

Article

Multi-Strategy Enhanced Harris Hawks Optimization for Global Optimization and Deep Learning-Based Channel Estimation Problems

Yunshan Sun, Qian Huang, Ting Liu *, Yuetong Cheng and Yanqin Li

School of Information Engineering, Tianjin University of Commerce, Tianjin 300134, China

* Correspondence: liuting@tjcu.edu.cn

Abstract: Harris Hawks Optimization (HHO) simulates the cooperative hunting behavior of Harris hawks and it has the advantages of fewer control parameters, simple principles, and excellent exploitation ability. However, HHO also has the disadvantages of slow convergence and easy falling into local optimality. Aiming at the above shortcomings, this paper proposes a Multi-strategy Enhanced Harris Hawks Optimization (MEHHO). Firstly, the map-compass operator and Cauchy mutation strategy are used to increase the population diversity and improve the ability of the algorithm to jump out of the local optimal. Secondly, a spiral motion strategy is introduced to improve the exploration phase to enhance search efficiency. Finally, the convergence speed and accuracy of the algorithm are improved by greedy selection to fully retain the dominant individuals. The global search capability of the proposed MEHHO is verified by 28 benchmark test functions, and then the parameters of the deep learning network used for channel estimation are optimized by using the MEHHO to verify the practicability of the MEHHO. Experimental results show that the proposed MEHHO has more advantages in solving global optimization problems and improving the accuracy of the channel estimation method based on deep learning.



Citation: Sun, Y.; Huang, Q.; Liu, T.; Cheng, Y.; Li, Y. Multi-Strategy Enhanced Harris Hawks Optimization for Global Optimization and Deep Learning-Based Channel Estimation Problems. *Mathematics* **2023**, *11*, 390. <https://doi.org/10.3390/math11020390>

Academic Editors: Yuri Popkov, Alexander Nazin, Zeev Volkovich and Ioannis G. Tsoulos

Received: 13 December 2022

Revised: 2 January 2023

Accepted: 10 January 2023

Published: 11 January 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: meta-heuristic algorithm; Harris hawks optimization; global optimization; benchmark test functions; deep learning; channel estimation; long short-term memory

MSC: 49K35; 68T20

1. Introduction

A meta-heuristic algorithm is a kind of algorithm that is inspired by nature, obtains inspiration from evolutionary rules, physical rules and biological social behavior in nature, and establishes a mathematical model by combining a stochastic algorithm and a local search algorithm. Inspired by different mechanisms, a variety of algorithms have been proposed. Some classical algorithms, such as Genetic Algorithms (GA) [1] and Differential Evolution Algorithm (DE) [2], are typical algorithms based on biological evolutionary rules. Simulated Annealing Algorithms (SA) [3], Henry Gas Solubility Optimization (HGSO) [4], and Atom Search Optimization (ASO) [5] are algorithms based on physical laws. Particle Swarm Optimization (PSO) [6], Ant Colony Optimization (ACO) [7], Artificial Bee Colony Algorithm (ABC) [8], Whale Optimization Algorithm (WOA) [9], and Salp Swarm Algorithm (SSA) [10] are algorithms based on biological social behavior, also known as Swarm Intelligence Algorithms. According to the No Free Lunch theory, there is no algorithm suitable for solving all optimization problems. Therefore, researchers have proposed many novel meta-heuristic algorithms and improved classical algorithms to solve a variety of more complex practical problems. New meta-heuristic algorithms include the Butterfly Optimization Algorithm (BOA) [11], Marine Predators Algorithm (MPA) [12], Mayfly Algorithm (MA) [13], Aquila Optimizer (AO) [14], Arithmetic Optimization Algorithm (AOA) [15], Sand Cat Swarm Optimization (SCSO) [16], Weighted Mean of

Vectors (INFO) [17], Runge Kutta optimizer (RUN) [18], and so on. With its advantages of good adaptability, independent and efficient exploration mechanism, no cost of gradient information, and simple implementation, the meta-heuristic algorithm has been widely used in different fields, and provided powerful tools for all walks of life to solve daily optimization problems, such as fault diagnosis [19,20], feature selection [21,22], microchannel radiator design [23], recognition watermarking [24], medical detection [25–27], path planning [28,29], chart pattern recognition [30], and Internet of Things [31–33], and so on.

