

Research Article

Multistrategy Harris Hawks Optimization Algorithm Using Chaotic Method, Cauchy Mutation, and Elite Individual Guidance

Lei Wen ¹, Guopeng Wang ², Longwang Yue ³, Xiaodan Liang ¹,
and Hanning Chen ¹

¹School of Computer Science and Technology, Tiangong University, Tianjin, China

²The Open University of China, Research and Development Center for E-Learning, Ministry of Education, Beijing, China

³School of Mechanical and Electrical Engineering, Henan University of Technology, Zhengzhou, Henan, China

Correspondence should be addressed to Guopeng Wang; wangguopeng@sohu.com and Longwang Yue; gfh110058@163.com

Received 14 June 2022; Revised 13 July 2022; Accepted 5 August 2022; Published 28 August 2022

Academic Editor: Shi Cheng

Copyright © 2022 Lei Wen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Aiming at the shortcomings of the Harris hawks optimization algorithm (HHO), such as poor initial population diversity, slow convergence speed, poor local optimization ability, and easily falling into local optimum, a Harris hawks optimization algorithm (CCCHHO) integrating multiple mechanisms is proposed. First, the population diversity is enhanced by the initialization of the chaotic method. Second, the cosine function is used to better simulate the characteristics of the periodic change of the energy of the prey in the repeated contests with the group of hawks, to better balance the exploration and exploitation of the algorithm. Third, Cauchy mutation on the optimal individual in the exploration phase is performed, and the characteristics of the Cauchy distribution to enhance the diversity of the population are used, which can effectively prevent the algorithm from falling into the local optimum. Fourth, the local optimization ability of the algorithm by using the ergodicity of the chaotic system in the exploitation phase to perform a chaotic local search for the optimal individual is enhanced, which can effectively jump out after the algorithm falls into the local optimum. Finally, we use the elite individuals of the population to guide the position update of the population's individuals, fully communicate with the dominant individuals, and speed up the convergence speed of the algorithm. Through the simulation experiments on CCCHHO with 11 different benchmark functions, CCCHHO is better than the gray wolf optimization algorithm (GWO), the Salp swarm algorithm (SSA), the ant lion optimization algorithm (ALO), and three improved HHO algorithms in terms of convergence speed and optimization accuracy, whether it is a unimodal benchmark function or a multimodal benchmark function. The experimental results show that CCCHHO has excellent algorithm efficiency and robustness.

1. Introduction

In recent years, meta-heuristic algorithms have attracted more and more scholars' attention. Because meta-heuristic algorithms are simple to implement, have no layer dependencies, and can jump out of local optima, they are used to solve different problems [1–4].

Meta-heuristic algorithms can be divided into four categories: physics-based, evolution-based, population-based, and human-based. The physics-based aspect mainly simulates the physical rules in the universe. Common

physics-based meta-heuristic algorithms include gravitational search algorithm (GSA) [5], central force optimization (CFO) [6], and Black Hole optimization algorithm (BH) [7]. The evolution-based methods are inspired by biological evolution and classical evolutionary algorithms such as genetic algorithm (GA) [8], differential evolution algorithm (DE) [9], and biogeography-based optimizer (BBO) [10]. The population-based algorithms are inspired by the collective behavior of biological populations and classical swarm intelligence algorithms such as ant colony optimization (ACO) [11], particle swarm optimization algorithm (PSO) [12],

artificial bee colony optimization algorithm (ABC) [13], and Monarch butterfly optimization algorithm [14]. More novel population-based algorithms are gray wolf optimization algorithm (GWO) [15], wolf pack optimization algorithm (WPA) [16], dragonfly algorithm (DA) [17], whale optimization algorithm (WOA) [18], ant lion optimization algorithm (ALO) [19], and Salp swarm algorithm (SSA) [20]. The human-based algorithms are inspired by human behavior, such as brain storm optimization (BSO) [21].

Harris hawks optimization algorithm (HHO) [22] is a swarm intelligence optimization algorithm proposed in recent years, which is derived from the group hunting behavior of Harris hawks. Because HHO has a simple structure, easy implementation, and high performance, it has attracted the attention of a large number of researchers since the algorithm was proposed. The researchers have improved the basic HHO algorithm in different aspects, and some of the improved algorithms have been applied to different fields. ElSayed et al. [23] used HHO combined with sequential quadratic programming (SQP) to the optimal coordination problem of directional overcurrent relays incorporating distributed generation. Abbasi et al. [24] used the chaos method, Gaussian mutation, differential evolution, and other methods to improve the basic HHO and apply the improved algorithm to the fatigue life analysis of tapered roller bearings. Jouhari et al. [25] introduced the Salp swarm algorithm (SSA) into the basic HHO, and the position update mechanism is selected through a parameter, and the improved algorithm is applied to the scheduling problem. Chen et al. [26] introduced chaotic drift mechanism into the basic HHO to improve the algorithm, and the improved algorithm is applied to the parameter identification problem of photovoltaic cells and modules. Jia et al. [27] used dynamic parameters to adjust prey escape energy factor and mutation mechanism to improve the basic HHO, and used the improved algorithm for the satellite image segmentation problem. Qu et al. [28] enhanced the information exchange between population individuals and introduced escape energy factors with chaotic disturbances to improve the basic HHO algorithm.

Although different scholars have proposed different improved HHO algorithms, they are not suitable for all optimization problems. Based on this, it makes sense to develop more efficient and accurate algorithms. In this article, a CCCHHO algorithm that integrates multiple strategies is proposed. The algorithm initializes the population through the chaotic method and uses the ergodicity of the chaotic system to enhance the diversity of the population, and uses the characteristics of the cosine function to periodically trigger the update of the prey escape energy that more efficiently balances the exploration and exploitation of the algorithm, and introduces a mutation strategy to enhance the global exploration ability that can prevent the algorithm from falling into the local optimum. The use of chaotic local search can effectively jump out after the algorithm falls into the local optimum and improve the local optimization ability of the algorithm at the same time. Finally, the elite individual guidance mechanism is used to update the population's individual position and use the greedy mechanism to obtain the optimal population as the initial population of the next iteration, which enhances the diversity of the population to accelerate the convergence speed of the algorithm. In order to

verify the performance of the proposed algorithm, 11 benchmark test functions are used for simulation analysis and compared with other meta-heuristic algorithms and some improved HHO algorithms.

