.8809E-11	1.1396E-12
$F_{17}$	5.5864E-06	1.9388E-10	0.13291	2.9897E-11	2.9897E-11	1.9264E-10
$F_{18}$	1.4147E-07	3.0199E-11	7.0405E-07	3.0199E-11	3.0199E-11	5.6239E-11
$F_{19}$	3.0161E-11	3.0161E-11	0.54439	3.0161E-11	3.0161E-11	1.2098E-12
$F_{20}$	2.2273E-09	2.1327E-05	1.0666E-07	0.00055611	3.0199E-11	0.5625
$F_{21}$	3.0199E-11	3.0199E-11	0.12235	3.0199E-11	3.0199E-11	1
$F_{22}$	3.0199E-11	3.0199E-11	0.077272	3.0199E-11	3.0199E-11	0.077272
$F_{23}$	3.0199E-11	3.0199E-11	6.765E-05	3.0199E-11	3.0199E-11	0.18153
+/-	15/5/3	21/2/0	18/4/1	20/2/1	23/0/0	19/4/0

**Fig. 6** Convergence curve of some functions



**Table 11** Results of comparison with the famous classical algorithms

Function	Metric	IHHO-EO	GA	PSO	DE	BBO
$F_1$	Mean	<b>0</b>	1.03E+03	1.83E+04	1.33E-03	7.59E+01
	Std	0	5.79E+02	3.01E+03	5.92E-04	2.75E+01
$F_2$	Mean	<b>0</b>	2.47E+01	3.58E+02	6.83E-03	1.36E-03
	Std	0	5.68E+00	1.35E+03	2.06E-03	7.45E-03
$F_3$	Mean	<b>0</b>	2.65E+04	4.05E+04	3.97E+04	1.21E+04
	Std	0	3.44E+03	8.21E+03	5.37E+03	2.69E+03
$F_4$	Mean	<b>0</b>	5.17E+01	4.39E+01	1.15E+01	3.02E+01
	Std	0	1.05E+01	3.64E+00	2.37E+00	4.39E+00
$F_5$	Mean	<b>0.049514</b>	1.95E+04	1.96E+07	1.06E+02	1.82E+03
	Std	0.052258	1.31E+04	6.25E+06	1.01E+02	9.40E+02
$F_6$	Mean	<b>0.00040275</b>	9.01E+02	1.87E+04	1.44E-03	6.71E+01
	Std	0.00055358	2.84E+02	2.92E+03	5.38E-04	2.20E+01
$F_7$	Mean	<b>0.00011401</b>	1.91E-01	1.07E+01	5.24E-02	2.91E-03
	Std	0.0001374	1.50E-01	3.05E+00	1.37E-02	1.83E-03
$F_8$	Mean	<b>− 20948.6474</b>	− 1.26E+04	− 3.86E+03	− 6.82E+03	− 1.24E+04
	Std	0.41048	4.51E+00	2.49E+02	3.94E+02	3.50E+01
$F_9$	Mean	<b>0</b>	9.04E+00	2.87E+02	1.58E+02	<b>0.00E+00</b>
	Std	0	4.58E+00	1.95E+01	1.17E+01	0.00E+00
$F_{10}$	Mean	<b>8.8818E-16</b>	1.36E+01	1.75E+01	1.21E-02	2.13E+00
	Std	0	1.51E+00	3.67E-01	3.30E-03	3.53E-01
$F_{11}$	Mean	<b>0</b>	1.01E+01	1.70E+02	3.52E-02	1.46E+00
	Std	0	2.43E+00	3.17E+01	7.20E-02	1.69E-01
$F_{12}$	Mean	<b>1.0635E-05</b>	4.77E+00	1.51E+07	2.25E-03	6.68E-01
	Std	9.6391E-06	1.56E+00	9.88E+06	1.70E-03	2.62E-01
$F_{13}$	Mean	<b>7.9115E-05</b>	1.52E+01	5.73E+07	9.12E-03	1.82E+00
	Std	7.4781E-05	4.52E+00	2.68E+07	1.16E-02	3.41E-01
$F_{14}$	Mean	<b>0.998</b>	<b>9.98E-01</b>	1.39E+00	1.23E+00	<b>9.98E-01</b>
	Std	2.4801E-15	4.52E-16	4.60E-01	9.23E-01	4.52E-16
$F_{15}$	Mean	<b>0.00033615</b>	3.33E-02	1.61E-03	5.63E-04	1.66E-02
	Std	4.5518E-05	2.70E-02	4.60E-04	2.81E-04	8.60E-03
$F_{16}$	Mean	<b>− 1.0316</b>	− 3.78E-01	− 1.03E+00	− 1.03E+00	− 8.30E-01
	Std	1.0508E-10	3.42E-01	2.95E-03	6.78E-16	3.16E-01
$F_{17}$	Mean	<b>0.39789</b>	5.24E-01	4.00E-01	3.98E-01	5.49E-01
	Std	4.0947E-10	6.06E-02	1.39E-03	1.69E-16	6.05E-02
$F_{18}$	Mean	<b>3</b>	<b>3.00E+00</b>	3.10E+00	<b>3.00E+00</b>	<b>3.00E+00</b>
	Std	7.6861E-13	0.00E+00	7.60E-02	0.00E+00	0.00E+00
$F_{19}$	Mean	<b>− 3.8628</b>	− 3.42E+00	− 3.86E+00	− 3.86E+00	− 3.78E+00
	Std	3.1006E-11	3.03E-01	1.24E-03	3.16E-15	1.26E-01
$F_{20}$	Mean	<b>− 3.2863</b>	− 1.61351	− 3.11088	− 3.27048	− 2.70774
	Std	0.055415	0.46049	0.029126	0.058919	0.357832
$F_{21}$	Mean	<b>− 10.1532</b>	− 6.66177	− 4.14764	− 9.64796	− 8.31508
	Std	9.1365E-08	3.732521	0.919578	1.51572	2.883867
$F_{22}$	Mean	<b>− 10.4029</b>	− 5.58399	− 6.01045	− 9.74807	− 9.38408
	Std	9.2175E-08	2.605837	1.962628	1.987703	2.597238
$F_{23}$	Mean	<b>− 10.5364</b>	− 4.69882	− 4.72192	− 10.5364	− 6.2351
	Std	2.4566E-08	3.256702	1.742618	8.88E-15	3.78462

