Mathematical Biosciences and Engineering

MBE, 18(6): 7076–7109.
DOI: 10.3934/mbe.2021352
Received: 26 June 2021
Accepted: 20 August 2021
Published: 24 August 2021

http://www.aimspress.com/journal/MBE

Research article

An improved hybrid Aquila Optimizer and Harris Hawks Optimization for global optimization

Shuang Wang[1], Heming Jia[1,*], Qingxin Liu[2] and Rong Zheng[1]

[1] School of Information Engineering, Sanming University, Sanming 365004, Fujian, China
[2] School of Computer Science and Technology, Hainan University, Haikou 570228, Hainan, China

* **Correspondence:** Email: jiahemінglucky99@126.com.

Abstract: This paper introduces an improved hybrid Aquila Optimizer (AO) and Harris Hawks Optimization (HHO) algorithm, namely IHAOHHO, to enhance the searching performance for global optimization problems. In the IHAOHHO, valuable exploration and exploitation capabilities of AO and HHO are retained firstly, and then representative-based hunting (RH) and opposition-based learning (OBL) strategies are added in the exploration and exploitation phases to effectively improve the diversity of search space and local optima avoidance capability of the algorithm, respectively. To verify the optimization performance and the practicability, the proposed algorithm is comprehensively analyzed on standard and CEC2017 benchmark functions and three engineering design problems. The experimental results show that the proposed IHAOHHO has more superior global search performance and faster convergence speed compared to the basic AO and HHO and selected state-of-the-art meta-heuristic algorithms.

Keywords: aquila optimizer; harris hawks optimization; hybrid algorithm; representative-based hunting; opposition-based learning

1. Introduction

Meta-heuristic optimization algorithms develop rapidly [1–3] because of its simple concept, flexibility and ability to avoid local optima, and have been widely used in solving various complex optimization problems in the real world [4,5]. According to different inspiration of the algorithms, meta-heuristics can be divided into three main categories: evolutionary, physics-based and swarm intelligence based techniques. The inspirations of evolutionary algorithms are the laws of evolution in

nature. There are some representative evolutionary algorithms such as Genetic Algorithm (GA) [6], Differential Evolution Algorithm (DE) [7], Evolution Strategy (ES) [8], Biogeography-Based Optimizer (BBO) [9] and Probability-Based Incremental Learning (PBIL) [10]. Inspired by the physical rules of the universe, physics-based techniques include Simulated Annealing (SA) [11], Gravity Search Algorithm (GSA) [12], Black Hole Algorithm (BH) [13], Multi-Verse Optimizer (MVO) [14], Sine Cosine Algorithm (SCA) [15], Arithmetic Optimization Algorithm (AOA) [16], Heat Transfer Relation-based Optimization Algorithm (HTOA) [17] and so forth. Swarm intelligence (SI) based methods belong to the most popular category, which are inspired by swarm behaviors of creatures in nature. The representative SI algorithms include Particle Swarm Optimization (PSO) [18], Ant Colony Optimization Algorithm (ACO) [19], Firefly Algorithm (FA) [20], Grey Wolf Optimizer (GWO) [21], Cuckoo Search Algorithm (CS) [22], Whale Optimization Algorithm (WOA) [23], Salp Swarm Algorithm (SSA) [24], Remora Optimization Algorithm [25], Slime Mould Algorithm (SMA) [26], and Horse herd Optimization Algorithm (HOA) [27].

The Aquila Optimizer (AO) [28] and Harris Hawks Optimization (HHO) [29] are both latest SI algorithms that simulate hunting behaviors of Aquila and Harris' hawks respectively. Due to the short time for AO to be proposed, there is no research on the improvement of AO yet, but AO has been used to solve the real-world optimization problems. AlRassas et al. [30] applied AO to optimize parameters of Adaptive Neuro-Fuzzy Inference System (ANFIS) model to boost the prediction accuracy of oil production forecasting. This research revels the good practicable performance of AO. For another thing, once the HHO was proposed, it attracted a large number of researchers to improve or apply it to solve optimization problems in many fields. Chen et al. [31] proposed the first powerful variant of HHO by integrating chaos, topological multi-population, and differential evolution (DE) strategies. Chaos mechanism is for exploitation, multi-population strategy is for global search ability, and the DE mechanism is for increasing the accuracy of the solutions. Inspired by the survival-of-the-fittest principle of evolutionary algorithms, Al-Betar et al. [32] proposed three new versions of HHO incorporated tournament, proportional and linear rank-based strategies respectively to accelerate convergence. The proposed new versions show a better balance between the exploration and exploitation and enhance local optima avoidance as well. Song et al. [33] utilized dimension decision strategy in CS to improve the convergence speed, and Gaussian mutation to increase the convergence accuracy and premature convergence avoidance. Yousri et al. [34] improved the exploration performance of HHO using the fractional calculus (FOC) memory concept. The hawks move with a fractional-order velocity, and the escaping energy of prey is adaptively adjusted based on FOC parameters to avoid local optima stagnation. Gupta et al. [35] enhanced the search-efficiency and premature convergence avoidance of HHO by adding a nonlinear energy parameter, different settings for rapid dives, opposition-based learning strategy and a greedy selection mechanism. Akdag et al. [36] introduced seven types of random distribution functions to increase the performance of HHO, and then applied the modified HHO to solve optimum power flow (OPF) problem. Yousri et al. [37] applied HHO to optimize parameters of the Proportional-Integral controller for designing load frequency control (LFC). Jia et al. [38] proposed a dynamic HHO using a mutation mechanism to avoid local optima and enhance the search capability. This improved HHO was applied for satellite image segmentation as well.

Otherwise, there are also attempts of hybrid algorithm of HHO. Hussain et al. [39] integrated sine-cosine algorithm (SCA) in HHO for numerical optimization and feature selection. The SCA integration is used to cater ineffective exploration in HHO, moreover exploitation is enhanced by

dynamically adjusting candidate solutions for avoiding solution stagnancy in HHO. Bao et al. [40] proposed HHO-DE by hybridizing HHO and DE algorithms. The convergence accuracy, ability to avoid local optima and stability are greatly improved compared to HHO and DE. Houssein et al. [41] proposed a hybrid algorithm called CHHO-CS by combining HHO with CS and chaotic maps. The CHHO-CS achieves a better balance between exploration and exploitation phases, and effectively avoids premature convergence. Kaveh et al. [42] hybridized HHO with Imperialist Competitive Algorithm (ICA). Combination of the exploration strategy of ICA and exploitation technique of HHO helps to achieve a better search performance. The satisfactory outcomes of several HHO-based hybrid algorithms proposed in the literature show potential research direction.

Thus, in view of defects in the slow convergence and local optima stagnation of HHO and inspired by the above researches, we try a hybridization to enhance the performance of HHO and AO. An improved hybrid Aquila Optimizer and Harris Hawks Optimization namely IHAOHHO is proposed. First of all, we combine the exploration phase of AO with the exploitation phase of HHO together. This operation extracts and retains the strong exploration and exploitation capabilities of basic AO and HHO. Then, in order to further improve the performance of IHAOHHO, the representative-based hunting (RH) and opposition-based learning (OBL) strategies are introduced into IHAOHHO. RH is mixed into the exploration phase to increase the diversification and OBL is added into the exploitation phase to avoid local optima stagnation, respectively. Thus, the capabilities of exploration, exploitation and local optima avoidance are effectively enhanced in the proposed algorithm. The standard and CEC2017 benchmark functions and three engineering design problems are utilized to test the exploration and exploitation capabilities of IHAOHHO. The proposed algorithm is compared with basic AO, HHO, and several well-known meta-heuristic algorithms, including HOA, SSA, WOA, GWO, MVO, IPOP-CMA-ES [43], LSHADE [44], Sine-cosine and Spotted Hyena-based Chimp Optimization Algorithm (SSC) [45] and RUNge Kutta Optimizer (RUN) [46]. The experimental results show that the proposed IHAOHHO algorithm outperforms other state-of-the-art algorithms.

The rest of this paper is organized as follows: The Section 2, provides a brief overview of the related work: basic AO and HHO algorithms, as well as the RH and OBL strategies. The Section 3, describes in detail the proposed hybrid algorithm. The Section 4, conducts simulation experiments and results analysis. Finally, Section 5, concludes the paper.

2. Preliminaries

2.1. Aquila Optimizer (AO)

AO is a latest novel swarm intelligence algorithm proposed by Abualigah et al. in 2021. There are four hunting behaviors of Aquila for different kinds of prey. Aquila can switch hunting strategies flexibly for different prey, and then uses its fast speed united with sturdy feet and claws to attack prey. The brief description of mathematical model can be described as follows.

Step 1: Expanded exploration: high soar with a vertical stoop

In this method, the Aquila flies high over the ground and explorers the search space widely, and then a vertical dive will be taken once the Aquila determines the area of the prey. The mathematical representation of this behavior is written as:

$$X(t+1) = X_{best}(t) \times (1 - \frac{t}{T}) + (X_M(t) - X_{best}(t) \times rand) \tag{1}$$

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

where $X_{best}(t)$ represents the best position obtained so far, and $X_M(t)$ denotes to the average position of all Aqulias in current iteration. t and T are the current iteration and the maximum number of iterations, respectively. N is the population size and $rand$ is a random number between 0 and 1.

Step 2: Narrowed exploration: contour flight with short glide attack

This is the most commonly used hunting method for Aquila. It uses short gliding to attack the prey after descending within the selected area and flying around the prey. The position update formula is represented as:

$$X(t+1)=X_{best}(t)\times LF(D)+X_R(t)+(y-x)\times rand \tag{3}$$

where, $X_R(t)$ represents a random position of the hawk, and D is the dimension size. $LF(D)$ represents Levy flight function, which is presented as follows:

$$LF(D)=s\times\frac{u\times\sigma}{|v|^{\frac{1}{\beta}}} \tag{4}$$

$$\sigma=\left(\frac{\Gamma(1+\beta)\times\sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2})\times\beta\times2^{(\frac{\beta-1}{2})}}\right) \tag{5}$$

where, s and β are constant values equal to 0.01 and 1.5 respectively, u and v are random numbers between 0 and 1. y and x are used to present the spiral shape in the search, which are calculated as follows:

$$\begin{cases}x=r\times\sin(\theta)\\ y=r\times\cos(\theta)\\ r=r_1+0.00565\times D_1\\ \theta=-\omega\times D_1+\frac{3\times\pi}{2}\end{cases} \tag{6}$$

where, r_1 means the number of search cycles between 1 and 20, D_1 is composed of integer numbers from 1 to the dimension size (D), and ω is equal to 0.005.

Step 3: Expanded exploitation: low flight with a slow descent attack

In the third method, when the area of prey is roughly determined, the Aquila descends vertically to do a preliminary attack. AO exploits the selected area to get close to and attack the prey. This behavior is presented as follows:

$$X(t+1)=(X_{best}(t)-X_M(t))\times\alpha-rand+((UB-LB)\times rand+LB)\times\delta \tag{7}$$

where α and δ are the exploitation adjustment parameters fixed to 0.1, UB and LB are the upper and lower bounds of the problem.

Step 4: Narrowed exploitation: walking and grab prey

In this method, the Aquila chases the prey in the light of its escape trajectory, and then attacks the prey on the ground. The mathematical representation of this behavior is as follows:

$$\begin{cases} X(t+1) = QF \times X_{best}(t) - (G_1 \times X(t) \times rand) \\ \qquad\qquad - G_2 \times LF(D) + rand \times G_1 \\ QF(t) = t^{\frac{2 \times rand - 1}{(1-T)^2}} \\ G_1 = 2 \times rand - 1 \\ G_2 = 2 \times (1 - \frac{t}{T}) \end{cases} \tag{8}$$

where $X(t)$ is the current position, and $QF(t)$ represents the quality function value which used to balance the search strategy. G_1 denotes the movement parameter of Aquila during tracking prey, which is a random number between [-1,1]. G_2 denotes the flight slope when chasing prey, which decreases linearly from 2 to 0.

The flowchart of AO is shown in Figure 1.

Figure1. AO algorithm flowchart.

2.2. Harris Hawks Optimization(HHO)

HHO is a new meta-heuristic optimization algorithm proposed by Heidari et al. in 2019. It is inspired by the unique cooperative foraging activities of Harris' hawk. Harris' hawk can show a variety of chasing patterns according to the dynamic nature of the environment and the escaping patterns of the prey. These switching activities are conducive to confuse the running prey, and the cooperative strategies can help Harris' hawk chase the detected prey to exhaustion, which increases its vulnerability. The brief description of mathematical model is as follows.

2.2.1. Exploration phase

The Harris' hawks usually perch on some random locations, wait and monitor the desert to detect the prey. There are two perching strategies based on the positions of other family members and the prey, which are selected in accordance with the random q value.

$$X(t+1)=\begin{cases} X_R(t)-rand\,|\,X_r(t)-2rand\times X(t)\,| & \text{q}\geq 0.5 \\ (X_{best}(t)-X_M(t))-rand(LB+rand(UB-LB)) & \text{q}<0.5 \end{cases} \tag{9}$$

where q is random number between 0 and 1.

2.2.2. Transition from exploration to exploitation phase

The HHO algorithm has a transition mechanism from exploration to exploitation phase based on the escaping energy of the prey, and then changes the different exploitative behaviors. The energy of the prey is modeled as follows, which decreases during the escaping behavior.

$$E=2E_0(1-\frac{t}{T}) \tag{10}$$

where E represents the escaping energy of the prey, E_0 is the initial state of the energy. When $|E|\geq 1$, the algorithm performs the exploration stage, and when $|E|<1$, the algorithm performs the exploitation phase.