The structure of this article is as follows. Section 2 introduces the basic HHO algorithm. Section 3 introduces the proposed CCCHHO algorithm. The fourth section is a simulation experiment, and the results are analyzed at the same time. Section 5 expounds on the conclusions and future work.

2. Harris Hawks Optimization Algorithm (HHO)

The Harris hawks optimization algorithm is a meta-heuristic algorithm inspired by the Harris hawks hunting. HHO contains multiple search mechanisms depending on the different strategies adopted by the hawk at different phases. A detailed description of these search mechanisms is given below.

2.1. Exploration Phase. At this phase, the Harris hawks update position with two different strategies based on the probability q . The formulas are as follows:

$$X(t+1) = \begin{cases} X_{\text{rand}}(t) - r_1 |X_{\text{rand}}(t) - 2r_2 X(t)|, & q \geq 0.5, \\ X_{\text{rabbit}}(t) - X_m(t) - r_3 (\text{LB} + r_4 (\text{UB} - \text{LB})), & q < 0.5, \end{cases} \quad (1)$$

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

where $X(t+1)$ and $X(t)$ represent the position vector of hawks in iteration $t+1$ and t , respectively, $X_{\text{rabbit}}(t)$ is the position of rabbit, $X_m(t)$ is the average position of the current population of hawks, $X_{\text{rand}}(t)$ represents randomly selected individuals in the current population, r_1, r_2, r_3, r_4 , and q are random numbers between 0 and 1, and LB and UB are the upper and lower bounds of the population, respectively, and N is the population number.

2.2. Transition from Exploration to Exploitation. HHO realizes the transformation of exploration and exploitation through the escape energy factor E . The formula as follows:

$$E = 2E_0 \left(1 - \frac{t}{T}\right), \quad (3)$$

where E_0 is a random number between -1 and 1 , T is the maximum number of iterations, and t is the current number of iterations. When $|E| \geq 1$, HHO performs the global search. Otherwise, local exploitation is performed.

2.3. Exploitation Phase. During the exploitation phase, HHO uses four different strategies to update location and decides which strategy to use by the escape energy factor E and the random number $r \in (0, 1)$, and r indicates the

chance of the prey to escape before the surprise pounce. When $r \geq 0.5$, it cannot escape. When $r < 0.5$, it can.

2.4. Soft Besiege. When $r \geq 0.5$ and $|E| \geq 0.5$, the prey has enough energy to try to escape by jumping, but ultimately, it cannot. The hawk's position update formula is as follows:

$$X(t+1) = X_{\text{rabbit}}(t) - X(t) - E|JX_{\text{rabbit}}(t) - X(t)|, \quad (4)$$

where $J = 2(1 - r_5)$ represents the random jump intensity when the prey escapes, and r_5 is a random number between 0 and 1.

2.5. Hard Besiege. When $r \geq 0.5$ and $|E| < 0.5$, the prey is captured by the hawks with lower energy. The hawk's position update formula is as follows:

$$X(t+1) = X_{\text{rabbit}}(t) - E|X_{\text{rabbit}}(t) - X(t)|. \quad (5)$$

2.6. Soft Besiege with Progressive Rapid Dives. When $r < 0.5$ and $|E| \geq 0.5$, the prey has enough energy to ensure a successful escape, so the soft besiege strategy of fast dive is implemented, and the position update formulas are as follows:

$$Y = X_{\text{rabbit}}(t) - E|JX_{\text{rabbit}}(t) - X(t)|, \quad (6)$$

$$Z = Y + S \times \text{LF}(D), \quad (7)$$

$$X(t+1) = \begin{cases} Y, & f(Y) < f(X(t)), \\ Z, & f(Z) < f(X(t)), \end{cases} \quad (8)$$

where D is the problem dimension, S is the D dimension random row vector, and LF is the Levf function. The formula is as follows:

$$\text{LF}(x) = 0.01 \times \frac{u \times \sigma}{|v|^{1/\beta}}, \quad (9)$$

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

where u and v are random numbers between 0 and 1, and β is 1.5.

2.7. Hard Besiege with Progressive Rapid Dives. When $r < 0.5$ and $|E| < 0.5$, prey energy is lower, but escape is still possible. A hard besiege strategy of rapid dive is implemented to reduce the average distance from the prey, and the position update formulas are as follows:

$$X(t+1) = \begin{cases} Y, & f(Y) < f(X(t)), \\ Z, & f(Z) < f(X(t)), \end{cases} \quad (10)$$

$$Y = X_{\text{rabbit}}(t) - E|JX_{\text{rabbit}}(t) - X_m(t)|, \quad (11)$$

$$Z = Y + S \times \text{LF}(D), \quad (12)$$

where D is the problem dimension, S is the D dimension random row vector, LF is the Levf function (Equation (9)), $X_{\text{rabbit}}(t)$ is the position of rabbit, and $X_m(t)$ is the average position of the current population of hawks.

3. Harris Hawks Optimization Algorithm with Multiple Strategies (CCCHHO)

Aiming at the problems existing in the HHO algorithm, the article improves the HHO through various aspects. By introducing the method of Logistic map to replace the random initialization method of the basic HHO algorithm, the characteristics of the chaotic system are used to help the algorithm generate more diverse populations. In the exploration phase, the Cauchy mutation strategy is used to enhance the global search ability that helps the algorithm jump out of the local optimum. A new formula is used to update the energy factor E to better balance the global search and local exploitation. In the exploitation phase, the chaotic local search strategy is used to enhance the local search ability. Finally, introduce the elite guidance strategy, which uses the dominant group to guide the update of the population individuals, and then, use the greedy strategy to save the better individuals to speed up the convergence speed of the algorithm.

3.1. Chaotic Maps. Chaotic systems have the characteristics of randomness and ergodicity. More diverse populations can be generated by using these characteristics, thereby improving the performance and speeding up the convergence speed of the algorithm. Kaur et al. [29] used the chaotic map, which generates the initialization population instead of the randomly generated population, which makes the improved algorithm easy to jump out of the local optimum and improves the optimization performance of the whale algorithm. At present, there are many different chaotic maps in the optimization field [30], mainly including Logistic map, Tent map, and Gauss map. This article uses the Logistic map to generate the initialization population. The Logistic map is defined as follows:

$$x_{i+1} = \mu x_i (1 - x_i), \quad i = 1, 2, \dots, N - 1, \quad (13)$$

where μ is 4, x_1 is a random number between 0 and 1, and N is the number of population's individuals.