The Harris Hawks Optimization (HHO) is a meta-heuristic algorithm proposed by A. Heidari et al. [34] based on the hunting behavior of Harris hawks. HHO has introduced the concepts of escape energy, population center, Lévy flight, etc., and has a superior solving ability in reference functions and constraint engineering problems. In addition, HHO has the advantages of fewer control parameters, easy programming and implementation. Therefore, HHO has been widely used in many fields and achieved good results. For example, HHO has been successfully applied to solve the optimization problem of model parameters in the photovoltaic field. A. Ramadan et al. [35] used the improved HHO to estimate the required parameters of different photovoltaic models and constructed their models with high precision. M. Naeijian et al. [36] used the improved HHO to find the optimal parameters of the single-diode model, the double-diode model, and the triple-diode model. H. Chen et al. [37] proposed a diversified enhanced HHO to efficiently identify the parameters of photovoltaic cells. In addition, HHO has been widely used to solve the multi-stage image segmentation threshold optimization problem in the image field. For example, H. Jia et al. [38] improved HHO by introducing dynamic control parameters and mutation operators, and applied them to the optimal segmentation threshold of multi-stage satellite images. A. Wunnava et al. [39] proposed a differential evolution adaptive HHO, which was used for two-dimensional Masi entropy multi-level image threshold segmentation. E. R. Esparza et al. [40] used the minimum cross entropy as a fitness function to propose an HHO-based optimal threshold solution method for multistage segmentation. They ran tests on medical images. In addition to the above fields, HHO has also been successfully applied in the field of wireless sensor networks (WSN) to solve the optimal parameter problem. For example, M. Srinivas et al. [41] proposed an energy-saving optimization method based on improved HHO to extend the life of WSN. S. J. Bhat et al. [42] chose the area as the fitness function to reduce the search area and used their HHO-AM to solve the optimal parameters to improve the positioning accuracy of WSN.

Similar to other meta-heuristic algorithms, HHO also has certain disadvantages, such as slow convergence due to low search efficiency in the exploration stage, and lack of population diversity leading to easy falling into local optimality. Because of the shortcomings of HHO, scholars have put forward some improvements. Aiming at the problem of few exploration behaviors in the exploration stage, A. Kaveh et al. [43] improved the exploration performance of the algorithm and accelerated the convergence rate by combining the imperialist competitive algorithm and making use of its excellent space exploration ability. A. Dehkordi et al. [44] used nine chaotic maps with different mathematical equations to enhance the population diversity and improve the exploration behavior of the algorithm. S. Gupta et al. [45] balanced the exploration and development stages of the algorithm by introducing nonlinear energy parameters to update the energy factor. To improve the diversity of the population, Q. Fan et al. [46] have proposed a quasi-reflection HHO, which introduces the quasi-reflection learning mechanism to increase the diversity of the population and thus improve the convergence accuracy of the algorithm. To improve the quality of candidate solutions in the process of global search, C. Liu [47] designed the improved algorithm to update the position according to the best individuals in the population in the process of global exploration, instead of searching aimlessly. To solve the problem of falling into the local optimal, L. Abualigah et al. [48] proposed two new search methods, which used the sine function and cosine function, respectively, to improve the convergence speed and the ability to jump out of the local optimal of the original algorithm.

In view of the shortcomings of HHO in the exploration stage, such as slow convergence speed and stagnation of sub-optimal solution, this paper completed the following work: (1) Spiral motion is introduced in the exploration stage to better simulate the hunting behavior of Harris hawks and improve the efficiency of exploration stage. (2) Map-compass operator and Cauchy mutation are used to enhance the population diversity, fully search the area near the optimal solution, and enhance the ability of the algorithm to jump out of the local optimal to solve the suboptimal solution stagnation problem. (3) By greedy selection to retain better individuals, unnecessary consumption is reduced in the search process and further accelerates the speed of early convergence.