**Table 12** Comparison results with other HHO variants

Function	Metric	IHHO-EO	HHO-SCA	QRHHO	THHO
$F_1$	Mean	<b>0</b>	1.8568E-91	0	3.11E-102
	Std	0	9.48882E-91	0	1.57E-101
$F_2$	Mean	<b>0</b>	2.46034E-51	3.55E-268	2.27E-51
	Std	0	1.11204E-50	0	9.67E-51
$F_3$	Mean	<b>0</b>	8.88008E-72	0	1.50E-77
	Std	0	4.86382E-71	0	8.05E-77
$F_4$	Mean	<b>0</b>	8.01525E-49	2.07E-244	1.34E-50
	Std	0	2.82586E-48	0	5.77E-50
$F_5$	Mean	0.028631	0.014313995	1.16E-01	<b>6.80E-03</b>
	Std	0.032677	0.020188417	1.06E-01	9.17E-03
$F_6$	Mean	<b>9.427E-05</b>	0.000223702	1.25E-03	2.48E-04
	Std	0.00028983	0.000338117	5.39E-04	4.12E-04
$F_7$	Mean	<b>9.6075E-05</b>	0.00012246	8.04E-05	2.02E-04
	Std	9.6785E-05	0.000110326	5.36E-05	2.23E-04
$F_8$	Mean	<b>- 12569.3743</b>	-12569.08113	-1.26E+04	-1.26E+04
	Std	0.066997	0.76690716	6.17E+02	1.04E+02
$F_9$	Mean	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
	Std	0	0	0	0
$F_{10}$	Mean	8.8818E-16	8.88178E-16	<b>8.88E-16</b>	<b>8.88E-16</b>
	Std	0	0	0	0
$F_{11}$	Mean	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
	Std	0	0	0	0
$F_{12}$	Mean	1.0136E-05	1.13E-05	9.37E-05	<b>8.15E-06</b>
	Std	2.0648E-05	1.5E-05	4.81E-05	1.37E-05
$F_{13}$	Mean	<b>7.9115E-05</b>	0.000113	1.42E-03	9.24E-05
	Std	7.4781E-05	0.000166	5.11E-04	1.47E-04
$F_{14}$	Mean	<b>0.998</b>	1.26E+00	1.0643	1.06E+00
	Std	2.4801E-15	4.47E-01	0.2522	2.48E-01
$F_{15}$	Mean	0.00033615	3.45E-04	<b>3.12E-04</b>	3.64E-04
	Std	4.5518E-05	4.03E-05	4.79E-06	3.99E-05
$F_{16}$	Mean	<b>- 1.0316</b>	-1.03E+00	<b>- 1.0316</b>	-1.03E+00
	Std	1.0508E-10	1.80E-09	6.53E-13	2.00E-09
$F_{17}$	Mean	<b>0.39789</b>	3.98E-01	0.398	3.98E-01
	Std	4.0947E-10	2.15E-05	1.29E-06	6.46E-06
$F_{18}$	Mean	<b>3</b>	<b>3.00E+00</b>	<b>3</b>	<b>3.00E+00</b>
	Std	7.6861E-13	9.98E-07	6.09E-07	6.82E-07
$F_{19}$	Mean	<b>- 3.8628</b>	-3.86E+0	-3.86	-3.86E+00
	Std	3.1006E-11	3.00E-03	3.93E-08	1.46E-03
$F_{20}$	Mean	<b>- 3.2863</b>	-3.09E+00	-3.27	-3.16E+00
	Std	0.055415	1.09E-01	7.71E-02	1.18E-01
$F_{21}$	Mean	-10.1532	-5.21E+00	-10.1532	<b>-1.02E+01</b>
	Std	9.1365E-08	8.95E-01	7.18E-06	1.07E-03
$F_{22}$	Mean	<b>- 10.4029</b>	-5.25E+00	-10.2258	-1.04E+01
	Std	9.2175E-08	9.26E-01	0.9704	6.66E-04
$F_{23}$	Mean	<b>- 10.5364</b>	-5.28E+00	<b>- 10.5364</b>	-1.05E+01
	Std	2.4566E-08	8.65E-01	2.60E-06	5.23E-04