2.2.3. Exploitation phase

In this phase, four different chasing and attacking strategies are proposed on the basis of the escaping energy of the prey and chasing styles of the Harris' hawks. Except for the escaping energy, parameter r is also utilized to choose the chasing strategy, which indicates the chance of the prey in successfully escaping ($r<0.5$) or not ($r\geq 0.5$) before attack.

i. Soft besiege

When $r\geq 0.5$ and $|E|\geq 0.5$, the prey still has enough energy and tries to escape, so the Harris' hawks encircle it softly to make the prey more exhausted and then attack it. This behavior is modeled as follows:

$$X(t+1)=\Delta X(t)-E\,|\,JX_{best}(t)-X(t)\,| \tag{11}$$

$$\Delta X(t) = X_{best}(t) - X(t) \tag{12}$$

$$J = 2(1 - rand) \tag{13}$$

where $\Delta X(t)$ indicates the difference between the position of prey and the current position, J represents the random jump strength of prey.

ii. Hard besiege

When $r \geq 0.5$ and $|E| < 0.5$, the prey has a low escaping energy, and the Harris' hawks encircle the prey readily and finally attack it. In this situation, the positions are updated as follows:

$$X(t+1) = X_{best}(t) - E|\Delta X(t)| \tag{14}$$

iii. Soft besiege with progressive rapid dives

When $|E| \geq 0.5$ and $r < 0.5$, the prey has enough energy to successfully escape, so the Harris' hawks perform soft besiege with several rapid dives around the prey and try to progressively correct its position and direction. This behavior is modeled as follows:

$$Y = X_{best}(t) - E|JX_{best}(t) - X(t)| \tag{15}$$

$$Z = Y + S \times LF(D) \tag{16}$$

$$X(t+1) = \begin{cases} Y & if \quad F(Y) < F(X(t)) \\ Z & if \quad F(Z) < F(X(t)) \end{cases} \tag{17}$$

where S is a random vector. Note that, only the better position between Y and Z is selected as the next position.

iv. Hard besiege with progressive rapid dives

When $|E| < 0.5$ and $r < 0.5$, the prey has no enough energy to escape, so the hawks perform a hard besiege to decrease the distance between their average position and the prey, and finally attack and kill the prey. The mathematical representation of this behavior is as follows:

$$Y = X_{best}(t) - E|JX_{best}(t) - X_M(t)| \tag{18}$$

$$Z = Y + S \times LF(D) \tag{19}$$

$$X(\mathrm{t}+1) = \begin{cases} Y & if \quad F(Y) < F(\mathrm{X}(t)) \\ Z & if \quad F(Z) < F(X(t)) \end{cases} \tag{20}$$

Note that only the better position between Y and Z will be the next position for the new iteration. The flowchart of HHO is displayed in Figure 2.

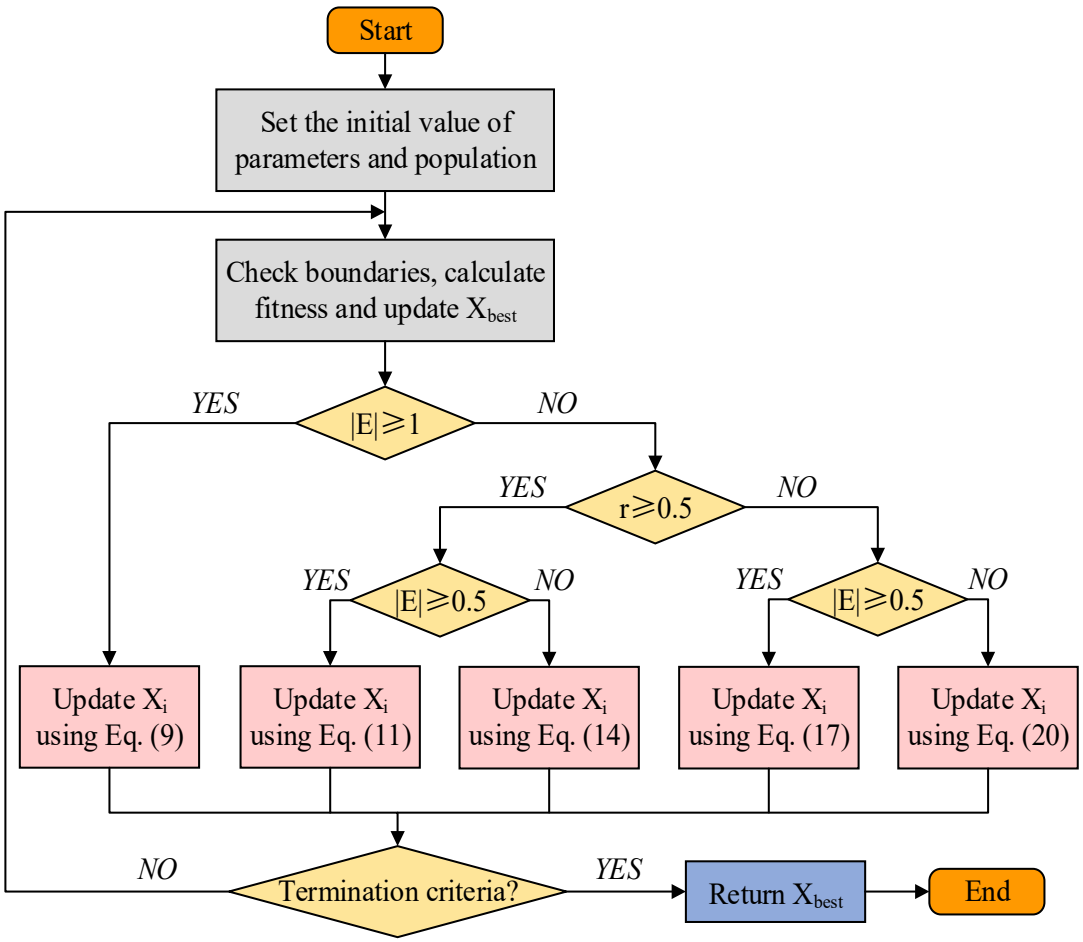


Figure 2. HHO algorithm flowchart.

2.3. Representative-based hunting(RH)

The strategy of representative-based hunting was first proposed to improve the exploration and diversification of GWO algorithm in 2021 [47]. To achieve this, an archive called representative archive (RA) is constructed to maintain the representative solutions. A random representative search agent is selected from the five-best search agents archived by the RA, and a random search agent is selected from the RA. Meanwhile, two random search agents are selected from the population. These four selections efficiently improve the diversity, exploration capability and premature convergence avoidance. The mathematical model of RH is as follows:

$$X(t+1)=X_{R\_best}+cd\times(X(t)-X_{R\_archive})+\sigma\times(X_{rand1}-X_{rand2}) \tag{21}$$

where X_{R_best} and $X_{R\_archive}$ are randomly selected from the five-best representative search agents and the whole archive, respectively. X_{rand1} and X_{rand2} are randomly selected from the whole population. σ and the Cauchy distribution cd are calculated by:

$$\sigma=\left(\frac{T-t}{T-1}\right)^{Exponent}\times(\sigma_{initial}-\sigma_{final})+\sigma_{final} \tag{22}$$

$$cd = z_0 + 0.1 \times \tan(\pi(rand - 0.5)) \tag{23}$$

σ is a nonlinearly decreasing parameter form 1 to 0, which leads to decrease the exploration to exploitation over the course of iterations, forming a transition from exploration to exploitation. The cd coefficient assists to enhance the random behavior, preferring exploration and escaping from the local optima.

2.4. Opposition-based learning(OBL)

Opposition-based learning (OBL) is a powerful optimization tool proposed by Tizhoosh in 2005 [48]. The main idea of OBL is simultaneously considering the fitness of an estimate and its corresponding counter estimate to obtain a better candidate solution (Figure 3). An optimization process usually starts at a random initial solution. If the random solution is near the optimal solution, the algorithm converges fast. However, it's possible that the initial solution is far away from the optimum or just at exact opposite position. In this case, it might take quite long time to converge or not converge at all. Thus, considering the opposite direction of the candidate solution in each step increases the probability of finding a better solution. We can choose the opposite point as the candidate solution once the opposite solution is beneficial and then proceed to the next iteration. The OBL concept has successfully been used in varieties of meta-heuristics algorithms [49–53] to improve the convergence speed. OBL is defined by:

$$x_{jOBL} = l_j + u_j - x_j, \quad j = 1, 2, \ldots, n \tag{24}$$

where x_{jOBL} represents the opposite solution, l_j and u_j are the lower and upper bounds of the problem in j^{th} dimension. The opposite solution described by Eq (24) can effectively help the population jump out of the local optima.

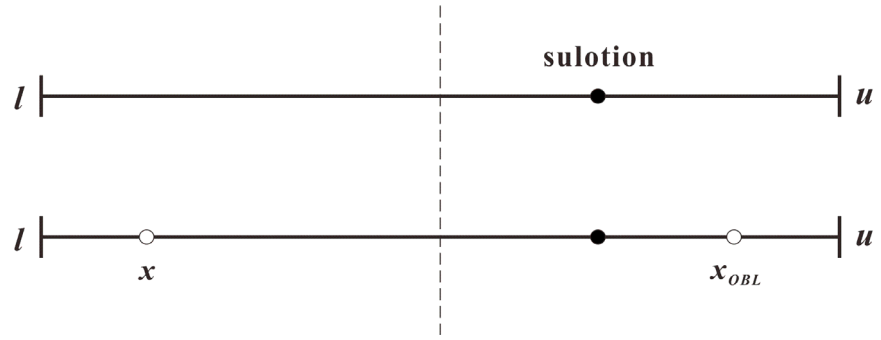


Figure 3. Diagram of OBL.

3. The proposed IHAOHHO algorithm

3.1. The detail design of IHAOHHO

The AO simulates hunting behaviors for fast-moving prey within a wide flying area in exploration phase. The characteristics of these behaviors make AO have a strong global search ability and fast

convergence speed. However, the selected search space is not exhaustively searched during the exploitation phase. The role of Levy flight is weak in the late iterations, which tends to result in premature convergence. Thus, the AO algorithm possesses good exploration capability and fast convergence speed, but it is hard to escape from local optima in the exploitation stage. For the HHO algorithm, the experimental results show that deficiencies of insufficient diversification of the population and low convergence speed exist in the exploration phase. On the basis of the energy and escape probability of the prey, four different hunting strategies are used to implement various position updating methods in the exploitation phase. In addition, the transition mechanism from exploration to exploitation is a good way to adapt to animal characteristics. As a whole, the energy of prey decreases with the increase of iterations, making the algorithm enter the exploitation stage.

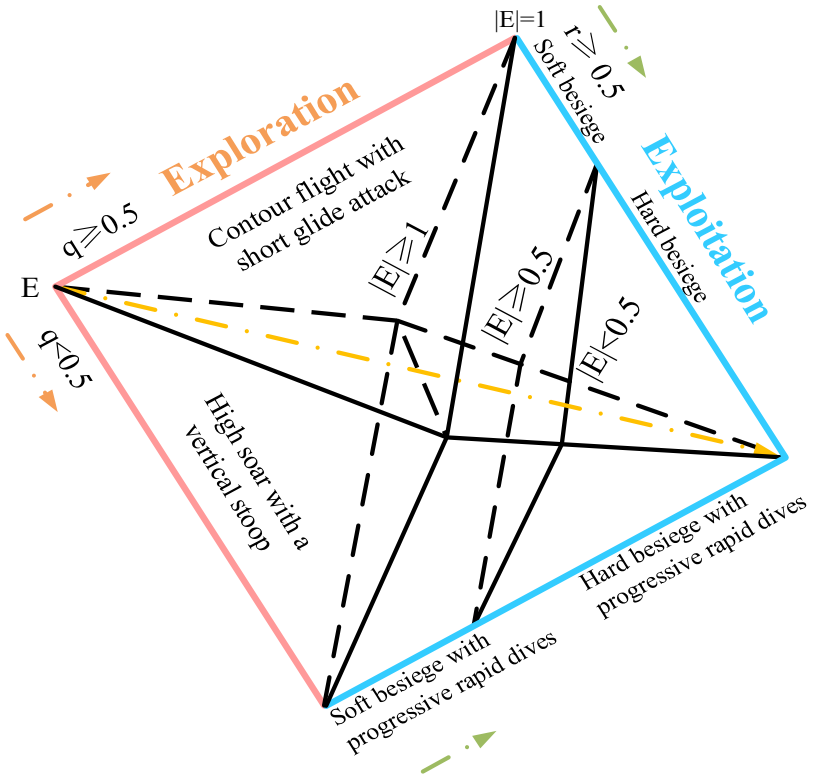


Figure 4. Different phases of IHAOHHO.

In this work, we retain the exploration phase of AO and the exploitation phase of HHO, and combine them together through the transition mechanism. The exploration phase of HHO is highly dominant with randomization that seems clueless search mechanism. In contrast, the position updating in exploration phase of AO is based on the best solution and average position with some randomness, which is more reasonable. And the four exploitation strategies based on the different values of E and r help the algorithm fully exploit the search space. This hybridization is beneficial to give full play to the advantages of these two basic algorithms. The global search capability, faster convergence speed, and detailed exploitation of the algorithms are all reserved. However, the diversity of the population in the exploration phase is insufficient due to the lack of randomness. As described in Section 2.3, RH

is designed for improving the exploration and diversification of an optimization algorithm. Selections from different sub-population can efficiently improve the diversity and exploration capability. Thus, RH strategy is utilized to further improve the diversification of the population in exploration phase, which is conducive to find the most promising region quickly. Besides, AO and HHO possess a common defect of local optima stagnation. The OBL strategy can utilize the opposite solution to make the population jump out of the local optima. Therefore, OBL strategy is added to the exploitation phase to enhance the ability to jump out of the local optima as well. All these strategies effectively improve the convergence speed, convergence accuracy and the overall optimization performance of the hybrid algorithm. This improved hybrid Aquila Optimizer and Harris Hawks Optimization algorithm is named IHAOHHO. Different phases of IHAOHHO are illustrated in Figure 4. The pseudo-code of IHAOHHO is given in Algorithm 1, and the summarized flowchart is shown in Figure 5.