The initial population position generated by using the Logistic map is more uniform distribution of population position compared to the randomly generated, which increases the diversity of the population and expands the search range of the hawks in space. To a certain extent, it improves the shortcomings of HHO that it is easy to fall into local optimum.

3.2. Cauchy Mutation. Cauchy mutation originates from the Cauchy distribution, and the standard Cauchy distribution probability density is as follows:

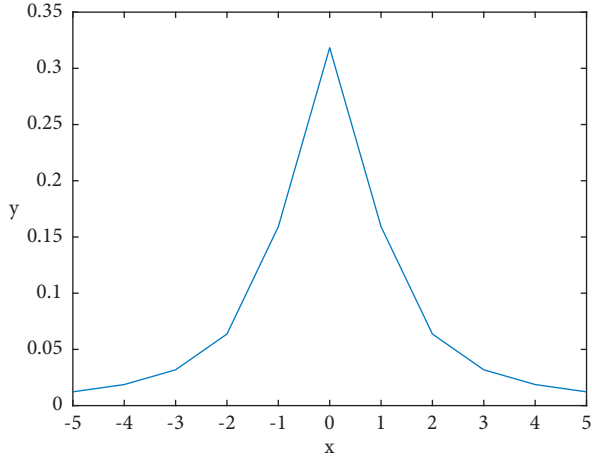


FIGURE 1: Standard Cauchy distribution probability density function curve.

$$f(x) = \frac{1}{\pi} \cdot \frac{1}{1+x^2}, \quad x \in (-\infty, +\infty). \quad (14)$$

Figure 1 shows that the probability distribution of the Cauchy distribution function is closer to the horizontal axis in the horizontal direction, and the slower it changes. Therefore, the Cauchy distribution can be regarded as infinite. So in terms of probabilities, the Cauchy distribution has a wider distribution range [31]. This means that by using the random numbers generated by the Cauchy distribution as the perturbation factor in the optimization process, one can obtain a relatively broad search space, which can prevent the algorithm from falling into the local optimum in the exploration phase in a way, and after falling into the local optimum, it is easier to jump out the optimum. After obtaining the optimal solution of the current population, use the following formula to update the current global optimal solution:

$$X'_{\text{best}} = X_{\text{best}} + X_{\text{best}} \times \text{cauchy}(0, 1) \frac{1}{2}. \quad (15)$$

Here, $\text{cauchy}(0, 1)$ is the standard Cauchy distribution.

3.3. Nonlinear Energy Factor E Based on Cosine Strategy.

In the HHO algorithm, the energy factor E is an important parameter to balance the exploration phase and the exploitation phase. The larger $|E|$, the more inclined the HHO algorithm is to perform the exploration. Conversely, the more inclined it is to perform the exploitation. In the HHO algorithm, the energy factor E decreases linearly from large to small, which cannot effectively describe the real situation of Harris hawks rounding up prey in nature. In the multi-round game of the Harris hawks and the prey, the energy of the prey cannot be simply reflected by linear changes. The energy of the prey should change periodically and eventually reach zero to be captured by the hawks. During each round of rounding up, the prey will get a short rest to recover a small amount of energy, but over time, the energy recovered

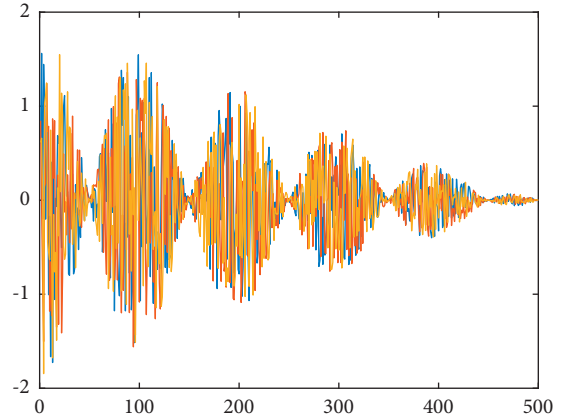


FIGURE 2: Energy escape factor.

by the prey will gradually decrease until the energy reaches zero. In this article, the cosine function is used to describe the periodic change of prey energy. The formula is as follows:

$$E = 2 \times \cos\left(\frac{5\pi t}{T}\right) \times \left(1 - \frac{t}{T}\right) \times (2\text{rand} - 1). \quad (16)$$

Figure 2 shows the change of energy escape factor during iteration. As we can see, the energy escape factor changes periodically until it becomes zero, which well describes the energy change of prey in the process of being rounded up.

3.4. Chaotic Local Search. Local search is one of the effective methods to prevent the algorithm from falling into the local optimum. In most cases, the solution is near the local optimum, and the algorithm cannot obtain it. After the algorithm falls into the local optimum, by using the local search method to search for the vicinity of the local optimal solution, one can effectively jump out of the local optimum and improve the performance of the algorithm. However, sometimes, the local method does not produce ideal results. The ergodic characteristics of chaotic systems can be considered, and the system starts from an initial state and follows its own motion law and experiences all state points in its attraction space without repetition for a long enough time. If a chaotic sequence of length k is superimposed on the optimal individual of the current population, it is equivalent to carrying out k times local search without repetition in the neighbourhood of the optimal individual of the current population. Introduce the chaotic local search strategy [32], and the ergodicity of the chaotic system can effectively prevent the algorithm from falling into the local area, and at the same time, the search efficiency and search range are enhanced compared with local search. The pseudocode of the chaotic local search algorithm is given in Algorithm 1.

In this article, Equation (13) is used in the chaotic search algorithm that the Logistic map produces a chaotic sequence. The chaotic sequence value generated by the Logistic map is between 0 and 1. If the value is directly superimposed on any dimension of the optimal

```

Initialization chaotic search times  $D$ 
Generate a chaotic sequence of length  $D$  by using a chaotic map
Get the best individual in the current population  $X_{\text{best}}$ 
Setting up the chaotic search counter  $k = 0$ 
While  $k < D$ 
    Superimpose an item of the chaotic sequence on any dimension in  $X_{\text{best}}$  to form a new individual  $X_{\text{best}}^{\text{cls}}$ 
    Calculate the fitness value  $f(X_{\text{best}}^{\text{cls}})$  of new individual  $X_{\text{best}}^{\text{cls}}$ 
    if  $f(X_{\text{best}}^{\text{cls}}) < f(X_{\text{best}})$ 
        Update the optimal individual and fitness value of the current population
    end if
     $k = k + 1$ 
end while