**Table 13** Friedman and Quade tests of Exp.2

Algorithms	Friedman	Quade
IHHO-EO	1.9348	1.7736
HHO-SCA	3.0217	3.2989
QRHHO	2.5652	2.5597
THHO	2.4783	2.3768
<i>p</i> -value	0.041482	0.00000568

HHO to illustrate the effectiveness of the IHHO-EO algorithm. The results showed that IHHO-EO algorithm was superior to other optimization algorithms in terms of the obtained best value, mean value and standard deviation. Through the Wilcoxon signed rank test, the *p* value of IHHO-EO in most functions was less than 0.05, indicating that IHHO-EO was significantly different from other algorithms in performance. In addition, IHHO-EO had faster convergence speed and accuracy according to the convergence graphs of the functions. In Exp.2, in order to

illustrate the superior performance of the proposed algorithm, IHHO-EO was compared with several other HHO variants. Friedman test and Quade test were performed, and the results showed that IHHO-EO got the highest ranking for functional experiments among the four HHO variants compared. Exp.3 verified the effectiveness of each improved strategy of IHHO-EO. The results indicated that the mean and standard deviation of the algorithm are improved in most functions after adding the proposed strategies. IHHO-EO combined these strategies and obtained the best performance. In Exp.4, IHHO-EO was compared with several optimization algorithms and one HHO variant on the complex 50-dimensional CEC2017 test functions. Experimental results indicated that IHHO-EO was also effective for complex function optimization problems. Finally, IHHO-EO was applied to the pressure vessel design problem and got the minimum manufacturing cost, which showed that the algorithm also had good competitiveness in the practical engineering design problem with constraints.

**Table 14** Comparison of various improvement strategies ( $F_1 \sim F_{13}$  of 50 dimensions)

Function	Metric	HHO	IHHO1	IHHO2	HHO-EO	IHHO-EO
$F_1$	Mean	3.2564E-93	1.4991E-209	<b>0</b>	1.032E-138	<b>0</b>
	Std	1.7836E-92	0	0	5.5966E-138	0
$F_2$	Mean	1.0592E-49	6.2337E-112	<b>0</b>	2.8466E-86	<b>0</b>
	Std	5.2174E-49	2.1295E-111	0	7.851E-86	0
$F_3$	Mean	1.1368E-70	2.2981E-161	<b>0</b>	8.7762E-68	<b>0</b>
	Std	6.1378E-70	1.1795E-160	0	4.8069E-67	0
$F_4$	Mean	1.6927E-46	4.3819E-97	<b>0</b>	6.9232E-53	<b>0</b>
	Std	9.2707E-46	2.3202E-96	0	2.2983E-52	0
$F_5$	Mean	0.025156	0.038481	<b>0.017575</b>	0.04701	0.049514
	Std	0.049794	0.058184	0.021293	0.05094	0.052258
$F_6$	Mean	0.00031938	0.00033723	<b>0.00021484</b>	0.00028573	0.00040275
	Std	0.00042628	0.00040536	0.0003158	0.00039863	0.00055358
$F_7$	Mean	0.00016787	0.00013888	0.0001195	0.00020249	<b>0.00011401</b>
	Std	0.00026998	0.00017664	0.00011661	0.00017603	0.0001374
$F_8$	Mean	-20948.3397	-20948.6803	-20146.5227	<b>-20949.0944</b>	-20948.6474
	Std	1.1392	0.38484	1258.893	0.045084	0.41048
$F_9$	Mean	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
	Std	0	0	0	0	0
$F_{10}$	Mean	<b>8.8818E-16</b>	<b>8.8818E-16</b>	<b>8.8818E-16</b>	<b>8.8818E-16</b>	<b>8.8818E-16</b>
	Std	0	0	0	0	0
$F_{11}$	Mean	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
	Std	0	0	0	0	0
$F_{12}$	Mean	7.3616E-06	1.4869E-05	1.1499E-05	<b>3.6383E-06</b>	1.0635E-05
	Std	1.4336E-05	2.1029E-05	2.076E-05	2.8377E-06	9.6391E-06
$F_{13}$	Mean	0.00012963	0.00017913	9.684E-05	<b>7.6751E-05</b>	0.00018826
	Std	0.00019134	0.00020107	0.00012243	6.8603E-05	0.00016501