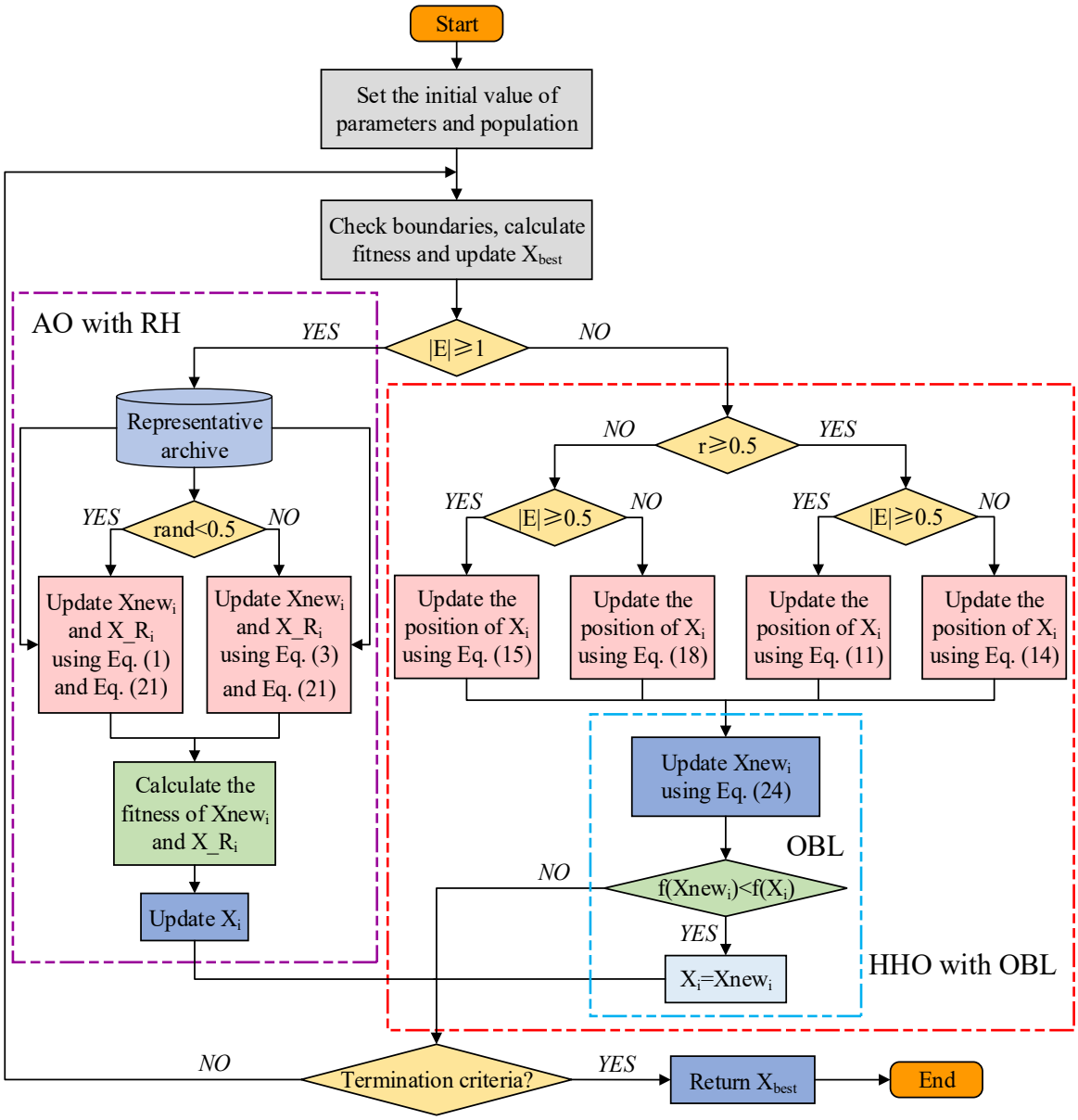


Figure 5. IHAOHHO algorithm flowchart.

Algorithm 1 Pseudo-code of IHAOHHO

Set the initial values of the population size N and the maximum number of iterations T
Initialize positions of the population X
While t < T
  Update x, y, cd
  For i = 1 to N
    If t < T/2 % Exploration part of AO
    Archiving
      If rand < 0.5
        Update the position of $Xnew_i$ and X_R_i using Eqs (1) and (21), respectively
        Calculate the fitness of $Xnew_i$ and X_R_i
        Update the position of X_i
      Else
        Update the position of $Xnew_i$ and X_R_i using Eqs (3) and (21), respectively
        Calculate the fitness of $Xnew_i$ and X_R_i
        Update the position of X_i
      End if
    Else % Exploitation part of HHO
      If r ≥ 0.5 and |E| ≥ 0.5
        Update the position of X_i using Eq (11)
      End if
      If r ≥ 0.5 and |E| < 0.5
        Update the position of X_i using Eq (14)
      End if
      If r < 0.5 and |E| ≥ 0.5
        Update the position of X_i using Eq (15)
      End if
      If r < 0.5 and |E| < 0.5
        Update the position of X_i using Eq (18)
      End if
      Update the position of $Xnew_i$ using Eq (24) % Opposition-based learning (OBL)
      If $f(Xnew_i) < f(X_i)$
        $X_i = Xnew_i$
      End if
    End if
    t = t + 1
  End for
  For i = 1 to N
    Check if the position goes out of the search space boundary and bring it back.
    Calculate the fitness of X_i
    Update X_{best}
  End for
End while
Return X_{best}

Table 1. Unimodal benchmark functions.

Function	Dim	Range	f_{min}
$F_1(x)=\sum_{i=1}^{n}x_i^2$	30	[-100, 100]	0
$F_2(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	30	[-10, 10]	0
$F_3(x)=\sum_{i=1}^{n}(\sum_{j-1}^{i}x_j)^2$	30	[-100, 100]	0
$F_4(x)=\max_i\{\lvert x_i\rvert,1\le i\le n\}$	30	[-100, 100]	0
$F_5(x)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	30	[-30, 30]	0
$F_6(x)=\sum_{i=1}^{n}(x_i+5)^2$	30	[-100, 100]	0
$F_7(x)=\sum_{i=1}^{n}ix_i^4+random[0,1)$	30	[-1.28,1.28]	0

Table 2. Multimodal benchmark functions.

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

3.2. Computational complexity analysis

The computational complexity is an important indicator for an algorithm, which is used to evaluate its time consumption during operating. The computational complexity of the IHAOHHO depends on three rules: initialization, evaluation of fitness, and updating of hawks. In the initialization stage, the computational complexity of positions generated of N hawks is O(N). Then, the computational complexity of fitness evaluation for the best solution is O(N) during the iteration

process. Finally, the computational complexities of position updating of hawks and fitness comparison in one iteration are O(2 × N × D) and O(N) respectively, where D is dimension size of the problem. Therefore, the total computational complexity of the proposed IHAOHHO algorithm is O(N × (1 + 2 × D × T + 2 × T)). As described in the literature, the computational complexity of AO and HHO are both O(N × (1 + D × T + T)). Compared to the basic AO and HHO, the computational complexity of IHAOHHO is slightly increased due to the RH and OBL strategies, which is acceptable for improving the convergence accuracy and speed of the algorithm.

4. Experimental results and discussion

In this section, we implement four main experiments to evaluate the performance of the proposed IHAOHHO algorithm. Standard benchmark function experiment is carried out firstly, which is used to evaluate the performance of the algorithm in solving 23 simple numerical optimization problems. Secondly, the CEC2017 benchmark functions are tested to assess the performance of the algorithm in solving complex numerical optimization problems. Then, sensitivity analysis is performed to investigate the effect of the control parameters. The last one is engineering design problems, which aims to assess the performance of IHAOHHO in solving real-world problems. All experiments are implemented in MATLAB R2016a on a PC with Intel (R) core (TM) i5-9500 CPU @ 3.00GHz and RAM 16GB memory on OS windows 10.

Table 3. Fixed-dimension multimodal benchmark functions.

Function	Dim	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, 65]	1
$F_{15}(x)=\sum_{i=1}^{11}[a_i-\frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}]^2$	4	[-5, 5]	0.00030
$F_{16}(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+x_2^4$	2	[-5, 5]	-1.0316
$F_{17}(x)=(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6)^2+10(1-\frac{1}{8\pi})\cos x_1+10$	2	[-5, 5]	0.398
$F_{18}(x)=[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)]$ $\times[30+(2x_1-3x_2)^2\times(18-32x_2+12x_1^2+48x_2-36x_1x_2+27x_2^2)]$	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	[-1, 2]	-3.86
$F_{20}(x)=-\sum_{i=1}^{4}c_i\exp(-\sum_{j=1}^{6}a_{ij}(x_j-p_{ij})^2)$	6	[0, 1]	-3.32
$F_{21}(x)=-\sum_{i=1}^{5}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0, 10]	-10.1532
$F_{22}(x)=-\sum_{i=1}^{7}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0, 10]	-10.4028
$F_{23}(x)=-\sum_{i=1}^{10}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0, 10]	-10.5363

4.1. Standard benchmark function experiments

We utilize 23 standard benchmark functions to test the performance of the IHAOHHO algorithm, which are divided into three types including unimodal, multimodal and fixed-dimension multimodal benchmark functions. The main characteristic of unimodal functions is that there is only one global optimum but no local optima. This kind of functions can be used to evaluate the exploitation capability and convergence rate of an algorithm. Unlike unimodal functions, multimodal and fixed-dimension multimodal functions have one global optimum and multiple local optima. These types of functions are utilized to evaluate the exploration and local optima avoidance capabilities. The benchmark function details are listed in Tables 1–3.

Table 4. Parameter settings for the comparative algorithms.

Algorithm	Parameters
IHAOHHO	$\sigma_{initial} = 1$; $\sigma_{final} = 0$; Exponent = 2
AO	U = 0.00565; $r_1 = 10$; $\omega = 0.005$; $\alpha = 0.1$; $\delta = 0.1$; $G_1 \in [-1, 1]$; $G_2 = [2, 0]$
HHO	$q \in [0, 1]$; $r \in [0, 1]$; $E_0 \in [-1, 1]$; $E_1 = [2, 0]$; $E \in$ [-2, 2];
HOA	$h_\beta = 0.9$; $h_\gamma = 0.5$; $s_\beta = 0.2$; $s_\gamma = 0.1$; $i_\gamma = 0.3$; $d_\alpha = 0.5$; $d_\beta = 0.2$; $d_\gamma = 0.1$; $r_\delta = 0.1$; $r_\gamma = 0.05$
SSA	$c_1 = [1, 0]$; $c_2 \in [0, 1]$; $c_3 \in [0, 1]$
WOA	$a_1 = [2, 0]$; $a_2 = [-1, -2]$; b = 1
GWO	a = [2, 0]
MVO	WEP $\in$ [0.2,1]; TDR $\in$ [0,1]; r1, r2, r3 $\in$ [0,1]

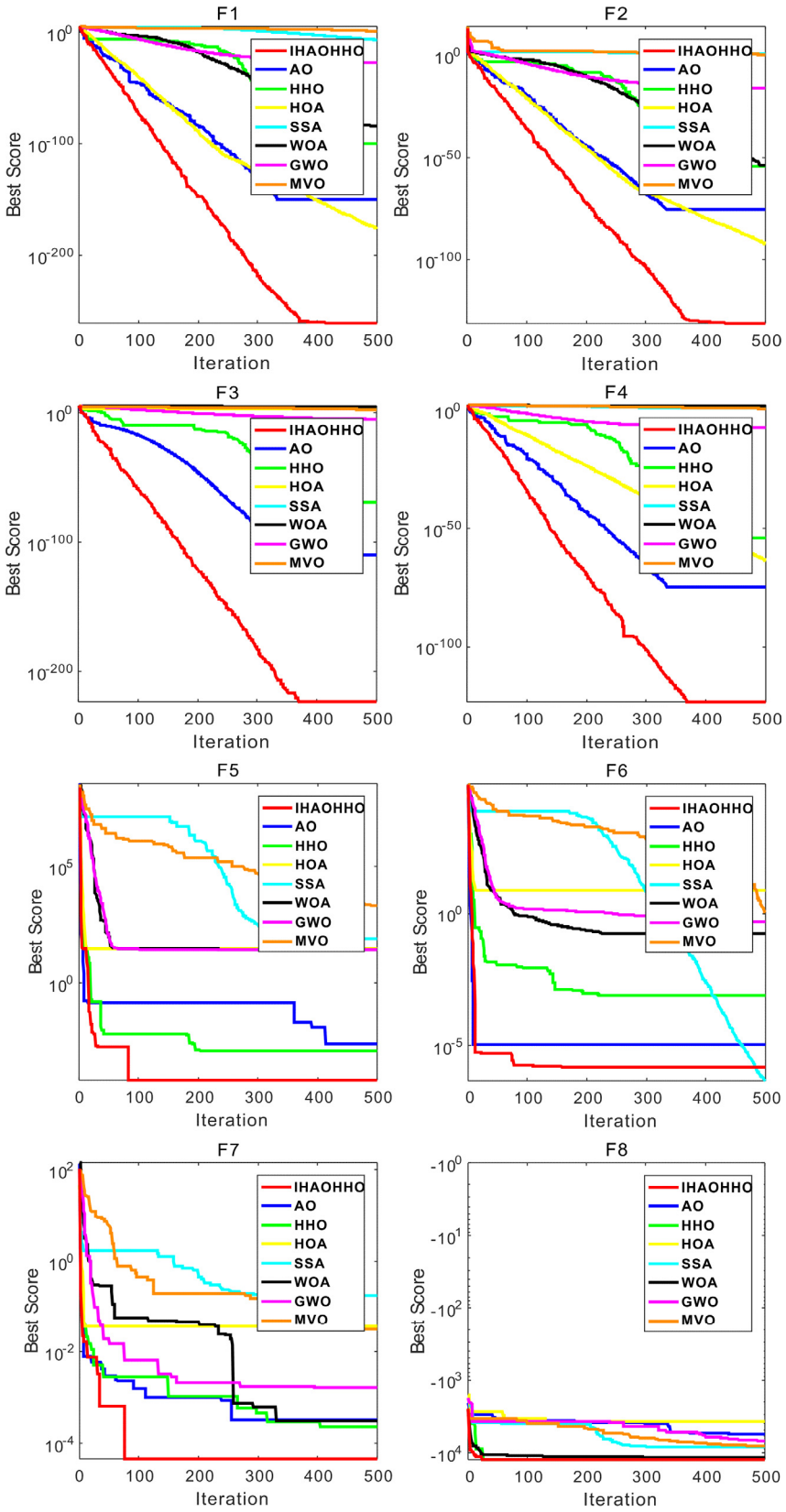
For verification of the results, IHAOHHO is compared with the basic AO, HHO, and HOA, SSA, WOA, GWO, MVO as several well-known meta-heuristic algorithms. For all tests, we set the population size N = 30, dimension size D = 30, maximum number of iterations T = 500, and run 30 times independently. The parameter settings of each algorithm are shown in Table 4. After all, the average and standard deviation results of these test functions are exhibited in Table 5. Figure 6 shows the convergence curves of 23 test functions. The partial search history, trajectory and average fitness maps are shown in Figure 7. The Wilcoxon signed-rank test results are also listed in Table 6.