```

ALGORITHM 1: Chaotic local search.

```

Initialize population size  $N$ , number of iterations  $T$ 
Use Equation (13) to perform chaotic strategy initialization population  $X_i$  ( $i = 1, 2, \dots, N$ )
Set the current number of iterations  $t = 0$ 
while  $t < T$ 
    Calculate the fitness value of hawks
    Set the best position as the position of the prey  $X_{\text{rabbit}}$ 
    for each individual  $X_i$ 
        Update escape energy  $E$  and jump strength  $J$  by using Equation (16)
        if  $(|E| \geq 1)$ 
            Update location by using Equation (1)
            Get the optimum of the current population and use Equation (15) to carry out the Cauchy mutation
        end if
        if  $(|E| < 1)$ 
            if  $(r \geq 0.5, |E| \geq 0.5)$ 
                Update location by using Equation (4)
            else if  $(r \geq 0.5, |E| < 0.5)$ 
                Update location by using Equation (5)
            else if  $(r < 0.5, |E| \geq 0.5)$ 
                Update location by using Equation (8)
            else if  $(r < 0.5, |E| < 0.5)$ 
                Update location by using Equation (10)
            end if
            Carry out chaotic local search
        end if
    end for
    Carry out elite individual guidance
     $t = t + 1$ 
end while

```

ALGORITHM 2: CCCHHO algorithm.

individual, the search will only be carried out in one direction, so that the local search performance will be greatly reduced. To solve this problem, this article uses

the number that chaotic search counter k to periodically adjust the search direction. The position update formula is as follows:

TABLE 1: Description of benchmark functions.

Function	V_no	Range	f_{\min}
$F_1(x) = \sum_{i=1}^n x_i^2$	30	[-100, 100]	0
$F_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	[-10, 10]	0
$F_3(x) = \max\{ x_i , 1 \leq i \leq n\}$	30	[-100, 100]	0
$F_4(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[-30, 30]	0
$F_5(x) = \sum_{i=1}^n ([x_i + 0.5])^2$	30	[-100, 100]	0
$F_6(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	30	[-500, 500]	-418.9829×5
$F_7(x) = -20 \exp(-0.2 \sqrt{(1/n) \sum_{i=1}^n x_i^2}) - \exp((1/n) \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	30	[-32, 32]	0
$F_8(x) = (\pi/n) \{10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$ $y_i = 1 + (x_i + 1/4)u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m x_i > a, \\ 0 - a < x_i < a, \\ k(-x_i - a)^m x_i < -a, \end{cases}$	30	[-50, 50]	0
$F_9(x) = 0.1 \{ \sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	30	[-50, 50]	0
$F_{10}(x) = ((1/500) + \sum_{j=1}^{25} (1/j + \sum_{i=1}^2 (x_i - a_{ij})^6))^{-1}$	2	[-65, 65]	1
$F_{11}(x) = \sum_{i=1}^{11} [a_i - (x_1(b_i^2 + b_i x_2)/b_i^2 + b_i x_3 + x_4)]^2$	4	[-5, 5]	0.00030

$$X_{\text{best}}^{\text{cls}} = X_{\text{best}} + (-1)^k \beta_k, \quad (17)$$

where X_{best} is the optimal individual in the current population, $X_{\text{best}}^{\text{cls}}$ is the new individual after a chaotic local search, k is the number that chaotic search counter, and β_k is the chaotic sequence value generated by the Logistic map.

3.5. Elite Individual Guidance Mechanism. In the gray wolf optimization algorithm, it is proposed that the position of the wolf pack follows the guidance of three individuals in the group, α wolf, β wolf, and δ wolf, which can effectively improve the search efficiency. Based on this, this article introduces an elite individual guidance mechanism. The top three better individuals in the current population are selected as leaders to guide the location update of other individuals, so that the information exchange between the population individuals and the better individual in the population is strengthened, thereby enhancing the diversity of the population. The hawk's position update formulas are as follows:

$$X' = X + A * (2X_\alpha - X_\beta - X_\delta), \quad (18)$$

$$A = 2ar - a, \quad (19)$$

where X is the current individual position, X' is the updated position, X_α is the optimal solution in the current population, X_β is the suboptimal solution in the current population, X_δ is the third optimal solution in the current population, a is linearly decreased from 2 to 0 in the iterative process, and r is a random number between 0 and 1.

Based on the greedy mechanism, the better one is selected as a new individual, and the rabbit position and rabbit energy are updated at the same time.

3.6. The Pseudocode of CCCHHO. Algorithm 2 pseudocode describes the details of CCCHHO.

4. Results and Discussion

In order to verify the performance of the proposed CCCHHO algorithm, this article tests the algorithm through 11 benchmark functions and gives the experimental results and analysis. The test functions can be divided into two groups: F1~F5 are unimodal benchmark functions, in which the characteristics with the unique global optimal value can evaluate the local exploitation capabilities of different optimization algorithms, and F6~F11 are multimodal benchmark functions, which can evaluate the global exploration of different algorithms and the ability to avoid local optimum. According to the best, mean, worst, and standard deviation of the results, the proposed CCCHHO is compared with other optimization algorithms, such as GWO, SSA, and ALO. At the same time, it is compared with the other improved HHO algorithms. The information of the benchmark functions F1-F11 is shown in Table 1.

4.1. Experimental Settings. All algorithms run using MATLAB R2019b on a computer with 16G memory, AMD4800U. The population size of all algorithms is set to 30, the maximum number of iterations is set to 500, and each algorithm loop executes 30 times.

TABLE 2: The result of the comparison of CCCHHO with HHO, GWO, SSA, and ALO.