**Table 15** Comparison of various improvement strategies ( $F_{14} \sim F_{23}$ )

Function	Metric	HHO	IHHO1	IHHO2	HHO-EO	IHHO-EO
$F_{14}$	Mean	1.3612	1.3955	1.6581	<b>0.998</b>	<b>0.998</b>
	Std	0.95204	0.55949	1.4949	1.0921E-15	2.4801E-15
$F_{15}$	Mean	0.00034558	0.0003402	0.00050592	<b>0.00032061</b>	0.00033615
	Std	3.4114E-05	3.4179E-05	0.00037236	1.6729E-05	4.5518E-05
$F_{16}$	Mean	<b>-1.0316</b>	<b>-1.0316</b>	<b>-1.0316</b>	<b>-1.0316</b>	<b>-1.0316</b>
	Std	9.1587E-09	1.3097E-10	2.0426E-09	1.5262E-14	1.0508E-10
$F_{17}$	Mean	0.3979	<b>0.39789</b>	<b>0.39789</b>	<b>0.39789</b>	<b>0.39789</b>
	Std	2.1698E-05	4.2938E-07	5.7438E-07	1.0114E-10	4.0947E-10
$F_{18}$	Mean	<b>3</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>3</b>
	Std	6.6727E-07	9.9591E-08	2.5212E-07	8.6292E-14	7.6861E-13
$F_{19}$	Mean	-3.8592	-3.8609	<b>-3.8626</b>	<b>-3.8628</b>	<b>-3.8628</b>
	Std	0.0053535	0.0026166	0.00095738	1.3105E-11	3.1006E-11
$F_{20}$	Mean	-3.1033	-3.1337	-3.1255	-3.2803	<b>-3.2863</b>
	Std	0.11826	0.11371	0.11942	0.053475	0.055415
$F_{21}$	Mean	-5.2122	-5.5285	-10.151	-5.9049	<b>-10.1532</b>
	Std	0.90209	1.5423	0.0046764	1.9324	9.1365E-08
$F_{22}$	Mean	-5.2432	-5.3976	-10.4013	-5.9735	<b>-10.4029</b>
	Std	0.8769	1.8235	0.0031258	2.0147	9.2175E-08
$F_{23}$	Mean	-5.5125	-5.1029	-10.533	-6.0298	<b>-10.5364</b>
	Std	1.7439	0.029196	0.010141	2.0499	2.4566E-08