4.1.1. Results of the algorithms on unimodal test functions (F_1–F_7)

Unimodal test functions are usually used to investigate the exploitation capability of the algorithm since they have only one global optimum and no local optima. As seen from Table 5, the IHAOHHO performs much better than other selected algorithms exclude F_6. For all unimodal functions exclude F_6, IHAOHHO obtains the smallest average values and standard deviations compared to other algorithms, which indicate the best accuracy and stability among all these algorithms. Hence, the exploitation capability of the proposed IHAOHHO algorithm is competitive with all the selected meta-heuristic algorithms.

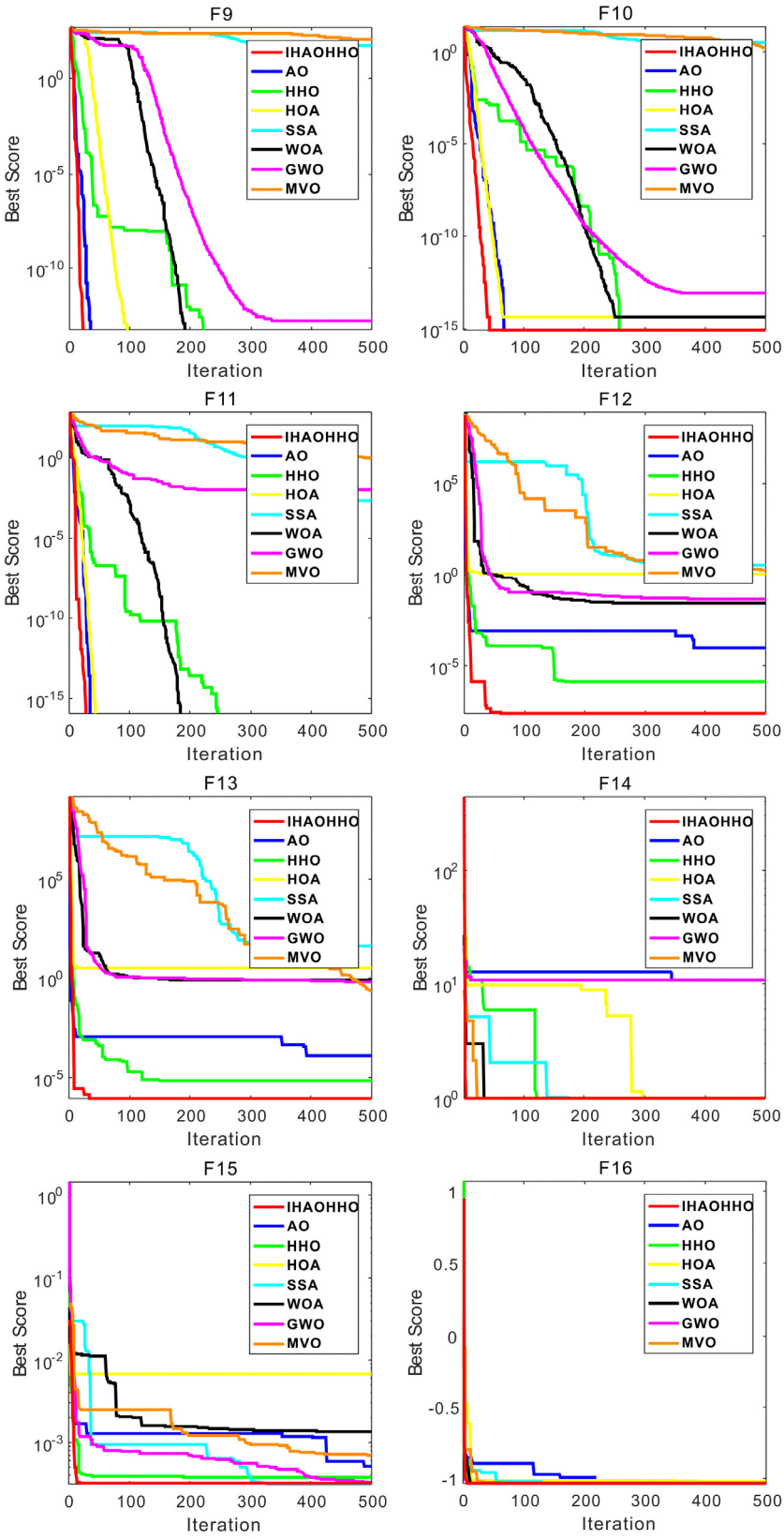
Table 5. Results of algorithms on 23 benchmark functions.

F		IHAOHHO	AO	HHO	HOA	SSA	WOA	MVO	GWO
F1	Avg	**3.3660E-253**	7.9345E-97	3.3401E-96	1.7886E-130	2.1594E-07	1.8120E-73	1.2681	1.0585E-27
	Std	**0**	4.3457E-96	1.7664E-95	8.6475E-130	3.8328E-07	8.0771E-73	3.8945E-01	9.7097E-28
F2	Avg	**1.5599E-127**	1.7619E-60	1.9990E-49	2.1595E-66	1.7056	8.8551E-51	3.7930	7.8893E-17
	Std	**8.5347E-127**	9.6504E-60	9.5039E-49	1.1828E-65	1.3736	3.2476E-50	1.6432E+01	5.7891E-17
F3	Avg	**2.7431E-199**	2.2314E-104	8.4798E-72	1.7856E+02	1.3173E+03	4.4381E+04	2.2448E+02	2.6928E-05
	Std	**0**	1.2222E-103	4.4387E-71	5.2479E+02	1.1198E+03	1.1940E+04	1.2075E+03	6.0686E-05
F4	Avg	**2.2207E-129**	1.6827E-69	1.1057E-49	4.5440E-65	1.2612E+01	5.6993E+01	2.1139	8.0152E-07
	Std	**1.1068E-128**	7.6613E-69	3.0441E-49	2.4668E-64	3.8228	2.5529E+01	1.0284	8.3055E-07
F5	Avg	**5.3852E-04**	5.4421E-03	2.1438E-02	2.8964E+01	2.2071E+02	2.7871E+01	3.8794E+02	2.7221E+01
	Std	**2.2657E-03**	1.2008E-02	2.4291E-02	3.5795E-02	3.4556E+02	4.6073E-01	7.0618E-02	7.9162E-01
F6	Avg	3.5863E-06	1.2028E-04	1.4716E-04	6.5645	**2.5056E-07**	3.4988E-01	1.1414	7.2412E-01
	Std	7.7280E-06	1.7490E-04	2.0847E-04	3.4100E-01	**3.8211E-07**	1.7102E-01	4.4812E-01	3.9184E-01
F7	Avg	**9.5324E-05**	1.0504E-04	1.3211E-04	6.7657E-02	1.6706E-01	2.7105E-03	3.4554E-02	2.1436E-03
	Std	**7.6717E-05**	8.6163E-05	1.4567E-04	4.0688E-02	7.8018E-02	3.2053E-03	1.5231E-02	1.3151E-03
F8	Avg	**-12569.3858**	-8010.0774	-12569.0257	-4106.2204	-7362.4999	-10632.1912	-7648.8103	-6101.7628
	Std	**1.8228E-01**	4.0684E+03	7.5912E-01	7.7530E+02	8.6690E+02	**0**	5.2123E+02	8.9531E+02
F9	Avg	**0**	**0**	**0**	8.0284E+01	5.3197E+01	**0**	1.2456E+02	3.3205
	Std	**0**	**0**	**0**	1.0686E+02	2.0211E+01	4.6777E-15	3.4040E+01	4.302
F10	Avg	**8.8818E-16**	**8.8818E-16**	**8.8818E-16**	5.7436E-15	2.6880	2.7886E-15	1.7605	9.8230E-14
	Std	**0**	**0**	**0**	1.7413E-15	6.5906E-01	2.421E-15	6.6839E-01	1.7092E-14
F11	Avg	**0**	**0**	**0**	2.0978E-01	2.3006E-02	1.1525E-02	8.6051E-01	5.3578E-03
	Std	**0**	**0**	**0**	3.7796E-01	1.5437E-02	4.4029E-02	8.3088E-02	7.3633E-03
F12	Avg	**2.6974E-07**	2.8068E-06	1.9506E-05	1.1656	7.8473	2.1363E-02	1.8746	5.7227E-02
	Std	**4.4163E-07**	5.7142E-06	3.8352E-05	2.1289E-01	2.7926	1.2107E-02	1.1276	9.6891E-02
F13	Avg	**3.0227E-06**	4.9439E-05	9.7772E-05	3.0571	1.8860E+01	5.3918E-01	1.7158E-01	6.2217E-01
	Std	**5.0785E-06**	9.2988E-05	9.5928E-05	1.8645E-01	1.6484E+01	2.9800E-01	1.0772E-01	2.5487E-01
F14	Avg	1.5932	2.0487	1.2629	2.9322	1.3943	2.8615	**0.998**	4.6223
	Std	9.2477E-01	2.1859	5.1727E-01	2.1542	9.5834E-01	3.3284	**2.7158E-11**	4.1450
F15	Avg	4.4227E-04	4.9359E-04	**3.9582E-04**	7.1269E-03	2.2249E-03	7.5110E-04	7.8593E-03	5.7748E-03
	Std	3.4587E-04	**1.1786E-04**	2.3573E-04	7.5345E-03	4.9361E-03	4.9981E-04	1.3437E-02	8.9518E-03
F16	Avg	**-1.0316**	-1.0312	**-1.0316**	-0.99752	**-1.0316**	**-1.0316**	**-1.0316**	**-1.0316**
	Std	3.1088E-08	4.0636E-04	7.1130E-09	3.1957E-02	**1.8547E-14**	7.6118E-10	4.5176E-07	3.6867E-08
F17	Avg	**3.9789E-01**	3.9813E-01	**3.9789E-01**	3.9790E-01	**3.9789E-01**	3.9790E-01	**3.9789E-01**	**3.9789E-01**
	Std	4.0428E-05	3.0290E-04	4.5913E-05	1.1204E-03	**9.2471E-15**	1.5755E-05	7.7424E-08	7.6565E-07
F18	Avg	**3**	3.0243	**3**	7.7479	**3**	3.0001	**3**	**3**
	Std	3.3248E-06	5.5712E-02	6.1620E-07	1.6153E+01	**1.9031E-13**	3.1781E-04	2.5080E-06	4.7166E-05
F19	Avg	-3.8258	-3.8535	-3.8597	-3.8619	**-3.8628**	-3.8551	-3.8628	-3.8598
	Std	7.1972E-02	8.7650E-03	3.8900E-03	6.2715E-04	**1.9037E-12**	8.6573E-03	3.7043E-06	3.9554E-03
F20	Avg	-3.0792	-3.1678	-3.0813	-3.2580	-3.2335	-3.2080	-3.2448	**-3.2721**
	Std	1.1913E-01	7.3334E-02	9.9164E-02	7.5769E-02	6.4415E-02	1.1882E-01	**5.9771E-02**	7.6156E-02
F21	Avg	**-10.1525**	-10.1422	-5.2145	-9.3937	-6.6406	-9.2821	-6.9627	-9.6452
	Std	**1.4463E-03**	1.8290E-02	8.8739E-01	1.2967	3.4554	1.9234	3.1618	1.5462
F22	Avg	**-10.4026**	-10.3914	-5.0820	-9.5706	-9.7954	-7.6313	-8.6587	-10.4009
	Std	**6.6551E-04**	2.1551E-02	8.7380E-03	1.5379	1.8875	3.0683	2.7631	1.2358E-03
F23	Avg	**-10.5359**	-10.5292	-5.1234	-9.8166	-8.7191	-7.0303	-7.5296	-10.5348
	Std	**6.6691E-04**	1.2664E-02	52035E-03	1.0080	3.1206	3.4442	3.5941	8.4873E-04



Continued on next page

Figure 6. Convergence curves of 23 benchmark functions.



Continued on next page

Figure 6. Convergence curves of 23 benchmark functions.