Function	Index	CCCHHO	HHO	GWO	SSA	ALO
F1	Best	0.00E+00	7.44E-115	6.94E-30	3.42E-08	2.25E-09
	Avg	0.00E+00	1.42E-98	9.05E-28	6.22E-07	9.83E-09
	Worst	0.00E+00	2.90E-97	5.04E-27	7.13E-06	5.86E-08
	Std	0.00E+00	5.43E-98	1.33E-27	1.50E-06	1.00E-08
F2	Best	6.30E-238	2.38E-60	1.93E-17	5.10E-06	1.68E-05
	Avg	1.69E-180	5.47E-52	8.89E-17	2.09E-03	5.72E-01
	Worst	5.07E-179	4.53E-51	1.99E-16	5.18E-02	3.95E+00
	Std	0.00E+00	1.22E-51	5.33E-17	9.47E-03	9.83E-01
F3	Best	3.04E-239	2.83E-57	9.35E-08	9.21E-06	2.04E-04
	Avg	3.72E-209	3.16E-50	6.08E-07	2.07E-05	4.69E-03
	Worst	1.07E-207	1.73E-49	1.88E-06	3.48E-05	5.53E-02
	Std	0.00E+00	5.37E-50	5.15E-07	6.34E-06	1.16E-02
F4	Best	0.00E+00	3.34E-04	2.57E+01	3.84E+00	5.57E-04
	Avg	0.00E+00	1.15E-02	2.69E+01	1.34E+02	1.07E+02
	Worst	0.00E+00	6.87E-02	2.87E+01	2.03E+03	6.67E+02
	Std	0.00E+00	1.61E-02	8.70E-01	3.79E+02	1.82E+02
F5	Best	0.00E+00	2.60E-10	5.27E-05	4.30E-10	1.83E-09
	Avg	0.00E+00	1.01E-04	7.06E-01	9.71E-10	1.11E-08
	Worst	0.00E+00	3.43E-04	1.43E+00	1.98E-09	8.91E-08
	Std	0.00E+00	1.08E-04	3.59E-01	4.14E-10	1.57E-08
F6	Best	-1.26E+04	-1.26E+04	-7.79E+03	-3.71E+03	-3.56E+03
	Avg	-1.26E+04	-1.26E+04	-6.02E+03	-2.83E+03	-2.54E+03
	Worst	-1.26E+04	-1.23E+04	-3.77E+03	-2.25E+03	-1.81E+03
	Std	1.85E-12	5.55E+01	7.57E+02	3.44E+02	5.44E+02
F7	Best	8.88E-16	8.88E-16	7.19E-14	8.15E-06	2.33E-05
	Avg	8.88E-16	8.88E-16	1.05E-13	7.30E-01	3.82E-01
	Worst	8.88E-16	8.88E-16	1.36E-13	2.32E+00	2.01E+00
	Std	0.00E+00	0.00E+00	1.65E-14	9.43E-01	6.73E-01
F8	Best	1.57E-32	3.91E-08	1.28E-02	1.33E-11	9.98E-10
	Avg	1.57E-32	5.78E-06	6.31E-02	9.92E-01	2.78E+00
	Worst	1.57E-32	2.89E-05	5.57E-01	6.37E+00	8.15E+00
	Std	5.57E-48	8.34E-06	9.56E-02	1.44E+00	2.58E+00
F9	Best	1.35E-32	1.28E-07	2.01E-01	5.81E-11	2.15E-09
	Avg	1.35E-32	1.01E-04	5.58E-01	2.56E-03	2.13E-03
	Worst	1.35E-32	5.26E-04	9.24E-01	1.10E-02	3.08E-02
	Std	5.57E-48	1.29E-04	1.93E-01	4.73E-03	6.37E-03
F10	Best	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01
	Avg	9.98E-01	1.49E+00	4.23E+00	1.13E+00	2.15E+00
	Worst	9.98E-01	5.93E+00	1.27E+01	1.99E+00	7.87E+00
	Std	2.40E-16	9.62E-01	3.80E+00	3.44E-01	1.76E+00
F11	Best	3.08E-04	3.09E-04	3.08E-04	5.14E-04	5.06E-04
	Avg	3.26E-04	4.50E-04	1.81E-03	4.19E-03	4.84E-03
	Worst	3.65E-04	1.40E-03	2.04E-02	2.04E-02	2.12E-02
	Std	1.55E-05	3.06E-04	5.05E-03	7.37E-03	8.01E-03

4.2. Comparison of CCCHHO and Other Optimization Algorithms. This section compares CCCHHO with four optimization algorithms: HHO, GWO, SSA, and ALO. Record the optimal value (best), mean value (avg), worst value (worst), and standard deviation (std) of each algorithm. The results of each algorithm are shown in Table 2.

From Table 2, it can be seen that the performance of the CCCHHO algorithm proposed in this article is greatly improved compared with the performance of other swarm intelligence optimization algorithms. For the F1~F5 function, the results of the four dimensions of

CCCHHO are much higher than the other four algorithms, especially the results of F1, F4, and F5 all reach the minimum value of the function. At the same time, the stability of CCCHHO is also far better than the other four algorithms. For the six functions of F6~F11, the results of F8 and F9 are better than the other four algorithms. And the results of F7 are the same as the HHO algorithm, better than GWO, SSA, and ALO. The mean value of F6 is the same as that of HHO, but the stability is better. It is also better than GWO, SSA, and ALO. The mean values of F10 and F11 are slightly superior, and the stability is also better.

TABLE 3: The result of comparison of CCCHHO with the improved algorithm.