The rest of the paper is organized as follows. Section 2 introduces the basic concept, principles, and implementation process of HHO. Section 3 shows the details of the proposed improvement strategies based on the different shortcomings. Section 4 validates the performance of the proposed Multi-strategy Enhanced Harris Hawks Optimization (MEHHO) through several experiments, and presents the results and discussion. Section 5 performs parameter optimization of the deep learning network used for channel estimation using the MEHHO. Section 6 summarizes the whole paper and puts forward the prospect of future research work.

2. Harris Hawks Optimization

In this section, the proposed HHO is modeled based on the hunting behavior of Harris hawks. The HHO simulates their cooperative hunting with multiple mechanisms in two stages, the exploration stage and the exploitation stage. The parameter E denotes the escape energy of the prey, and HHO realizes the transition from the exploration phase to the exploitation phase according to E . The formula is shown in Equation (1):

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

where E_0 is the initial energy state of prey, which varies randomly between $(-1, 1)$ during each iteration. The calculation formula is $E_0 = 2 * rand - 1$, $rand$ is the random number of $(0, 1)$, T and t are the maximum and the current number of iterations.

2.1. Exploration Phase

When $|E| \geq 1$, hawks are in the exploration phase. They may perch randomly in tall trees in search of prey, or they may stalk and monitor prey with their companions. Assuming that a position choice is made between the following two strategies with equal probability:

$$X_{t+1}^{i,j} = \begin{cases} X_t^{rand,j} - r_1 |X_t^{rand,j} - 2r_2 X_t^{i,j}| & q \geq 0.5 \\ (X_t^{prey,j} - X_t^{av,j}) - r_3 [LB^j + r_4 (UB^j - LB^j)] & q < 0.5 \end{cases} \tag{2}$$

where $X_t^{i,j}$ and $X_{t+1}^{i,j}$ are the current position of the i th hawk in the j th dimension and the new position of the next iteration, respectively; $i \in [1, N_{pop}]$, $j \in [1, Dim]$, N_{pop} is the total number of hawks; Dim is the corresponding problem dimension; $X_t^{rand,j}$ and $X_t^{prey,j}$ are the position of a hawk and the position of the prey randomly selected in the j th dimension, respectively. r_1, r_2, r_3, r_4 and q are five different random numbers in the range of 0 to 1. LB^j and UB^j are the lower and upper bounds of the search space in the j th dimension, $X_t^{av,j}$ is the current average position of hawks in the j th dimension:

$$X_t^{av,j} = \frac{1}{N_{pop}} \sum_{i=1}^N X_t^{i,j} \tag{3}$$

2.2. Exploitation Phase

When $|E| < 1$, the algorithm enters the development stage. During this phase, the hawks will conduct a raid hunt on the prey that are stalked and observed during the exploration phase. However, in the process of hunting in nature, the prey will also try to escape from the hunt, so hawks will also adopt different pursuit modes for the various escape behaviors of the prey. HHO proposed four strategies to simulate this chasing and hunting behavior, each of which is described below. HHO uses the escape energy E and the escape probability r to determine which strategy to adopt.

2.2.1. Soft Besiege

In the case of $|E| \geq 0.5$ and $r \geq 0.5$, the prey has enough energy to escape in a random jump, yet hawks have surrounded the prey. At this point, the hawk chooses to use soft besiege to consume the physical strength of the prey, and then successfully hunt. The mathematical models are shown in Equations (4)–(6):

$$X_{t+1}^{i,j} = \Delta X_t^j - E \times \left| Jump X_t^{prey,j} - X_t^{i,j} \right| \tag{4}$$

$$\Delta X_t^j = X_t^{prey,j} - X_t^{i,j} \tag{5}$$

$$Jump = 2(1 - r_5) \tag{6}$$

where ΔX_t^j is the distance between the current position of the prey in the j th dimension and the current position of the i th hawk; r_5 is a random number between $(0, 1)$; and $Jump$ represents random jump intensity, which varies randomly between $(0, 2)$ in each iteration.