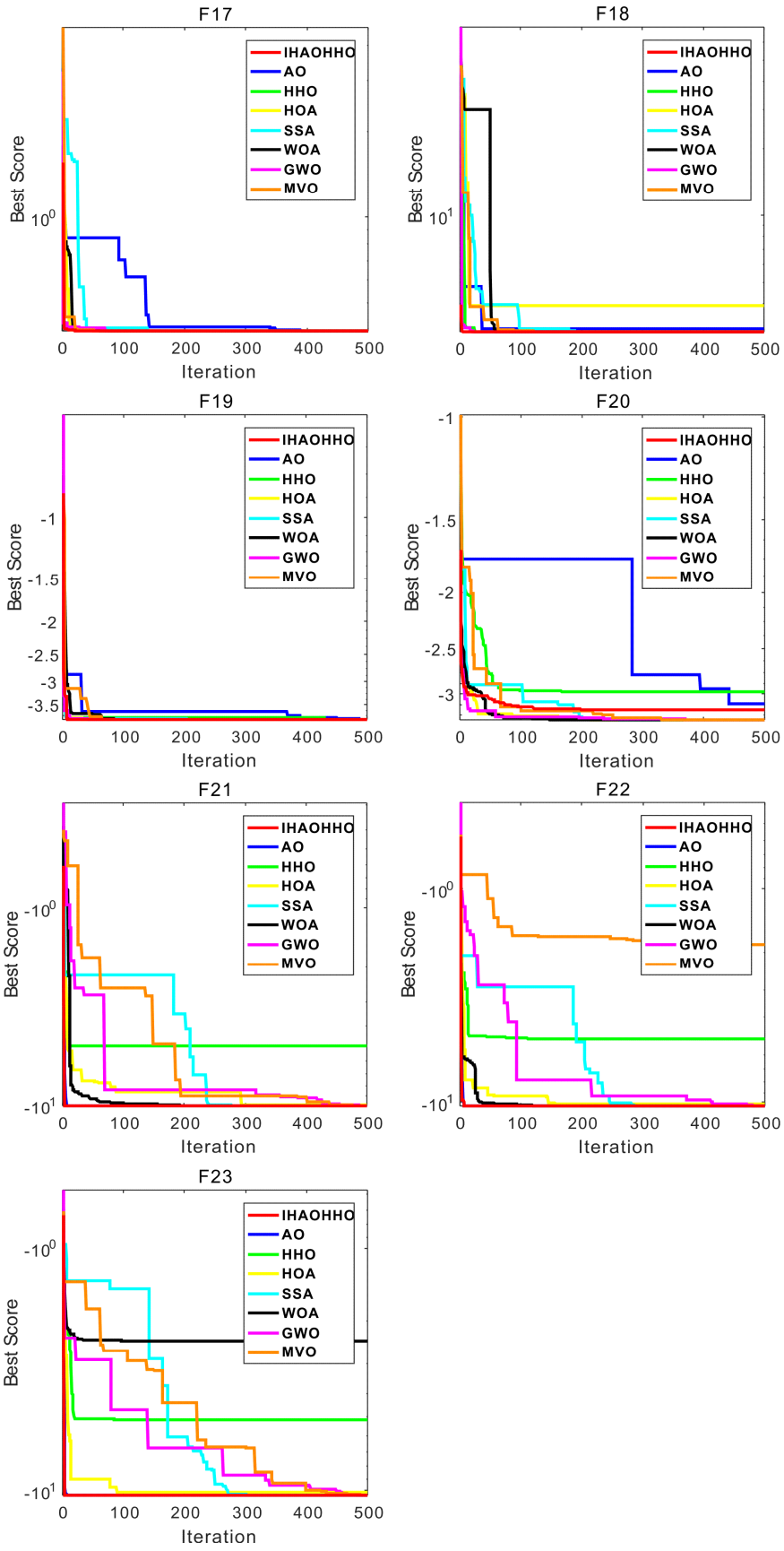


Figure 6. Convergence curves of 23 benchmark functions.

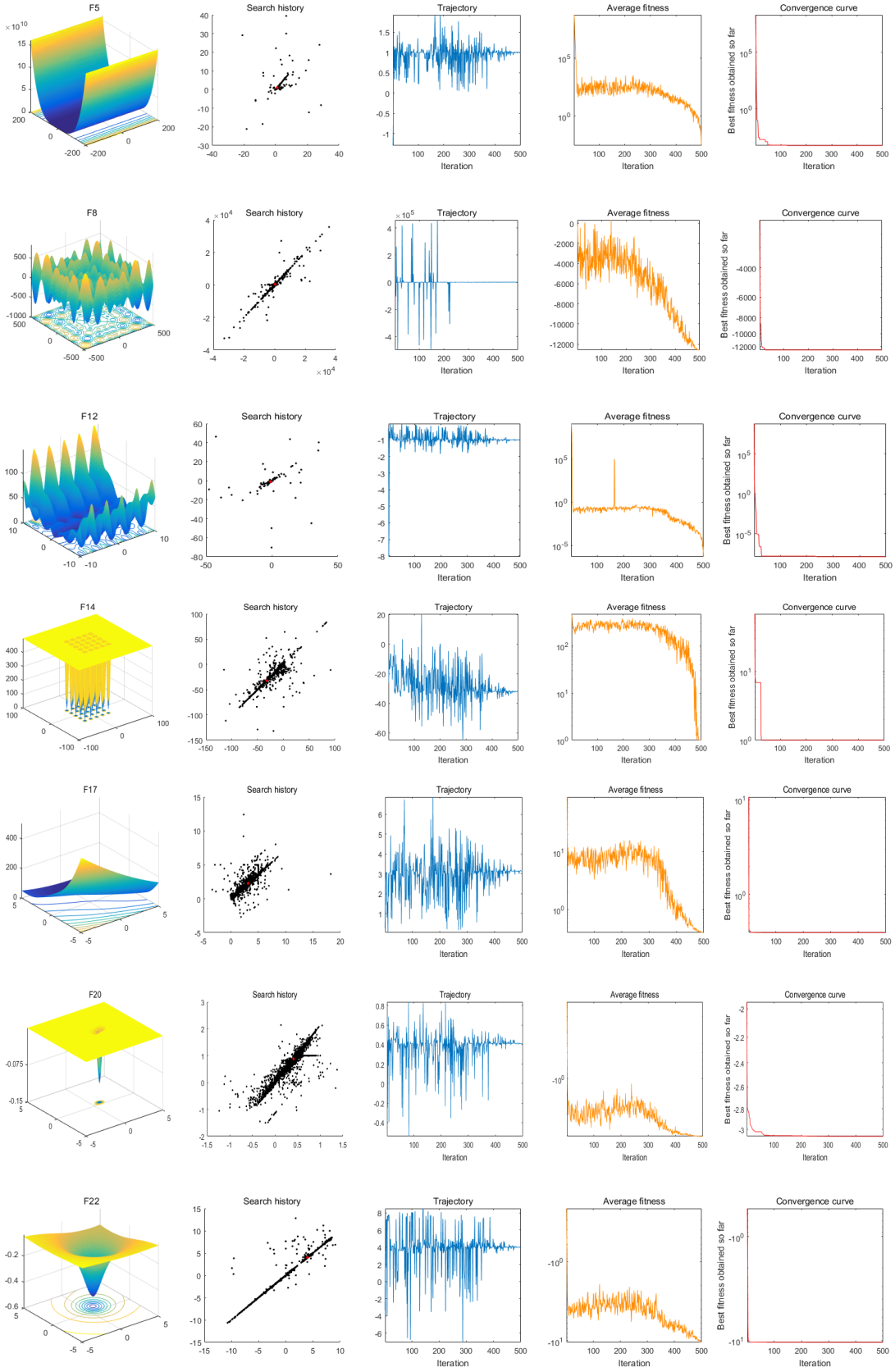


Figure 7. Parameter space, search history, trajectory, average fitness, and convergence curve of IHAOHHO.

4.1.2. Results of the algorithms on multimodal test functions (F_8–F_{23})

Multimodal test functions F_8–F_{23} contain plentiful local optima whose number increases exponentially with the dimension size of the problem. These functions are very useful to evaluate the exploration ability and local optima avoidance of the algorithm. It can be seen from Table 5 that IHAOHHO outperforms other algorithms in most of the multimodal and fixed-dimension multimodal functions. For multimodal functions F_8–F_{13}, IHAOHHO shows completely superiority than other selected algorithms with the best average values and standard deviations. For ten fixed-dimensions multimodal functions F_{14}–F_{23}, IHAOHHO performs barely satisfactory. The IHAOHHO outperforms others in terms of both average values and standard deviations in F_{21}–F_{23}, and achieves the best accuracy for F_{16}–F_{18}. These results reveal that IHAOHHO can also provide superior exploration capability.

4.1.3. Analysis of convergence behavior

Search agents tend to change drastically to investigate promising regions of the search space in early iterations, and then exploit the region in detail and converge gradually as the number of iterations increases. Convergence curves of the IHAOHHO, AO, HHO, HOA, SSA, WOA, GWO, and MVO for 23 standard benchmark functions are given in Figure 6, which show the convergence rate of algorithms. As we can see, IHAOHHO shows competitive performance compared to other state-of-the-art algorithms. The IHAOHHO algorithm presents faster convergence speed than all other algorithms in F_1–F_4 and F_8–F_{11}. For else test functions, IHAOHHO may not have much advantages than other algorithms in terms of convergence speed with the reason that some algorithms are excellent as well, but the convergence accuracy of IHAOHHO is better than other algorithms in most of the test functions.

The superiority of IHAOHHO in terms of convergence speed is likely to come from the RH strategy in exploration phase. To be specific, the RH strategy provides better randomness and diversity for the search agents, making search agents explore the search space widely and randomly. The improvement of randomness and diversity increases the probability of finding the most promising region quickly. The advantage of convergence accuracy is likely to be derived from the OBL strategy, which improves randomness of search agents. The search agents can choose the better one to jump out of the local optima in each iteration. These two strategies help the hybrid algorithm outperforms the basic AO and HHO. Overall, IHAOHHO can efficiently achieve great solutions for all 23 standard benchmark functions.

4.1.4. Qualitative results and analysis

Furthermore, Figure 7 shows us the results of several representative test functions on search history, trajectory, average fitness and convergence curve. From search history maps, we can see the search agents' distribution of the IHAOHHO while exploring and exploiting the search space. Because of the fast convergence, the vast majority of search agents are concentrated near the global optimum. Inspecting trajectory figures in Figure 7, the first search agent constantly oscillates in the first dimension of the search space, which suggests that the search agent investigates the most promising areas and better solutions widely. This powerful search capability is likely to come from the RH and OBL strategies. The average fitness presents if exploration and exploitation are conducive to improve the first random population and an accurate approximation of the global optimum can be found in the

end. Similarly, it can be noticed that the average fitness oscillates like trajectories in the early iterations, and then decreases abruptly and levels off accordingly. The average fitness maps show the great improvement of the first random population and the acquisition of the final global optimal approximation as well. At last, the convergence curves reveal the best fitness values found by search agents after each of iteration. By observing this, the IHAOHHO shows very fast convergence speed.

Table 6. P-values from the Wilcoxon signed-rank test for the results in Table 5.

F	IHAOHHO vs AO	IHAOHHO vs HHO	IHAOHHO vs HOA	IHAOHHO vs SSA	IHAOHHO vs WOA	IHAOHHO vs MVO	IHAOHHO vs GWO
F1	6.1035E-05	6.1035E-05	6.1035E-05	6.1035E-05	6.1035E-05	6.1035E-05	6.1035E-05
F2	6.1035E-05	6.1035E-05	6.1035E-05	6.1035E-05	6.1035E-05	6.1035E-05	6.1035E-05
F3	6.1035E-05	6.1035E-05	6.1035E-05	6.1035E-05	6.1035E-05	6.1035E-05	6.1035E-05
F4	6.1035E-05	6.1035E-05	6.1035E-05	6.1035E-05	6.1035E-05	6.1035E-05	6.1035E-05
F5	**0.10699**	1.2207E-04	6.1035E-05	6.1035E-05	6.1035E-05	6.1035E-05	6.1035E-05
F6	1.8311E-04	1.8311E-04	6.1035E-05	1.1597E-03	6.1035E-05	6.1035E-05	6.1035E-05
F7	**6.3721E-02**	**0.10699**	6.1035E-05	6.1035E-05	6.1035E-05	6.1035E-05	6.1035E-05
F8	6.1035E-05	**0.63867**	6.1035E-05	6.1035E-05	6.1035E-05	6.1035E-05	6.1035E-05
F9	NaN	NaN	6.1035E-05	6.1035E-05	NaN	6.1035E-05	6.1035E-05
F10	NaN	NaN	6.1035E-05	6.1035E-05	1.9531E-03	6.1035E-05	6.1035E-05
F11	NaN	NaN	**0.1250**	6.1035E-05	NaN	6.1035E-05	NaN
F12	2.6245E-03	1.2207E-04	6.1035E-05	6.1035E-05	6.1035E-05	6.1035E-05	6.1035E-05
F13	2.1545E-02	1.2207E-04	6.1035E-05	6.1035E-05	6.1035E-05	6.1035E-05	6.1035E-05
F14	**0.97797**	**0.45428**	1.8066E-02	2.6245E-03	**0.18762**	6.1035E-05	3.5339E-02
F15	6.1035E-05	**0.45428**	6.1035E-05	6.1035E-05	6.1035E-05	6.1035E-05	**0.3028**
F16	6.1035E-05	8.3618E-03	6.1035E-05	6.1035E-05	**9.4604E-02**	6.1035E-05	**5.5359E-02**
F17	6.1035E-04	**0.45428**	6.1035E-05	6.1035E-05	**0.27686**	3.0518E-04	1.0254E-02
F18	**8.3252E-02**	1.2207E-04	**8.3252E-02**	6.1035E-05	**0.48871**	**0.67877**	**0.13538**
F19	**7.2998E-02**	1.8066E-02	**7.2998E-02**	6.1035E-05	1.8066E-02	6.1035E-05	3.0151E-02
F20	6.1035E-04	**0.93408**	8.5449E-04	1.5259E-03	2.0142E-03	6.1035E-05	8.5449E-04
F21	3.3569E-03	6.1035E-05	6.1035E-05	4.2120E-02	1.2207E-04	2.0142E-03	1.8311E-04
F22	1.2207E-04	6.1035E-05	6.1035E-05	**8.3252E-02**	1.8311E-04	4.126E-02	8.3618E-03
F23	1.1597E-03	6.1035E-05	6.1035E-05	3.3026E-02	6.1035E-05	1.6882E-02	6.7139E-03

4.1.5. The Wilcoxon test

The Wilcoxon signed-rank test is a non-parametric statistical test and useful to evaluate the statistical performance differences between the proposed IHAOHHO algorithm and other algorithms. As is well-known, p-values less than 0.05 indicate that there is a significant difference between the two compared algorithms. The calculated results of Wilcoxon signed-rank test between IHAOHHO and other seven algorithms for each benchmark functions are listed in Table 6. According to the criterion of 0.05, IHAOHHO outperforms all other algorithms in varying degrees. This superiority is statistically significant on unimodal functions F_1–F_6, which strongly indicates that IHAOHHO possesses high exploitation. IHAOHHO also shows better results on multimodal function F_8–F_{23}, which may suggest that IHAOHHO has a high capability of exploration. To sum up, the IHAOHHO

algorithm can provide better results on almost all benchmark functions than other comparative algorithms.