Function	Index	CCCHHO	THHO	MHHO	OBLHHO
F1	Best	0.00E+00	0.00E+00	1.58E-141	0.00E+00
	Avg	0.00E+00	6.48E-108	4.94E-124	7.20E-299
	Worst	0.00E+00	1.92E-106	1.01E-122	2.16E-297
	Std	0.00E+00	3.51E-107	1.92E-123	0.00E+00
F2	Best	6.30E-238	2.74E-178	1.53E-72	9.02E-222
	Avg	1.69E-180	1.01E-56	8.05E-63	2.35E-170
	Worst	5.07E-179	1.54E-55	1.95E-61	7.02E-169
	Std	0.00E+00	3.40E-56	3.57E-62	0.00E+00
F3	Best	3.04E-239	6.44E-170	9.30E-71	1.15E-212
	Avg	3.72E-209	3.41E-56	1.42E-63	1.41E-159
	Worst	1.07E-207	8.97E-55	2.95E-62	4.22E-158
	Std	0.00E+00	1.64E-55	5.43E-63	7.70E-159
F4	Best	0.00E+00	3.92E-08	9.05E-06	5.63E-06
	Avg	0.00E+00	6.65E-03	5.84E-03	1.01E-02
	Worst	0.00E+00	3.87E-02	3.96E-02	8.90E-02
	Std	0.00E+00	9.68E-03	8.69E-03	2.01E-02
F5	Best	0.00E+00	5.93E-09	1.89E-07	5.03E-07
	Avg	0.00E+00	1.64E-04	8.92E-05	1.76E-04
	Worst	0.00E+00	1.13E-03	4.92E-04	8.03E-04
	Std	0.00E+00	2.92E-04	1.06E-04	2.13E-04
F6	Best	-1.26E+04	-1.26E+04	-1.26E+04	-1.26E+04
	Avg	-1.26E+04	-1.25E+04	-1.25E+04	-1.25E+04
	Worst	-1.26E+04	-9.95E+03	-1.19E+04	-9.05E+03
	Std	1.85E-12	4.78E+02	1.23E+02	6.43E+02
F7	Best	8.88E-16	8.88E-16	8.88E-16	8.88E-16
	Avg	8.88E-16	8.88E-16	8.88E-16	8.88E-16
	Worst	8.88E-16	8.88E-16	8.88E-16	8.88E-16
	std	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F8	Best	1.57E-32	7.43E-09	1.20E-09	1.48E-08
	Avg	1.57E-32	1.11E-05	5.14E-06	6.55E-06
	Worst	1.57E-32	4.83E-05	3.97E-05	6.32E-05
	Std	5.57E-48	1.23E-05	7.67E-06	1.25E-05
F9	Best	1.35E-32	4.52E-08	9.51E-08	3.10E-07
	Avg	1.35E-32	2.40E-04	6.75E-05	7.72E-05
	Worst	1.35E-32	9.31E-04	2.60E-04	5.53E-04
	Std	5.57E-48	2.64E-04	7.84E-05	1.21E-04
F10	Best	9.98E-01	9.98E-01	9.98E-01	9.98E-01
	Avg	9.98E-01	1.49E+00	1.66E+00	1.41E+00
	Worst	9.98E-01	5.93E+00	5.93E+00	2.98E+00
	Std	2.40E-16	1.26E+00	1.50E+00	6.12E-01
F11	Best	3.08E-04	3.08E-04	3.08E-04	3.10E-04
	Avg	3.26E-04	3.66E-04	3.85E-04	3.47E-04
	Worst	3.65E-04	4.75E-04	1.68E-03	4.23E-04
	Std	1.55E-05	4.93E-05	2.48E-04	3.24E-05

4.3. *Comparison of CCCHHO and Other Improved HHO Algorithms.* This section compares CCCHHO with THHO [33], MHHO [34], and OBLHHO [35]. We record the optimal value (best), mean value (avg), worst value (worst), and standard deviation (std) of each algorithm, and the results of each algorithm are shown in Table 3.

As can be seen from Table 3, the results of the CCCHHO algorithm in 7 benchmark functions that F1~F5, F8, and F9 are higher than other improved HHO algorithms in 4 dimensions, among which F1, F4, and F5 have reached the minimum value of the function. The

mean values of the three functions F6, F10, and F11 are slightly better than other improved algorithms; the results of F7 are the same as those of HHO and the improved HHO algorithm. At the same time, the CCCHHO algorithm has better advantages in the standard deviation of all functions except F7. The results show that the CCCHHO algorithm outperforms the other improved algorithms in terms of accuracy and stability.

The convergence curves of CCCHHO, HHO, and other improved HHO algorithms in each benchmark function are shown in Figures 3 and 4.

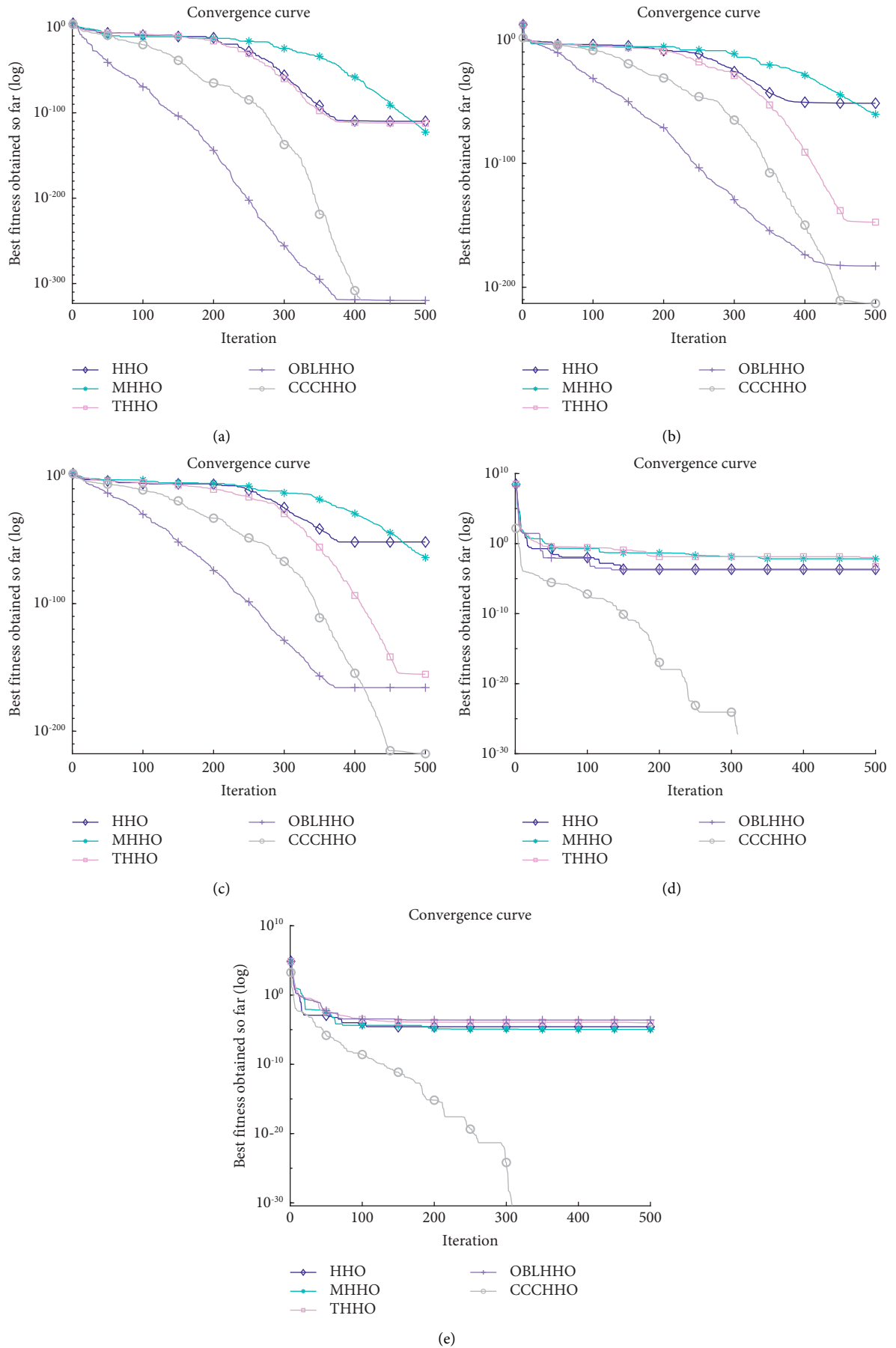


FIGURE 3: Convergence curve of unimodal function. (a) F1. (b) F2. (c) F3. (d) F4.(e) F5.