2.2.2. Hard Besiege

In the case of $|E| < 0.5$ and $r \geq 0.5$, the prey does not have enough energy to escape, and the hawks have surrounded the prey; then, the hawks will choose a hard besiege fast raid hunt. This behavior model is shown in Equation (7):

$$X_{t+1}^{i,j} = X_t^{prey,j} - E \times \left| \Delta X_t^j \right| \tag{7}$$

2.2.3. Soft Besiege with Progressive Rapid Dives

In the case of $|E| \geq 0.5$ and $r < 0.5$, the prey has enough energy to escape the siege and make a zigzag movement, and the hawks have not completely formed an encircling attack on the prey. At this point, the hawks choose to continue to expend prey energy and gradually establish a complete encircle. The process descriptions of this strategy are shown in Equations (8)–(11):

$$X_{t+1}^{i,j} = \begin{cases} Y_{t+1}^{i,j} & \text{if } f(Y_{t+1}^{i,j}) < f(X_t^{i,j}) \\ Z_{t+1}^{i,j} & \text{if } f(Z_{t+1}^{i,j}) < f(X_t^{i,j}) \end{cases} \tag{8}$$

$$Y_{t+1}^{i,j} = X_t^{prey,j} - E \times \left| Jump X_t^{prey,j} - X_t^{i,j} \right| \tag{9}$$

$$Z_{t+1}^{i,j} = Y_{t+1}^{i,j} + S^j \times L^j \tag{10}$$

$$L^j \sim LF(u^j, v^j, \beta^j) \tag{11}$$

where S^j is a random number; $LF(\cdot)$ is the Lévy flight function:

$$LF(u^j, v^j, \beta^j) = 0.01 \times \frac{u^j \times \sigma}{|v^j|^{\frac{1}{\beta^j}}} \tag{12}$$

$$\sigma = \left(\frac{\Gamma(1 + \beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right)^{\frac{1}{\beta}} \tag{13}$$

where u and v are random numbers between $(0, 1)$; $u \sim N(0, \delta^2)$; $v \sim N(0, 1)$; default $\beta = 1.5$.

2.2.4. Hard Besiege with Progressive Rapid Dives

In the case of $|E| < 0.5$ and $r < 0.5$, the prey does not have enough energy to escape, but hawks do not completely surround them, so they will choose this strategy to accelerate and shorten the average position distance between them and the prey to form a hard encircling circle before the raid. The models of this strategy are shown in Equations (14)–(16):

$$X_{t+1}^{i,j} = \begin{cases} Y_{t+1}^{i,j} & \text{if } f(Y_{t+1}^{i,j}) < f(X_t^{i,j}) \\ Z_{t+1}^{i,j} & \text{if } f(Z_{t+1}^{i,j}) < f(X_t^{i,j}) \end{cases} \tag{14}$$

$$Y_{t+1}^{i,j} = X_t^{prey,j} - E \times \left| \text{Jump} X_t^{prey,j} - X_t^{av,j} \right| \tag{15}$$

$$Z_{t+1}^{i,j} = Y_{t+1}^{i,j} + S^j \times L^j \tag{16}$$

3. Multi-Strategy Enhanced Harris Hawks Optimization

In view of the shortcomings of the original HHO, the following improvement strategies are proposed: First, the map-compass operator and Cauchy mutation strategy are introduced to achieve global optimal value-guidance and sufficient search of the current optimal solution neighborhood, increasing the population diversity, and improving the ability of the algorithm to jump out of the local optimal; secondly, the spiral motion strategy is adopted in the exploration stage to improve the exploration efficiency and accelerate the convergence speed in the exploration stage. Finally, through greedy selection the dominant individuals are fully retained.

3.1. Improved Strategy Based on Map-Compass Operator and Cauchy Mutation

One of the main disadvantages of HHO is the lack of diversity in solving complex optimization problems, which leads to the precocious phenomenon. Therefore, the map-compass operator is introduced to perturb the optimal individual before the algorithm enters the exploration stage, and Cauchy mutation is integrated to help the algorithm jump out of the local optimal.