Table 7. Descriptions of the benchmark functions from CEC2017.

Function	Name	Dim	Range	f_{min}
Unimodal functions				
F24	Shifted and Rotated Bent Cigar Function	10	[-100, 100]	100
F25	Shifted and Rotated Zakharov Function	10	[-100, 100]	300
Multimodal functions				
F26	Shifted and Rotated Rosenbrock's Function	10	[-100, 100]	400
F27	Shifted and Rotated Rastrigin's Function	10	[-100, 100]	500
F28	Shifted and Rotated Expanded Scaffer's F6 Function	10	[-100, 100]	600
F29	Shifted and Rotated Lunacek Bi-RastriginFunction	10	[-100, 100]	700
F30	Shifted and Rotated Non-Continuous Rastrigin's Function	10	[-100, 100]	800
F31	Shifted and Rotated Levy Function	10	[-100, 100]	900
F32	Shifted and Rotated Schwefel's Function	10	[-100, 100]	1000
Hybrid functions (N is basic number of functions)				
F33	Hybrid Function 1 (N = 3)	10	[-100, 100]	1100
F34	Hybrid Function 2 (N = 3)	10	[-100, 100]	1200
F35	Hybrid Function 3 (N = 3)	10	[-100, 100]	1300
F36	Hybrid Function 4 (N = 4)	10	[-100, 100]	1400
F37	Hybrid Function 5 (N = 4)	10	[-100, 100]	1500
F38	Hybrid Function 6 (N = 4)	10	[-100, 100]	1600
F39	Hybrid Function 6 (N = 5)	10	[-100, 100]	1700
F40	Hybrid Function 6 (N = 5)	10	[-100, 100]	1800
F41	Hybrid Function 6 (N = 5)	10	[-100, 100]	1900
F42	Hybrid Function 6 (N = 6)	10	[-100, 100]	2000
Composite functions (N is basic number of functions)				
F43	Composite Function 1 (N = 3)	10	[-100, 100]	2100
F44	Composite Function 2 (N = 3)	10	[-100, 100]	2200
F45	Composite Function 3 (N = 4)	10	[-100, 100]	2300
F46	Composite Function 4 (N = 4)	10	[-100, 100]	2400
F47	Composite Function 5 (N = 5)	10	[-100, 100]	2500
F48	Composite Function 6 (N = 5)	10	[-100, 100]	2600
F49	Composite Function 7 (N = 6)	10	[-100, 100]	2700
F50	Composite Function 8 (N = 6)	10	[-100, 100]	2800
F51	Composite Function 9 (N = 6)	10	[-100, 100]	2900
F52	Composite Function 10 (N = 3)	10	[-100, 100]	3000

4.2. Experiments on the CEC2017 benchmark function

Standard benchmark function experiments demonstrate the superior performance on simple problems of the proposed IHAOHHO algorithm. The complex functions can help to investigate the performance on intricate problems. One of the most challenging test function suites called CEC2017 [54] is selected to further test the performance of IHAOHHO, which contains 30 functions with more than

half of the challenging hybrid and composition functions as shown in Table 7. The test results are compared to some well-known and latest algorithms proposed recently, in which IPOP-CMA-ES and LSHADE are the best behaved on CEC2017 in the literature. As the previous section described, each algorithm is ran 30 times with 500 iterations, and average and standard deviation results are presented in Table 8. From the comparison results, the proposed IHAOHHO obtains the 3rd rank following IPOP-CMA-ES and LSHADE, and exceeds SSC, RUN and HOA methods completely. It reveals that IHAOHHO can also achieve better results on complex functions.

Table 8. Comparison results of algorithms on CEC2017.

F		IHAOHHO	IPOP-CMAES	LSHADE	SSC	RUN	HOA
F24	Avg	1.77467E + 09	1.00000E + 02	1.00000E + 02	2.50699E + 09	3.75483E + 04	3.50163E + 08
	Std	1.11254E + 09	0.00000E + 00	0.00000E + 00	1.72594E + 09	1.40126E + 04	1.57335E + 08
F25	Avg	4.69684E + 03	3.00000E + 02	3.00000E + 02	4.84926E + 03	5.05458E + 04	6.03840E + 03
	Std	1.99656E + 03	0.00000E + 00	0.00000E + 00	2.04210E + 03	8.29687E + 03	2.15957E + 03
F26	Avg	4.15353E + 02	4.00091E + 02	4.00000E + 02	6.35108E + 02	5.13264E + 02	4.53729E + 02
	Std	1.69972E + 01	4.43520E-02	0.00000E + 00	1.98490E + 02	1.81438E + 01	3.02199E + 01
F27	Avg	5.57466E + 02	5.19197E + 02	5.02340E + 02	5.60349E + 02	6.53772E + 02	5.74943E + 02
	Std	1.01328E + 01	8.41520E + 00	8.75000E-01	1.84807E + 01	2.91163E + 01	1.10416E + 01
F28	Avg	6.29286E + 02	6.00000E + 02	6.00000E + 02	6.36104E + 02	6.40548E + 02	6.37022E + 02
	Std	7.67590E + 00	0.00000E + 00	2.59000E-07	1.01438E + 01	8.25192E + 00	8.80160E + 00
F29	Avg	7.69414E + 02	7.32212E + 02	7.12230E + 02	8.13508E + 02	9.36961E + 02	7.79986E + 02
	Std	1.20032E + 01	3.93460E + 00	6.05000E-01	1.64856E + 01	5.73895E + 01	1.27443E + 01
F30	Avg	8.40446E + 02	8.14912E + 02	8.02140E + 02	8.46992E + 02	9.21845E + 02	8.55157E + 02
	Std	7.39460E + 00	8.47500E + 00	1.03100E + 00	1.04671E + 01	2.62983E + 01	8.87940E + 00
F31	Avg	1.36798E + 03	9.00000E + 02	9.00000E + 02	1.44797E + 03	3.52693E + 03	1.09708E + 03
	Std	2.12360E + 02	0.00000E + 00	0.00000E + 00	2.89598E + 02	8.96934E + 02	1.40757E + 02
F32	Avg	2.20829E + 03	2.29077E + 03	1.07003E + 03	2.89454E + 03	5.14578E + 03	2.87917E + 03
	Std	1.90189E + 02	2.37652E + 02	5.65600E + 01	2.06134E + 02	7.73937E + 02	2.06107E + 02
F33	Avg	1.20861E + 03	1.16709E + 03	1.10002E + 03	1.38357E + 03	1.26564E + 03	1.30871E + 03
	Std	2.31887E + 01	1.35981E + 02	1.20000E-01	1.02038E + 02	3.23837E + 01	9.44024E + 01
F34	Avg	9.65212E + 06	1.61308E + 03	1.33295E + 03	2.29908E + 06	1.38367E + 07	1.75532E + 07
	Std	6.54481E + 06	1.73789E + 02	8.62600E + 01	6.06195E + 06	9.36578E + 06	1.17226E + 07
F35	Avg	1.73810E + 04	1.35805E + 03	1.30374E + 03	4.09334E + 04	2.63421E + 04	1.34045E + 06
	Std	1.00028E + 04	6.27520E + 01	3.26000E + 00	2.07982E + 04	1.45826E + 04	1.11750E + 06
F36	Avg	2.51467E + 03	1.47391E + 03	1.40019E + 03	6.63735E + 03	2.27458E + 05	2.80598E + 03
	Std	1.21526E + 03	4.18650E + 01	4.50000E-01	1.23909E + 03	1.87179E + 05	1.25523E + 03
F37	Avg	8.52054E + 03	1.51384E + 03	1.50033E + 03	1.77014E + 04	1.42723E + 04	3.15987E + 04
	Std	3.16501E + 03	2.10632E + 01	2.00000E-01	7.89396E + 03	3.59942E + 03	3.55718E + 04
F38	Avg	1.90369E + 03	1.66968E + 03	1.60087E + 03	1.92836E + 03	2.84572E + 03	2.00480E + 03
	Std	1.00750E + 02	3.53711E + 01	3.60000E-01	1.12527E + 02	3.28867E + 02	1.58304E + 02
F39	Avg	1.78843E + 03	1.77638E + 03	1.70137E + 03	1.79644E + 03	2.24703E + 03	1.90857E + 03
	Std	3.41834E + 01	1.23878E + 01	3.84000E + 00	1.87221E + 01	2.22813E + 02	8.69727E + 01
F40	Avg	1.71900E + 04	1.90952E + 03	1.80359E + 03	1.27448E + 05	6.11908E + 05	3.62014E + 05
	Std	1.05037E + 04	2.39660E + 01	7.60000E + 00	1.40079E + 05	7.60372E + 05	2.40106E + 05
F41	Avg	1.58292E + 05	1.91811E + 03	1.90026E + 03	2.81911E + 04	4.43675E + 05	1.66843E + 04
	Std	4.02208E + 05	2.12890E + 01	3.00000E-02	6.61395E + 03	3.45206E + 05	1.56893E + 04

Continued on next page

F		IHAOHHO	IPOP-CMAES	LSHADE	SSC	RUN	HOA
F42	Avg	2.17157E + 03	2.05983E + 03	2.00023E + 03	2.28443E + 03	2.56095E + 03	2.26788E + 03
	Std	6.49892E + 01	1.85766E + 01	4.30000E-01	6.52079E + 01	1.70413E + 02	8.48882E + 01
F43	Avg	2.30614E + 03	2.31934E + 03	2.25542E + 03	2.32175E + 03	2.44601E + 03	2.35311E + 03
	Std	2.47564E + 01	8.46660E + 00	5.21600E + 01	6.07100E + 01	2.52926E + 01	4.32033E + 01
F44	Avg	2.41213E + 03	2.73101E + 03	2.30010E + 03	3.59786E + 03	3.31843E + 03	2.35109E + 03
	Std	8.37419E + 01	6.26970E + 01	1.70000E-01	7.07466E + 02	1.86094E + 03	3.37269E + 01
F45	Avg	2.62413E + 03	2.63000E + 03	2.60230E + 03	2.65854E + 03	2.80612E + 03	2.72046E + 03
	Std	2.12841E + 01	5.55150E + 00	1.42000E + 00	2.74840E + 01	2.95349E + 01	2.25245E + 01
F46	Avg	2.78904E + 03	2.70961E + 03	2.68830E + 03	2.80479E + 03	2.98605E + 03	2.82014E + 03
	Std	2.75184E + 01	9.78340E + 00	9.16500E + 01	2.35894E + 01	4.61846E + 01	5.20187E + 01
F47	Avg	2.93523E + 03	2.93212E + 03	2.92377E + 03	3.02323E + 03	2.93603E + 03	2.96740E + 03
	Std	1.96255E + 01	8.33760E + 00	2.13000E + 01	5.86012E + 01	2.67693E + 01	1.82177E + 01
F48	Avg	3.28386E + 03	3.21722E + 03	2.90000E + 03	4.05668E + 03	4.50153E + 03	3.33849E + 03
	Std	2.31698E + 02	2.03330E + 02	0.00000E + 00	2.87720E + 02	1.27120E + 03	4.73594E + 02
F49	Avg	3.11226E + 03	3.08917E + 03	3.08903E + 03	3.11634E + 03	3.31206E + 03	3.23484E + 03
	Std	1.80612E + 01	4.24482E + 01	1.05000E + 00	2.68443E + 01	3.57031E + 01	5.28018E + 01
F50	Avg	3.21692E + 03	3.27673E + 03	3.15435E + 03	3.25502E + 03	3.28169E + 03	3.49994E + 03
	Std	1.07032E + 02	9.66220E + 00	1.10920E + 01	3.19693E + 01	2.06487E + 01	1.36289E + 03
F51	Avg	3.35870E + 03	3.27842E + 03	3.13492E + 03	3.36830E + 03	4.24526E + 03	3.37358E + 03
	Std	6.51044E + 01	5.74741E + 01	3.87000E + 00	9.54409E + 01	2.74106E + 02	7.28448E + 01
F52	Avg	3.29001E + 06	3.28462E + 04	3.41838E + 03	7.61412E + 06	3.99731E + 06	4.12253E + 06
	Std	2.57991E + 06	2.60735E + 04	2.28600E + 01	4.84549E + 06	2.71864E + 06	4.95066E + 06

Table 9. Sensitivity analysis on the IHAOHHO's parameters.