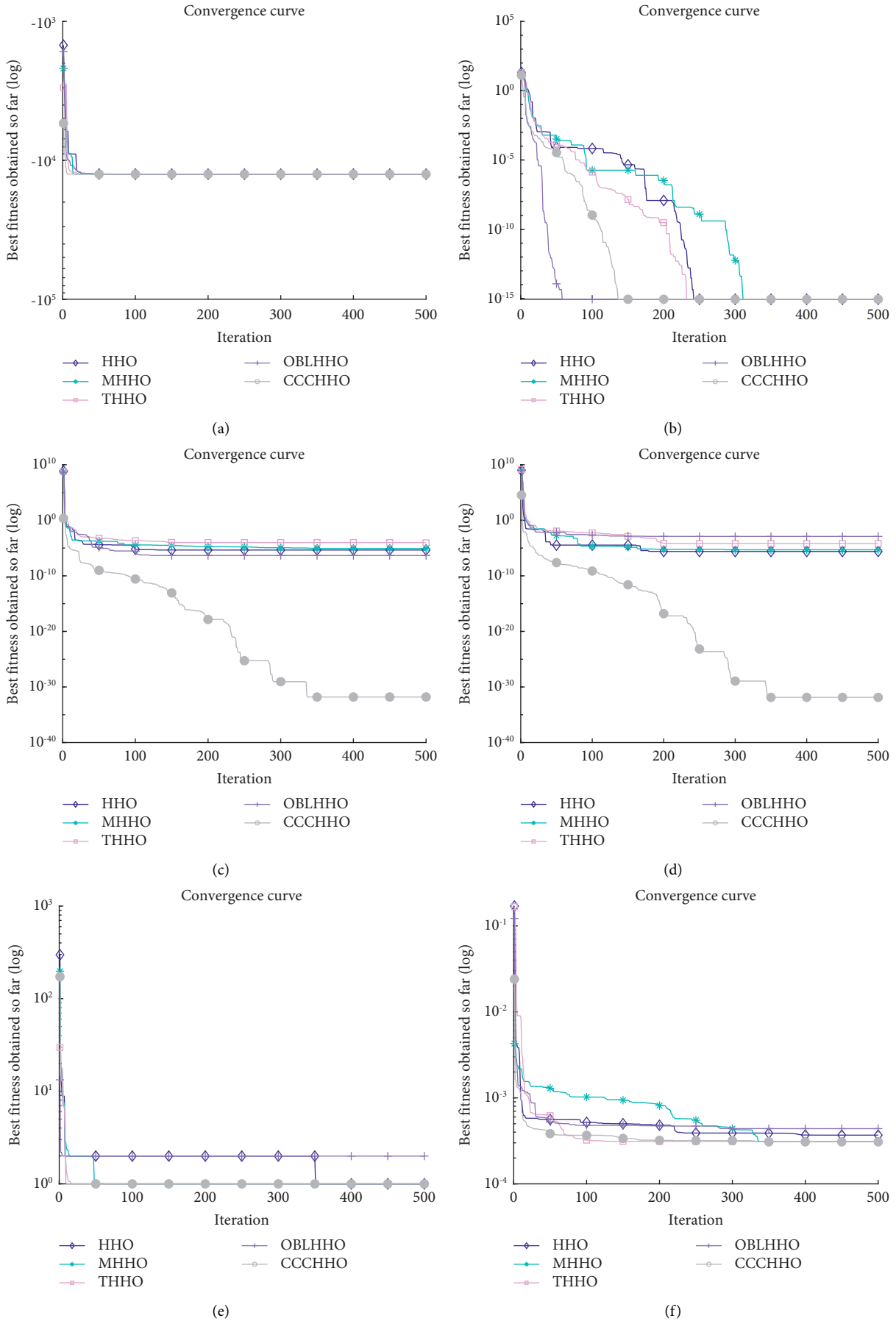


FIGURE 4: Convergence curve of multimodal function. (a) F6. (b) F7. (c) F8. (d) F9. (e) F10. (f) F11.

Figure 3 shows the convergence curves of each algorithm on the unimodal benchmark function. Among the five improved algorithms, the CCCHHO proposed in this article has the best performance in optimization. In F1~ F3, CCCHHO converges the fastest and has the highest accuracy. In F4 and F5, it can be seen that CCCHHO has fallen into the local optimum many times in the iterative process. In the early phase, it jumped out of the local optimum through the Cauchy mutation strategy, successfully jumped out of the local optimum through the chaotic local search strategy in the later phase of the iteration, and finally found a better solution.

Figure 4 shows the convergence curve of each algorithm on the multimodal benchmark function. For the F6 function, the convergence speed of CCCHHO is relatively fast, and the optimization results are basically the same. In the F7 function, the convergence speed of CCCHHO is only second to OBLHHO. For the F8 function and the F9 function, compared with the other improved HHO algorithms, they fell into local optimum very early and have not jumped out of it. CCCHHO jumped out the local optimal value many times through Cauchy mutation, chaotic local search, and elite individual guidance strategy and finally found a better solution. In F10~ F11, the final results of CCCHHO and the other improved algorithms are better, and the convergence speed of CCCHHO is relatively fast.

5. Conclusions

In this article, a new Harris hawks optimization algorithm with multistrategy (CCCHHO) is proposed by introducing chaotic method initialization population, Cauchy mutation, nonlinear escape energy factor based on cosine function, chaotic local search, and elite individual guidance mechanism to improve the optimization performance of the basic HHO algorithm. In order to verify the performance of the proposed algorithm, 11 different types of benchmark functions were tested to analyze the exploration ability, exploitation ability, and convergence behavior of CCCHHO. The experimental results show that the exploration, exploitation, and convergence speed of CCCHHO is better than that of the basic HHO, three improved HHO algorithms, and three other swarm intelligence optimization algorithms. In future work, higher-dimensional problems will be tested and evaluated, and CCCHHO will be applied to practical engineering problems, such as parameter optimization and shop scheduling.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This work was supported by the National Key R&D Program of China (2019YFB1706302).