**Table 16** Results of benchmark functions (CEC2017  $f_1 \sim f_{30}$ )

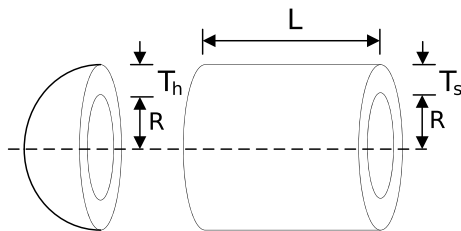
Function	Metric	GOA	DA	SCA	HHO	SCHHO	IHHO-EO
$f_1$	Mean	3.20E+10	3.30E+11	7.38E+09	2.84E+08	7.72E+03	<b>2.88E+03</b>
	Std	1.19E+06	8.90E+8	1.00E+09	1.10E+07	3.31E+01	3.71E+03
$f_3$	Mean	4.76E+05	6.05E+07	1.34E+05	1.37E+05	<b>5.82E+03</b>	8.30E+03
	Std	9.96E+05	1.45E+06	3.44E+05	1.61E+06	1.18E+03	4.70E+03
$f_4$	Mean	4.47E+04	1.26E+04	1.52E+04	7.86E+03	4.31E+02	<b>1.48E+02</b>
	Std	4.16E+03	1.70E+03	9.99E+03	4.14E+02	5.72E+01	5.45E+01
$f_5$	Mean	2.84E+03	8.25E+02	1.22E+03	8.96E+02	5.12E+02	<b>3.21E+02</b>
	Std	5.72E+02	2.10E+02	1.20E+02	8.64E+01	1.29E+01	2.51E+01
$f_6$	Mean	6.92E+02	7.10E+02	6.81E+02	5.72E+02	6.47E+02	<b>2.10E-03</b>
	Std	1.94E+02	2.23E+02	5.74E+02	2.04E+01	8.66E+01	6.50E-03
$f_7$	Mean	<b>2.42E+00</b>	3.86E+01	1.21E+01	6.63E+01	1.83E+01	4.80E+02
	Std	2.80E+02	2.23E+02	5.74E+02	2.40E+01	9.26E+00	5.93E+01
$f_8$	Mean	1.55E+03	1.55E+03	1.41E+03	1.20E+03	1.02E+03	<b>3.28E+02</b>
	Std	8.52E+01	2.33E+01	9.95E+01	3.44E+01	2.03E+01	2.77E+01
$f_9$	Mean	5.98E+04	4.31E+04	2.87E+04	2.77E+04	1.11E+04	<b>1.09E+04</b>
	Std	9.21E+04	6.03E+03	1.22E+03	3.02E+04	2.16E+02	1.18E+03
$f_{10}$	Mean	1.54E+04	1.59E+04	1.54E+04	9.87E+03	<b>3.66E+03</b>	6.65E+03
	Std	3.22E+02	5.23E+02	7.35E+03	6.38E+02	1.62E+02	8.06E+02
$f_{11}$	Mean	3.40E+04	2.43E+04	1.22E+04	1.68E+03	1.46E+04	<b>1.80E+02</b>
	Std	5.52E+03	2.27E+03	6.29E+02	1.88E+02	7.21E+02	3.71E+01
$f_{12}$	Mean	6.05E+10	2.66E+10	2.15E+10	2.48E+08	<b>8.03E+05</b>	1.72E+06
	Std	1.12E+10	2.23E+10	6.73E+10	3.01E+08	7.28E+05	1.02E+06
$f_{13}$	Mean	1.65E+10	1.44E+10	4.55E+09	1.38E+07	3.94E+05	<b>9.29E+03</b>
	Std	1.00E+10	1.20E+09	2.91E+09	3.62E+07	3.73E+05	8.74E+03
$f_{14}$	Mean	7.63E+07	4.23E+07	7.10E+06	2.40E+06	<b>5.61E+03</b>	1.35E+05
	Std	8.28E+06	4.20E+07	1.01E+06	1.60E+06	3.73E+03	1.03E+05
$f_{15}$	Mean	9.89E+08	3.54E+09	9.77E+08	1.20E+06	1.05E+05	<b>1.10E+04</b>
	Std	9.22E+08	3.20E+09	5.21E+06	1.11E+06	6.54E+03	6.93E+03
$f_{16}$	Mean	8.78E+03	5.52E+03	6.20E+03	4.33E+03	3.10E+03	<b>2.14E+03</b>
	Std	1.12E+03	4.33E+03	2.87E+03	3.61E+02	1.59E+02	4.80E+02
$f_{17}$	Mean	1.40E+05	5.38E+03	4.81E+03	3.96E+03	3.12E+03	<b>1.98E+03</b>
	Std	5.10E+02	3.72E+03	2.22E+03	4.27E+02	2.39E+02	3.17E+02
$f_{18}$	Mean	4.27E+07	6.62E+07	3.85E+08	3.78E+06	<b>2.63E+04</b>	6.98E+05
	Std	7.74E+06	3.61E+07	2.74E+08	6.55E+05	2.25E+03	4.77E+05
$f_{19}$	Mean	4.67E+08	1.69E+09	5.77E+08	1.19E+06	5.25E+05	<b>1.99E+04</b>
	Std	3.54E+07	2.63E+08	3.88E+08	1.05E+05	3.05E+05	1.11E+04
$f_{20}$	Mean	4.92E+03	3.36E+03	4.20E+03	3.67E+03	2.42E+03	<b>1.26E+03</b>
	Std	3.34E+03	1.90E+02	2.11E+02	2.83E+03	2.62E+02	2.45E+02
$f_{21}$	Mean	3.32E+03	2.53E+03	2.69E+03	2.88E+03	1.60E+03	<b>6.16E+02</b>
	Std	1.14E+02	6.15E+01	8.00E+02	1.33E+01	5.80E+01	5.96E+01
$f_{22}$	Mean	1.83E+04	1.31E+04	1.84E+04	1.11E+04	<b>2.51E+03</b>	7.45E+03
	Std	3.64E+03	7.75E+03	3.34E+03	4.71E+03	5.12E+02	9.49E+02
$f_{23}$	Mean	3.66E+03	3.61E+03	3.97E+03	3.86E+03	1.99E+03	<b>1.13E+03</b>
	Std	6.04E+01	4.02E+01	4.74E+01	5.81E+01	7.01E+01	2.13E+02
$f_{24}$	Mean	3.42E+03	4.26E+03	3.88E+03	4.41E+03	5.70E+03	<b>2.10E+03</b>
	Std	2.46E+02	4.00E+02	3.09E+02	6.41E+02	2.66E+02	4.40E+02
$f_{25}$	Mean	2.34E+04	7.52E+03	8.63E+03	3.25E+03	1.87E+03	<b>5.70E+02</b>
	Std	4.73E+02	2.57E+02	1.36E+01	3.02E+01	2.26E+01	2.82E+01
$f_{26}$	Mean	3.02E+04	1.16E+04	1.25E+04	1.07E+04	5.95E+03	<b>3.00E+02</b>
	Std	1.53E+04	2.11E+03	2.38E+03	6.12E+03	1.96E+03	1.19E-01
$f_{27}$	Mean	5.98E+03	5.60E+03	4.80E+03	4.88E+03	3.40E+03	<b>8.22E+02</b>