F	$\sigma_{initial} = 1$, E = 1	$\sigma_{initial} = 1$, E = 1.5	$\sigma_{initial} = 1$, E = 2	$\sigma_{initial} = 1.5$, E = 1	$\sigma_{initial} = 1.5$, E = 1.5	$\sigma_{initial} = 1.5$, E = 2	$\sigma_{initial} = 2$, E = 1	$\sigma_{initial} = 2$, E = 1.5	$\sigma_{initial} = 2$, E = 2
F5	2.62E - 06	3.89E - 06	**3.80E - 08**	1.90E - 06	2.44E - 06	5.84E - 08	4.24E - 06	8.99E - 07	3.79E - 06
F7	4.04E - 08	6.20E - 08	**1.47E - 08**	5.79E - 08	1.32E - 07	6.93E - 08	3.18E - 08	1.60E - 08	2.08E - 08
F9	0	0	**0**	0	0	0	0	0	0
F13	1.16E - 10	1.04E - 11	**3.20E - 12**	3.29E - 11	3.81E - 11	2.52E - 11	5.34E - 12	8.14E - 11	7.61E - 11
F14	5.4418	0.43653	**0.10935**	0.54588	0.87306	0.87306	0.54591	1.7461	5.1709
F18	1.18E - 09	3.02E - 09	**7.59E - 11**	8.23E - 11	25.1496	25.1379	1.96E - 08	2.73E - 06	25.1593
F25	6.50E + 07	6.63E + 07	6.55E + 07	5.50E + 07	4.99E + 07	**4.84E + 07**	6.22E + 07	6.56E + 07	7.25E + 07
F26	2.94E + 04	2.14E + 04	1.51E + 04	**1.26E + 04**	1.87E + 04	1.86E + 04	1.47E + 04	2.76E + 04	2.70E + 04
F30	2.43E + 035	**1.73E + 03**	1.78E + 03	2.18E + 03	2.12E + 03	2.10E + 03	1.74E + 03	2.04E + 03	2.13E + 03
F33	6.61E + 04	5.12E + 04	**2.92E + 04**	8.56E + 04	4.04E + 04	6.08E + 04	5.48E + 04	4.45E + 04	6.75E + 04
F39	2.43E + 04	1.94E + 04	1.73E + 04	1.54E + 04	1.68E + 04	1.26E + 04	**1.21E + 04**	1.46E + 04	1.22E + 04
F44	4.31E + 04	5.06E + 04	4.78E + 04	5.88E + 04	**4.08E + 04**	9.79E + 04	4.93E + 04	6.43E + 04	7.79E + 04
F49	1.91E + 05	1.95E + 05	**1.85E + 05**	1.93E + 05	1.87E + 05	2.01E + 05	1.90E + 05	1.95E + 05	1.88E + 05

4.3. Sensitivity analysis

The performance of an optimization algorithm is affected by the values of the control parameters. For the sake of better performance, the influence of the parameters should be investigated to select the appropriate values. The IHAOHHO algorithm owns three parameters $\sigma_{initial}$, σ_{final} and *Exponent* in

Eq (22). At the end of the iteration, the algorithm needs to search in detail and minimize randomness as much as possible. Thus, σ_{final} should be equal to 0 to get rid of the random term in Eq (21). Next, the left two parameters $\sigma_{initial}$ and *Exponent* are assessed by the representative standard and CEC2017 benchmark functions in Table 9. The mean-square error values are obtained using benchmark functions from different categories including unimodal, multimodal and fixed-multimodal of standard benchmark functions, and unimodal, multimodal, hybrid and composite of CEC2017 with different parameters. The best performance bolded is obtained by values 1 and 2 for parameters $\sigma_{initial}$ and *Exponent*.

4.4. Experiments on engineering design problems

Considering equality and inequality constraints is a necessary process for optimization because most optimization problems have constraints in the real world. In this subsection, three well-known constrained engineering design problems, which include speed reducer design problem, tension/compression spring design problem and three-bar truss design problem, are solved to further verify the performance of IHAOHHO. The results of IHAOHHO are compared to the basic AO, HHO, and HOA, SSA, WOA, GWO, MVO as well. The parameter settings are the same as the previous numerical experiments. For all tests, each algorithm is ran 15 times independently. The best result among 15 times for each algorithm and the Wilcoxon signed-rank test results between IHAOHHO and other algorithms are shown in Tables 10–12.

4.4.1. Speed reducer design problem

This problem aims to optimize seven variables to minimize the speed reducer's total weights, which include the face width (x_1), module of teeth (x_2), a discrete design variable on behalf of the teeth in the pinion (x_3), length of the first shaft between bearings (x_4), length of the second shaft between bearings (x_5), diameters of the first shaft (x_6) and diameters of the second shaft (x_7). Four constraints: covering stress, bending stress of the gear teeth, stresses in the shafts and transverse deflections of the shafts as shown in Figure 8 should be satisfied. The mathematical formulation is represented as follows:

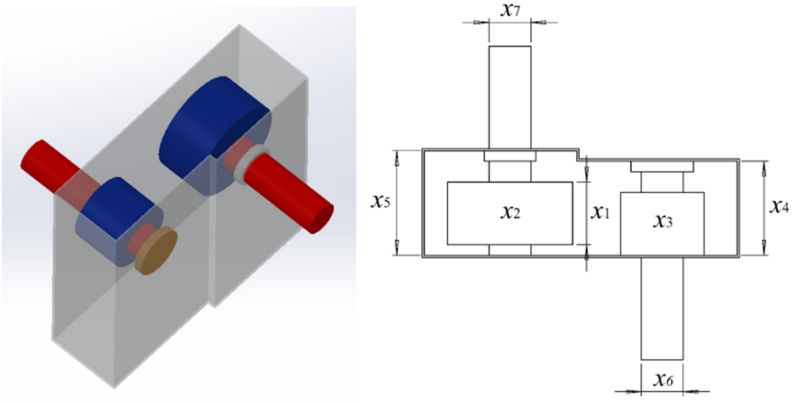


Figure 8. Speed reducer design problem.

Minimize

$$f(\vec{x}) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3),$$

Subject to

$$g_1(\vec{x}) = \frac{27}{x_1x_2^2x_3} - 1 \le 0,$$
$$g_2(\vec{x}) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \le 0,$$
$$g_3(\vec{x}) = \frac{1.93x_4^3}{x_2x_3x_6^4} - 1 \le 0,$$
$$g_4(\vec{x}) = \frac{1.93x_5^3}{x_2x_3x_7^4} - 1 \le 0,$$
$$g_5(\vec{x}) = \frac{\sqrt{(\frac{745x_4}{x_2x_3})^2 + 16.9\times10^6}}{110.0x_6^3} - 1 \le 0,$$
$$g_6(\vec{x}) = \frac{\sqrt{(\frac{745x_4}{x_2x_3})^2 + 157.5\times10^6}}{85.0x_6^3} - 1 \le 0,$$
$$g_7(\vec{x}) = \frac{x_2x_3}{40} - 1 \le 0,$$
$$g_8(\vec{x}) = \frac{5x_2}{x_1} - 1 \le 0,$$
$$g_9(\vec{x}) = \frac{x_1}{12x_2} - 1 \le 0,$$
$$g_{10}(\vec{x}) = \frac{1.5x_6 + 1.9}{x_4} - 1 \le 0,$$
$$g_{11}(\vec{x}) = \frac{1.1x_7 + 1.9}{x_5} - 1 \le 0,$$

Variable range

$$2.6 \le x_1 \le 3.6,$$
$$0.7 \le x_2 \le 0.8,$$
$$17 \le x_3 \le 28,$$
$$7.3 \le x_4 \le 8.3,$$
$$7.8 \le x_5 \le 8.3,$$
$$2.9 \le x_6 \le 3.9,$$
$$5.0 \le x_7 \le 5.5,$$

Compared to other algorithms, IHAOHHO can obviously achieve better results in the speed reducer design problem, as shown in Table 10. P-values in Table 10 show us the significant difference between IHAOHHO and other algorithms, proving the statistical superiority of the proposed algorithm.

Table 10. Comparison of IHAOHHO results with other competitors for the speed reducer design problem.

Algorithm	Optimum variables							Optimum weight	P-value
	x_1	x_2	x_3	x_4	x_5	x_6	x_7		
IHAOHHO	**3.49683**	**0.7**	**17**	**7.33302**	**7.8**	**3.35006**	**5.28575**	**2995.816**	NaN
AO	3.49688	0.7	17	8.10828	7.8	3.37081	5.28578	3008.168	0.025574
HHO	3.49731	0.7	17	7.3	7.8	3.47527	5.28482	3028.6976	0.035339
HOA	3.56008	0.7	17	7.34912	7.8	3.49325	5.28415	3058.577	6.1035e-05
SSA	3.49732	0.7	17	8.03843	7.80061	3.52296	5.28577	3049.1538	0.012451
WOA	3.4976	0.7	17	7.3	7.8	3.44134	5.28525	3019.3398	0.043721
MVO	3.52164	0.7	17	7.44477	8.29729	3.43143	5.2842	3038.4984	0.018066
GWO	3.49231	0.7	17.0038	8.1759	8.04815	3.35214	5.28783	3013.2315	0.0026245

4.4.2. Tension/compression spring design problem

In this case, the intention is to minimize the weight of the tension/compression spring shown in Figure 9. Constraints on surge frequency, shear stress and deflection must be satisfied during optimum design. There are three parameters need to be minimized, including the wire diameter(d), mean coil diameter(D) and the number of active coils (N). The mathematical form of this problem can be written as follows:

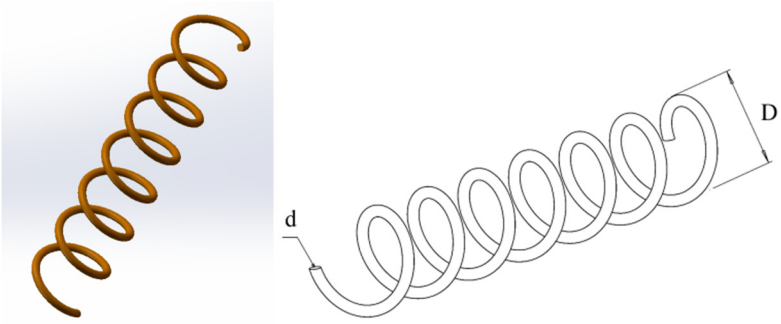


Figure.9 Tension/compression spring design problem.

Consider

$$\vec{x} = [x_1\ x_2\ x_3\ x_4] = [\mathrm{d}\ \mathrm{D}\ \mathrm{N}],.$$

Minimize

$$f(\vec{x}) = (x_3 + 2)x_2x_1^2,$$

Subject to

$$g_1(\vec{x}) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \le 0,$$
$$g_2(\vec{x}) = \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} + \frac{1}{5108 x_1^2} \le 0,$$
$$g_3(\vec{x}) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \le 0,$$
$$g_4(\vec{x}) = \frac{x_1 + x_2}{1.5} - 1 \le 0,$$

Variable range

$$0.05 \le x_1 \le 2.00,$$
$$0.25 \le x_2 \le 1.30,$$
$$2.00 \le x_3 \le 15.00,$$

The experiment results are listed in Table 11 and show that the IHAOHHO can attain the best weight values compared to all other algorithms. IHAOHHO obtains the significant different results compared to others exclude HOA.

Table 11. Comparison of IHAOHHO results with other competitors for the tension/compression spring design problem.

Algorithm	Optimum variables			Optimum weight	P-value
	d	D	N		
IHAOHHO	**0.054826**	**0.49772**	**5.273**	**0.010881**	NaN
AO	0.051647	0.38603	9.3553	0.011692	6.1035e-05
HHO	0.059559	0.64197	3.4141	0.012329	0.047913
HOA	0.054031	0.47388	6.0876	0.011188	**0.63867**
SSA	0.05	0.326589	12.8798	0.012149	0.00030518
WOA	0.059166	0.62905	3.534	0.012186	0.047913
MVO	0.059421	0.63742	3.4573	0.012282	0.025574
GWO	0.057335	0.57116	4.1668	0.011579	6.1035e-05

4.4.3. Three-bar truss design problem

The three-bar truss design problem is a classical optimization application in civil engineering field. The main intention of this case is to minimize the weight of a truss with three bars by considering two structural parameters as illustrated in Figure 10. Deflection, stress and buckling are the three main constrains. The mathematical formulation of this problem is given:

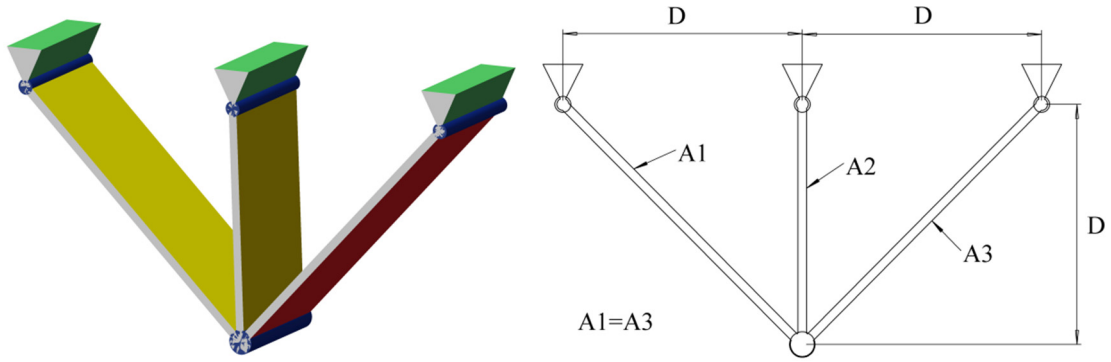


Figure 10. Three-bar truss design problem.