References

- [1] L. Ma, M. Huang, S. Yang, R. Wang, and X. Wang, "An adaptive localized decision variable analysis approach to large-scale multiobjective and many-objective optimization," *IEEE Transactions on Cybernetics*, vol. 52, no. 7, pp. 6684–6696, 2022.
- [2] L. Ma, X. Wang, X. Wang, L. Wang, Y. Shi, and M. Huang, "TCDA: truthful combinatorial double auctions for mobile edge computing in industrial Internet of Things," *IEEE Transactions on Mobile Computing*, p. 1, 2021.
- [3] L. Ma, N. Li, Y. Guo et al., "Learning to optimize: reference vector reinforcement learning adaption to constrained many-objective optimization of industrial copper burdening system," *IEEE Transactions on Cybernetics*, pp. 1–14, 2021.
- [4] X. Xue, J. Chen, and X. Yao, "Efficient user involvement in semiautomatic ontology matching," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 5, no. 2, pp. 214–224, 2021.
- [5] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: a gravitational search algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.
- [6] R. A. Formato, "Central force optimization," *Progress In Electromagnetics Research*, vol. 77, no. 1, pp. 425–491, 2007.
- [7] A. Hatamlou, "Black hole: a new heuristic optimization approach for data clustering," *Information Sciences*, vol. 222, pp. 175–184, 2013.
- [8] J. H. Holland, "Genetic algorithms," *Scientific American*, vol. 267, no. 1, pp. 66–72, 1992.
- [9] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [10] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.
- [11] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.
- [12] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the ICNN'95-international conference on neural networks*, pp. 1942–1948, IEEE, Perth, WA, Australia, November 1995.
- [13] L. Ma, X. Wang, M. Huang, Z. Lin, L. Tian, and H. Chen, "Two-level master-slave RFID networks planning via hybrid multiobjective artificial bee colony optimizer," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 5, pp. 861–880, 2019.
- [14] L. Sun, S. Chen, J. Xu, and Y. Tian, "Improved monarch butterfly optimization algorithm based on opposition-based learning and random local perturbation," *Complexity*, vol. 2019, Article ID 4182148, 2019.
- [15] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
- [16] H.-S. Wu, F. Zhang, and L. Wu, "New swarm intelligence algorithm-wolf pack algorithm," *Systems Engineering and Electronics*, vol. 35, no. 11, pp. 2430–2438, 2013.

- [17] S. Mirjalili, "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Computing & Applications*, vol. 27, no. 4, pp. 1053–1073, 2016.
- [18] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.
- [19] S. Mirjalili, "The ant lion optimizer," *Advances in Engineering Software*, vol. 83, pp. 80–98, 2015.
- [20] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp Swarm Algorithm: a bio-inspired optimizer for engineering design problems," *Advances in Engineering Software*, vol. 114, pp. 163–191, 2017.
- [21] L. Ma, S. Cheng, and Y. Shi, "Enhancing learning efficiency of brain storm optimization via orthogonal learning design," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 11, pp. 6723–6742, 2021.
- [22] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019.
- [23] S. K. ElSayed and E. E. Elattar, "Hybrid Harris hawks optimization with sequential quadratic programming for optimal coordination of directional overcurrent relays incorporating distributed generation," *Alexandria Engineering Journal*, vol. 60, no. 2, pp. 2421–2433, 2021.
- [24] A. Abbasi, B. Firouzi, P. Sendur, A. A. Heidari, H. Chen, and R. Tiwari, "Multi-strategy Gaussian Harris hawks optimization for fatigue life of tapered roller bearings," *Engineering with Computers*, pp. 1–27, 2021.
- [25] H. Jouhari, D. Lei, M. A. A. Al-qaness et al., "Modified Harris Hawks optimizer for solving machine scheduling problems," *Symmetry*, vol. 12, no. 9, p. 1460, 2020.
- [26] H. Chen, S. Jiao, M. Wang, A. A. Heidari, and X. Zhao, "Parameters identification of photovoltaic cells and modules using diversification-enriched Harris hawks optimization with chaotic drifts," *Journal of Cleaner Production*, vol. 244, Article ID 118778, 2020.
- [27] H. Jia, C. Lang, D. Oliva, W. Song, and X. Peng, "Dynamic Harris hawks optimization with mutation mechanism for satellite image segmentation," *Remote Sensing*, vol. 11, no. 12, p. 1421, 2019.
- [28] C. Qu, W. He, X. Peng, and X. Peng, "Harris hawks optimization with information exchange," *Applied Mathematical Modelling*, vol. 84, pp. 52–75, 2020.
- [29] G. Kaur and S. Arora, "Chaotic whale optimization algorithm," *Journal of Computational Design and Engineering*, vol. 5, no. 3, pp. 275–284, 2018.
- [30] S. Arora and P. Anand, "Chaotic grasshopper optimization algorithm for global optimization," *Neural Computing & Applications*, vol. 31, no. 8, pp. 4385–4405, 2019.
- [31] R. Chunhui, L. Sheng, Z. Weikang, and Z. Weiwei, "Optimization and application of Cauchy mutation camel algorithm," *Computer Engineering and Applications*, vol. 57, no. 21, pp. 87–94, 2021.
- [32] Y. Tan, G.-Z. Tan, B. Yang, Z.-C. Zhao, and L. Huang, "Differential evolution algorithm with chaotic-local-search strategy for mixed-integer nonlinear programming problems," *Journal of Chinese Computer Systems*, vol. 33, no. 6, pp. 1306–1309, 2012.
- [33] M. A. Al-Betar, M. A. Awadallah, A. A. Heidari, H. Chen, H. Al-Khraisat, and C. Li, "Survival exploration strategies for Harris hawks optimizer," *Expert Systems with Applications*, vol. 168, Article ID 114243, 2021.
- [34] Y. Zhang, X. Zhou, and P.-C. Shih, "Modified Harris Hawks optimization algorithm for global optimization problems," *Arabian Journal for Science and Engineering*, vol. 45, no. 12, Article ID 10974, 2020.
- [35] O. M. Ismael, O. S. Qasim, and Z. Y. Algarni, "Improving Harris hawks optimization algorithm for hyperparameters estimation and feature selection in v-support vector regression based on opposition-based learning," *Journal of Chemometrics*, vol. 34, no. 11, Article ID e3311, 2020.