**Table 16** (continued)

Function	Metric	GOA	DA	SCA	HHO	SCHHO	IHHO-EO
$f_{28}$	Std	4.73E+02	8.01E+02	3.64E+03	4.02E+02	1.62E+02	1.36E+02
	Mean	1.22E+04	7.01E+03	8.80E+03	3.65E+03	3.13E+03	<b>5.02E+02</b>
$f_{29}$	Std	6.78E+02	2.75E+01	5.83E+01	3.55E+02	1.18E+02	2.37E+01
	Mean	1.99E+04	3.25E+04	8.38E+03	6.08E+03	5.11E+03	<b>1.27E+03</b>
$f_{30}$	Std	1.08E+03	2.82E+03	1.33E+02	4.93E+02	3.13E+02	3.07E+02
	Mean	7.94E+09	2.04E+09	1.53E+09	6.20E+07	<b>1.17E+05</b>	9.10E+05
	Std	7.58E+09	6.89E+08	6.36E+08	2.46E+06	6.25E+04	2.47E+05
IHHO-EO:Better/Worse		28/1	29/0	29/0	29/0	22/7	

**Table 17** Comparison of results for pressure vessel design problem

Algorithm	$x_1$	$x_2$	$x_3$	$x_4$	Optimal cost
IHHO-EO	<b>0.7833929</b>	<b>0.3872152</b>	<b>40.58359</b>	<b>196.3577</b>	<b>5895.1713</b>
IEHHO	0.822896	0.406769	42.63711	170.0741	5966.301
HHO	0.81758383	0.4072927	42.09174576	176.7196352	6000.46259
GWO	0.8125	0.4345	42.089181	176.758731	6051.5639
GA	0.812500	0.437500	42.097398	176.654050	6059.9463
HPSO	0.812500	0.437500	42.0984	176.6366	6059.7143
BA	0.812500	0.437500	42.098445	176.636595	6059.7143
WOA	0.812500	0.437500	42.0982699	176.638998	6059.7410

**Fig. 7** Pressure vessel design problem

In future work, IHHO-EO can be applied for more engineering design problems, such as compression spring design problem, welded beam design problem [54], and typical combinatorial optimization problems such as traveling salesman problem [23].

**Acknowledgements** This work was supported by National Natural Science Foundation of China (Grant Nos. 61872153 and 61972288).

**Author Contributions** H. Zhang: Conceptualization, Methodology, Software, Data curation, Formal analysis, Writing-original draft M. Chen and P. Li: Conceptualization, Methodology, Formal analysis, Investigation, Writing-review & editing, Supervision J. Junjie: Conceptualization, Methodology, Formal analysis

## Declarations

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

## References

1. Abd Elaziz M, Oliva D, Xiong S (2017) An improved opposition-based sine cosine algorithm for global optimization. *Expert Syst Appl* 90:484–500
2. Al-Betar MA, Awadallah MA, Heidari AA, Chen H, Al-Khraisat H, Li C (2021) Survival exploration strategies for Harris hawks optimizer. *Expert Syst Appl* 168:114243
3. Arora S, Singh S (2019) Butterfly optimization algorithm: a novel approach for global optimization. *Soft Comput* 23:715–734
4. Bak P, Sneppen K (1993) Punctuated equilibrium and criticality in a simple model of evolution. *Phys Rev Lett* 71:4083
5. Bak P, Tang C, Wiesenfeld K (1988) Self-organized criticality. *Phys Rev A* 38:364
6. Balamurugan R, Ratheesh S, Venila YM (2021) Classification of heart disease using adaptive harris hawk optimization-based clustering algorithm and enhanced deep genetic algorithm. *Soft Comput* 1–17
7. Boettcher S, Percus A (2000) Nature's way of optimizing. *Artif Intell* 119:275–286
8. Chen H, Heidari AA, Chen H, Wang M, Pan Z, Gandomi AH (2020) Multi-population differential evolution-assisted harris hawks optimization: framework and case studies. *Fut Gen Comput Syst* 111:175–198