Consider

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

Minimize

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

Subject to

$$g_1(\vec{x}) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \le 0,$$

$$g_2(\vec{x}) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \le 0,$$

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

Consider

$$0 \le x_1, x_2 \le 1,$$

where $l = 100cm, P = 2KN / cm^2, \sigma = 2KN / cm^2$.

Results of IHAOHHO for solving three-bar truss design problem are listed in Table 12. It can be observed that IHAOHHO outperforms other comparative optimization algorithms. Also, it is clear that the p-values between IHAOHHO and other methods are all smaller than 0.05.

As a summary, the experiments in this section prove the superiority of the proposed IHAOHHO algorithm in different characteristics and applications in the real world. IHAOHHO performs better than the basic AO and HHO, and selected well-known algorithms in various degrees, which demonstrates the exploration and exploitation capabilities improvement of IHAOHHO. Excellent performance in solving engineering design problems suggests that IHAOHHO can be widely used in real-world optimization problems.

Table 12. Comparison of IHAOHHO results with other competitors for the three-bar truss design problem.

Algorithm	Optimum variables		Optimum weight	P-value
	x_1	x_2		
IHAOHHO	**0.79182**	**0.39856**	**263.8608**	NaN
AO	0.7932	0.39476	263.8689	0.04126
HHO	0.7824	0.42541	263.8795	0.010254
HOA	0.78192	0.4268	263.8841	0.030151
SSA	0.78139	0.42834	263.8895	6.1035e-05
WOA	0.77541	0.44621	263.9826	0.00012207
MVO	0.78196	0.42668	263.8837	6.1035e-05
GWO	0.79341	0.39418	263.8703	0.00018311

5. Conclusions

In this paper, an improved hybrid Aquila Optimizer and Harris Hawks Optimization algorithm is proposed by combining the exploration part of AO with the exploitation part of HHO. The advantageous parts of basic AO and HHO are combined to keep the well-behaved exploration and exploitation capabilities. Two strategies including representative-based hunting and opposition-based learning are incorporated into the proposed IHAOHHO to further improve the optimization performance. The representative-based hunting strategy can effectively enhance the diversity of the population and fully explore the search space. The opposition-based learning strategy contributes to keep the algorithm from trapping in local optima. This algorithm is evaluated by standard benchmark functions and CEC2017 test functions to analyze its exploration, exploitation and local optima avoidance capabilities. The experiments show competitive results compared to other state-of-the-art meta-heuristic algorithms, which prove that IHAOHHO has better optimization performance than others. Three engineering design problems are solved as well to further verify the superiority of the algorithm, and the results are also competitive with other meta-heuristic algorithms.

The performance of the proposed algorithm on CEC2017 benchmark functions still needs to be improved. The exploration and exploitation capabilities need to be further investigated to break the limitations on CEC2017 test suit. And the transition from exploration to exploitation phase of IHAOHHO is simple. For further work, the transition mechanism can be improved to provide a better balance between the exploration and exploitation phases of this algorithm. Besides, the IHAOHHO algorithm can only solve single-objective optimization problems. Multi-objective version of IHAOHHO may be developed to solve multi-objective problems in the future.

Acknowledgements

This work was partially supported by Sanming University introduces high-level talents to start scientifc research funding support Project (20YG01, 20YG14), Guiding science and technology projects in Sanming City (2020-S-39, 2021-S-8), Educational research projects of young and middle-aged teachers in Fujian Province (JAT200638, JAT200618), Scientifc research and development fund of Sanming University (B202029, B202009), Collaborative education project of industry university

cooperation of the Ministry of Education (202002064014), School level education and teaching reform project of Sanming University (J2010306, J2010305), Higher education research project of Sanming University (SHE2102, SHE2013).

Conflict of interest

On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. I. Boussad, J. Lepagnot, P. Siarry, A survey on optimization metaheuristics, *Inf. Sci.*, **237** (2013), 82–117.
2. T. Dokeroglu, E. Sevinc, T. Kucukyilmaz, A. Cosar, A survey on new generation metaheuristic algorithms, *Comput. Ind. Eng.*, **137** (2019), 106040.
3. K. Hussain, M. Salleh, C. Shi, Y. Shi, Metaheuristic research: a comprehensive survey, *Artif. Intell. Rev.*, **52** (2019), 2191–2233.
4. L. Abualigah, A. Diabat, Advances in sine cosine algorithm: a comprehensive survey, *Artif. Intell. Rev.*, **54** (2021), 2567–2608.
5. L. Abualigah, A. Diabat, A comprehensive survey of the Grasshopper optimization algorithm: results, variants, and applications, *Neural Comput. Appl.*, **32** (2020), 15533–15556.
6. J. H. Holland, Genetic algorithms, *Sci. Am.*, **267** (1992), 66–72.
7. R. Storn, K. Price, Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces, *J. Glob. Optim.*, **11** (1997), 341–359.
8. I. Rechenberg, Evolutionsstrategien, in *Simulationsmethoden in der Medizin und Biologie*, Springer, Berlin, Heidelberg, (1978), 83–114.
9. D. Simon, Biogeography-based optimization, *IEEE Trans. Evol. Comput.*, **12** (2008), 702–713.
10. D. Dasgupta, Z. Michalewicz, Evolutionary algorithms in engineering applications, *DBLP*, 1997.
11. S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, Optimization by simmulated annealing, *Science*, **220** (1983), 671–80.
12. E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA: a gravitational search algorithm, *Inf. Sci.*, **179** (2009), 2232–2248.
13. A. Hatamlou, Black hole: a new heuristic optimization approach for data clustering, *Inf. Sci.*, **222** (2013), 175–84.
14. S. Mirjalili, S. M. Mirjalili, A. Hatamlou, Multi-Verse Optimizer: a nature-inspired algorithm for global optimization, *Neural Comput. Appl.*, **27** (2015), 495–513.
15. S. Mirjalili, SCA: A sine cosine algorithm for solving optimization problems, *Knowl.-Based Syst.*, **96** (2016).
16. L. Abualigah, A. Diabat, S. Mirjalili, M. A. Elaziz, A. H. Gandomi, The arithmetic optimization algorithm, *Comput. Methods Appl. Mech. Eng.*, **376** (2021), 113609.
17. F. Asef, V. Majidnezhad, M. R. Feizi-Derakhshi, S. Parsa, Heat transfer relation-based optimization algorithm (HTOA), *Soft Comput.*, (2021), 1–30.
18. J. Kennedy, R. Eberhart, Particle swarm optimization, in *Proceedings of the 1995 IEEE international conference on neural networks (ICNN '93)*, IEEE, **4** (1995), 1942–1948.

19. M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization, *IEEE Comput. Intell.*, **1** (2006), 28–39.
20. X. S. Yang, Firefly algorithm, stochastic test functions and design optimisation, *Int. J. Bio-Inspired Comput.*, **2** (2010), 78–84.
21. S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey wolf optimizer, *Adv. Eng. Software*, **69** (2014), 46–61.
22. A. H. Gandomi, X. S. Yang, A. H. Alavi, Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems, *Eng. Comput.*, **29** (2013), 17–35.
23. S. Mirjalili, A. Lewis, The whale optimization algorithm, *Adv. Eng. Software*, **95** (2016), 51–67.
24. S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, S. M. Mirjalili, Salp swarm algorithm: A bio-inspired optimizer for engineering design problems, *Adv. Eng. Software*, **114** (2017), 163–191.
25. H. Jia, X. Peng, C. Lang, Remora optimization algorithm, *Expert Systems with Applications*, **185** (2021), 115665.
26. S. M. Li, H. L. Chen, M. J. Wang, A. A. Heidari, S. Mirjalili, Slime mould algorithm: A new method for stochastic optimization, *Future Gener. Comput. Syst.*, **111** (2020), 300–323.
27. F. Miarnaeimi, G. Azizyan, M. Rashki, Horse herd optimization algorithm: a nature-inspired algorithm for high-dimensional optimization problems, *Knowl.-Based Syst.*, **213** (2020).
28. L. Abualigah, D. Yousri, M. A. Elaziz, A. A. Ewees, M. A. A. Al-qaness, A. H. Gandomi, Aquila Optimizer: a novel meta-heuristic optimization algorithm, *Comput. Ind. Eng.*, **157** (2021), 107250.
29. A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. L. Chen, Harris Hawks optimization: algorithm and applications, *Future Gener. Comput. Syst.*, **97** (2019), 849–872.
30. A. M. AlRassas, M. A. A. Al-qaness, A. A. Ewees, S. Ren, M. Abd Elaziz, R. Damaševičius, et al., Optimized ANFIS model using Aquila Optimizer for oil production forecasting, *Processes*, **9** (2021), 1194.
31. C. Hao, A. A. Heidari, H. Chen, M. Wang, Z. Pan, A. H. Gandomi, Multi-population differential evolution-assisted harris hawks optimization: framework and case studies, *Future Gener. Comput. Syst.*, **111** (2020), 175–198.
32. M. A. Al-Betar, M. A. Awadallah, A. A. Heidari, H. Chen, C. Li, Survival exploration strategies for harris hawks optimizer, *Expert Syst. Appl.*, **168** (2020), 114243.
33. S. Song, P. Wang, A. A. Heidari, M. Wang, S. Xu, Dimension decided harris hawks optimization with gaussian mutation: balance analysis and diversity patterns, *Knowl.-Based Syst.*, **215** (2020), 106425.
34. D. Yousri, S. Mirjalili, J. A. T. Machado, S. B. Thanikantie, O. Elbaksawi, A. Fathy, Efficient fractional-order modified Harris Hawks optimizer for proton exchange membrane fuel cell modeling, *Eng. Appl. Artif. Intell.*, **100** (2021), 104193.
35. S. Gupta, K. Deep, A. A. Heidari, H. Moayedi, M. Wang, Opposition-based learning Harris hawks optimization with advanced transition rules: Principles and analysis, *Expert Syst. Appl.*, **158** (2020), 113510.
36. O. Akdag, A. Ates, C. Yeroglu, Modification of harris hawks optimization algorithm with random distribution functions for optimum power flow problem, *Neural Comput. Appl.*, **33** (2021).
37. D. Yousri, A. Fathy, S. B. Thanikanti, Recent methodology based Harris Hawks optimizer for designing load frequency control incorporated in multi-interconnected renewable energy plants, *Sustainable Energy Grids Networks*, **22** (2020), 100352.

38. H. Jia, C. Lang, D. Oliva, W. Song, X. Peng, Dynamic harris hawks optimization with mutation mechanism for satellite image segmentation, *Remote Sens.*, **11** (2019), 1421.
39. K. Hussain, N. Neggaz, W. Zhu, E. H. Houssein, An efficient hybrid sine-cosine Harris hawks optimization for low and high-dimensional feature selection, *Expert Syst. Appl.*, **176** (2021), 114778.
40. X. Bao, H. Jia, C. Lang, A novel hybrid harris hawks optimization for color image multilevel thresholding segmentation, *IEEE Access*, **7** (2019), 76529–76546.
41. E. H. Houssein, M. E. Hosney, M. Elhoseny, D. Oliva, M. Hassaballah, Hybrid Harris hawks optimization with cuckoo search for drug design and discovery in chemoinformatics, *Sci. Rep.*, **10** (2020), 14439.
42. A. Kaveh, P. Rahmani, A. D. Eslamlou, An efficient hybrid approach based on Harris Hawks optimization and imperialist competitive algorithm for structural optimization, *Eng. Comput.*, (2021), 4598.
43. A. Auger, N. Hansen, A restart cma evolution strategy with increasing population size, *IEEE Congr. Evol. Comput.*, **2** (2005), 1769–1776.
44. R. Tanabe, A. S. Fukunaga, Improving the search performance of SHADE using linear population size reduction, *IEEE Congr. Evol. Comput.*, 2014.
45. G. Dhiman, SSC: A hybrid nature-inspired meta-heuristic optimization algorithm for engineering applications, *Knowl.-Based Syst.*, **222** (2021), 106926.
46. I. Ahmadianfar, A. A. Heidari, A. H. Gandomi, X. Chu, H. Chen, RUN beyond the metaphor: an efficient optimization algorithm based on Runge Kutta method, *Expert Syst. Appl.*, **181** (2021), 115079.
47. M. Banaie-Dezfouli, M. H. Nadimi-Shahraki, Z. Beheshti, R-gwo: representative-based grey wolf optimizer for solving engineering problems, *Appl.Soft Comput.*, (2021), 107328.
48. H. Tizhoosh, Opposition-based learning: A new scheme for machine intelligence, in *Proceedings of the International Conference on Computational Intelligence for Modeling*, (2005), 695–701.
49. S. Rahnamayan, H. R. Tizhoosh, M. M. A. Salama, Opposition-based differential evolution, *IEEE Trans. Evol. Comput.*, **12** (2014), 64–79.
50. Z. Jia, L. Li, S. Hui, Artificial bee colony using opposition-based learning, *Adv. Intell. Syst. Comput.*, **329** (2015), 3–10.
51. M. A. Elaziz, D. Oliva, S. Xiong, An improved opposition-based sine cosine algorithm for global optimization, *Expert Syst. Appl.*, **90** (2017), 484–500.
52. A. A. Ewees, M. A. Elaziz, E. H. Houssein, Improved grasshopper optimization algorithm using opposition-based learning, *Expert Syst. Appl.*, **112** (2018), 156–172.
53. C. Fan, N. Zheng, J. Zheng, L. Xiao, Y. Liu, Kinetic-molecular theory optimization algorithm using opposition-based learning and varying accelerated motion, *Soft Comput.*, **24** (2020).
54. N. H. Awad, M. Z. Ali, P. N. Suganthan, J. J. Liang, B. Y. Qu, Problem definitions and evaluation criteria for the CEC2017, in *Special Session and Competition on Single Objective Real-Parameter Numerical Optimization*, IEEE Congress on Evolutionary Computation, 2017.

AIMS Press