9. Chen MR, Huang YY, Zeng GQ, Lu KD, Yang LQ (2021) An improved bat algorithm hybridized with extremal optimization and Boltzmann selection. *Expert Syst Appl* 175:114812
10. Chen MR, Li X, Zhang X, Lu YZ (2010) A novel particle swarm optimizer hybridized with extremal optimization. *Appl Soft Comput* 10:367–373
11. Chen MR, Lu YZ (2008) A novel elitist multiobjective optimization algorithm: multiobjective extremal optimization. *Eur J Oper Res* 188:637–651
12. Chen MR, Lu YZ, Yang G (2006) Population-based extremal optimization with adaptive lévy mutation for constrained optimization. In: *International Conference on Computational and Information Science*, Springer, New York, pp 144–155
13. Dehkordi AA, Sadiq AS, Mirjalili S, Ghafoor KZ (2021) Nonlinear-based chaotic harris hawks optimizer: algorithm and internet of vehicles application. *Appl Soft Comput* 107574
14. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. *J Mach Learn Res* 7:1–30
15. Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol Comput* 1:3–18
16. Dokeroglu T, Sevinc E, Kucukyilmaz T, Cosar A (2019) A survey on new generation metaheuristic algorithms. *Comput Ind Eng* 137:106040
17. Fan Q, Chen Z, Li Z, Xia Z, Lin Y (2020a) An efficient refracted salp swarm algorithm and its application in structural parameter identification. *Eng Comput* 1–15
18. Fan Q, Chen Z, Xia Z (2020) A novel quasi-reflected harris hawks optimization algorithm for global optimization problems. *Soft Comput* 24:14825–14843
19. Gandomi AH, Yang XS, Alavi AH, Talatahari S (2013) Bat algorithm for constrained optimization tasks. *Neural Comput Appl* 22:1239–1255
20. García S, Fernández A, Luengo J, Herrera F (2010) Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power. *Inf Sci* 180:2044–2064
21. Gharehchopogh FS, Abdollahzadeh B (2021) An efficient harris hawk optimization algorithm for solving the travelling salesman problem. *Cluster Comput* 1–25
22. Gölcük İ, Ozsoydan FB (2021) Quantum particles-enhanced multiple harris hawks swarms for dynamic optimization problems. *Expert Syst Appl* 167:114202
23. Gutin G, Punnen AP (2006) *The traveling salesman problem and its variations*, vol 12. Springer, New York
24. Hakli H, Kiran MS (2020) An improved artificial bee colony algorithm for balancing local and global search behaviors in continuous optimization. *Int J Mach Learn Cybern* 11:2051–2076
25. He F, Liu H, Liu C, Bao G (2021) Analysis of radar technology identification model for potential geologic hazard based on convolutional neural network and harris hawks optimization algorithm. *Soft Comput* pp 1–15
26. He Q, Wang L (2007) A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. *Appl Math Comput* 186:1407–1422
27. He Y, Wang X (2021) Group theory-based optimization algorithm for solving knapsack problems. *Knowl-Based Syst* 219:104445
28. He Y, Wang X, Gao S (2019) Ring theory-based evolutionary algorithm and its application to d {0-1} kp. *Appl Soft Comput* 77:714–722
29. He Y, Zhang F, Mirjalili S, Zhang T (2022) Novel binary differential evolution algorithm based on taper-shaped transfer functions for binary optimization problems. *Swarm Evol Comput* 69:101022
30. Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019) Harris hawks optimization: algorithm and applications. *Fut Gen Comput Syst* 97:849–872
31. Holland JH (1992) *Genetic algorithms*. *Sci Am* 267:66–73
32. Hussain K, Neggaz N, Zhu W, Houssein EH (2021) An efficient hybrid sine-cosine harris hawks optimization for low and high-dimensional feature selection. *Expert Syst Appl* 176:114778
33. Hussien AG, Amin M (2021) A self-adaptive harris hawks optimization algorithm with opposition-based learning and chaotic local search strategy for global optimization and feature selection. *Int J Mach Learn Cybern* 1–28
34. Jiang Q, Shao F, Lin W, Gu K, Jiang G, Sun H (2017) Optimizing multistage discriminative dictionaries for blind image quality assessment. *IEEE Trans Multimed* 20:2035–2048
35. Kamboj VK, Nandi A, Bhadoria A, Sehgal S (2020) An intensify harris hawks optimizer for numerical and engineering optimization problems. *Appl Soft Comput* 89:106018
36. Kannan BK, Kramer SN (1994) An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. *J Mech Des* 116:405–411
37. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: *Proceedings of ICNN'95-international conference on neural networks*, IEEE, pp 1942–1948
38. Li C, Li J, Chen H, Jin M, Ren H (2021) Enhanced harris hawks optimization with multi-strategy for global optimization tasks. *Expert Syst Appl* 185:115499
39. Liang JJ, Qu BY, Suganthan PN (2013) 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* 635:490
40. Long W, Wu T, Cai S, Liang X, Jiao J, Xu M (2019) A novel grey wolf optimizer algorithm with refraction learning. *IEEE Access* 7:57805–57819
41. Luo J, Chen MR (2014) Improved shuffled frog leaping algorithm and its multi-phase model for multi-depot vehicle routing problem. *Expert Syst Appl* 41:2535–2545
42. McCarthy J (1989) Block-conjugate-gradient method. *Phys Rev D* 40:2149
43. Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. *Knowl-Based Syst* 89:228–249
44. Mirjalili S (2016) Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput Appl* 27:1053–1073
45. Mirjalili S (2016) Sca: a sine cosine algorithm for solving optimization problems. *Knowl-Based Syst* 96:120–133
46. Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM (2017) Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. *Adv Eng Softw* 114:163–191
47. Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67
48. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61
49. Nocedal J, Wright S (2006) *Numerical optimization*. Springer, New York
50. Qu C, He W, Peng X, Peng X (2020) Harris hawks optimization with information exchange. *Appl Math Model* 84:52–75
51. Rao RV, Savsani VJ, Vakharia D (2011) Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. *Comput Aided Des* 43:303–315
52. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) Gsa: a gravitational search algorithm. *Inf Sci* 179:2232–2248

53. Rodríguez-Esparza E, Zanella-Calzada LA, Oliva D, Heidari AA, Zaldivar D, Pérez-Cisneros M, Foong LK (2020) An efficient harris hawks-inspired image segmentation method. *Expert Syst Appl* 155:113428
54. Sadollah A, Bahreininejad A, Eskandar H, Hamdi M (2013) Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems. *Appl Soft Comput* 13:2592–2612
55. Saremi S, Mirjalili S, Lewis A (2017) Grasshopper optimisation algorithm: theory and application. *Adv Eng Softw* 105:30–47
56. Simon D (2008) Biogeography-based optimization. *IEEE Trans Evol Comput* 12:702–713
57. Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Global Optim* 11:341–359
58. Tanyildizi E, Demir G (2017) Golden sine algorithm: a novel math-inspired algorithm. *Adv Electr Comput Eng* 17:71–78
59. Tizhoosh HR (2005) Opposition-based learning: a new scheme for machine intelligence. In: *International conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (CIMCA-IAWTIC'06)*, IEEE, pp 695–701
60. Van Laarhoven PJ, Aarts EH (1987) Simulated annealing. In: *Simulated annealing: theory and applications*. Springer, pp 7–15
61. Wang G, Jin J (2019) Moth-flame optimization algorithm fused on refraction principle and opposite-based learning. *Comput Eng Appl* 55:46–51
62. Wang R, Zhang Z, Ng WW, Wu W (2021) An improved group theory-based optimization algorithm for discounted 0–1 knapsack problem. *Adv Comput Intell* 1:1–11
63. Wolpert D, Macready W (1997) No free lunch theorems for optimization. *IEEE Trans Evol Comput* 1:67–82
64. Wu G (2016) Across neighborhood search for numerical optimization. *Inf Sci* 329:597–618
65. Xiao C, Yu M, Wang H, Zhang B, Wang D (2021) Prognosis of electric scooter with intermittent faults: dual degradation processes approach. *IEEE Trans Veh Technol* 71:1411–1425
66. Yang XS (2010a) Firefly algorithm, levy flights and global optimization. In: *Research and development in intelligent systems XXVI*, Springer, pp 209–218
67. Yang XS (2010b) *Nature-inspired metaheuristic algorithms*. Luniver Press
68. Yao X, Liu Y, Lin G (1999) Evolutionary programming made faster. *IEEE Trans Evol Comput* 3:82–102
69. Yu M, Xiao C, Zhang B (2020) Event-triggered discrete component prognosis of hybrid systems using degradation model selection. *IEEE Trans Industr Electron* 68:11470–11481
70. Yu X, Xu W, Li C (2021) Opposition-based learning grey wolf optimizer for global optimization. *Knowl-Based Syst* 226:107139
71. Zeng GQ, Xie XQ, Chen MR, Weng J (2019) Adaptive population extremal optimization-based pid neural network for multivariable nonlinear control systems. *Swarm Evol Comput* 44:320–334

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.