This paper draws on the idea of the map-compass operator in the Pigeon Swarm Optimization algorithm [49] to perturb the global extreme value so that it can lead all hawks to fly to a new location. This strategy can increase the diversity of the hawk population and improve the probability of finding a better solution. Therefore, before the exploration stage, the position update formula is added, as shown in Equation (17):

$$X_{t+1}^{i,j} = X_t^{prey,j} \times e^{-\tau \times t} + \left| X_t^{prey,j} - X_t^{i,j} \right| \tag{17}$$

where τ is the map-compass factor; the value range is $(0, 1)$.

At the same time, Cauchy mutation is used to make HHO search the neighborhood of the current optimal solution more fully and in a more diversified way, to further improve the ability of the algorithm to jump out of the local optimal. In probability theory, the Cauchy distribution is a very common continuous distribution, and its probability function form is shown in Equation (18):

$$f(x) = \frac{1}{\pi} \left(\frac{\gamma}{\gamma^2 + (x - \eta)^2} \right), \quad \gamma > 0, -\infty < x < \infty \tag{18}$$

where η is a random real number that lies in the interval $(-\infty, \infty)$. A special case is the standard Cauchy distribution obtained when $\gamma = 1$ and $\eta = 0$, and its probability density function is shown in Equation (19):

$$f(x) = \frac{1}{\pi} \left(\frac{1}{1 + x^2} \right) \tag{19}$$

At present, the commonly used variations in meta-heuristic algorithms are the Gaussian variation and the Cauchy variation [50,51]. The comparison between the Gaussian distribution and the Cauchy distribution is shown in Figure 1. As shown in Figure 1, the graph of the Cauchy distribution has a relatively small peak value near the origin. After the mutation of the HHO, the search for the global optimal value will be increased, and the search time for the adjacent local interval will be reduced. MEHHO’s search ability for the global optimal value has been significantly improved, and the global exploration ability and local mining ability of the algorithm can be better balanced. In addition, compared with the Gaussian distribution, the two wings of the Cauchy distribution are flatter and wider, and the closer they are to the horizontal axis in the horizontal direction, the slower they fall. From the perspective of probability, the Cauchy distribution has a wider distribution range and allows a larger runout variation, which is more suitable for improving the global exploration ability of the algorithm. Therefore, the Cauchy mutation is selected in this paper to generate more diverse population individuals to search the main search space, so that HHO can quickly jump out of the local optimal value points.

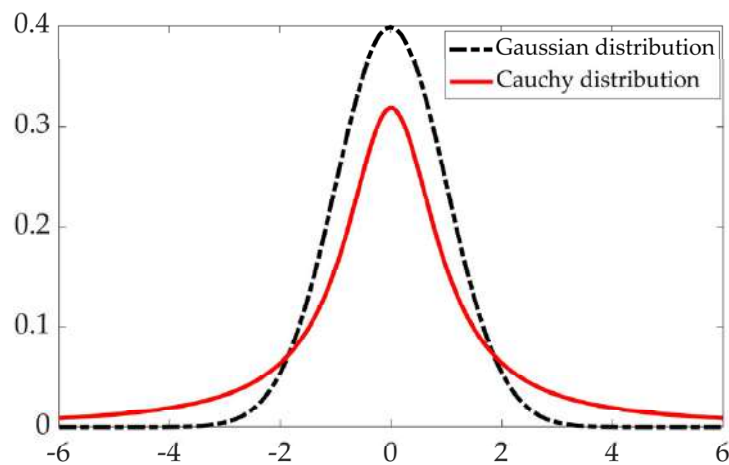


Figure 1. Comparison between Cauchy distribution and Gaussian distribution.

The position update formula of the fusion map-compass operator and the Cauchy mutation is shown in Equation (20):

$$X_{t+1}^{i,j} = X_t^{prey,j} \times e^{-\tau \times t} + Cauchy(0,1) \times |X_t^{prey,j} - X_t^{i,j}| \tag{20}$$

where $Cauchy(0,1)$ is the standard Cauchy distribution.

3.2. Position Update Mechanism Based on Spiral Motion and Greedy Strategy

Slow convergence is another drawback of HHO. In the exploration phase, individuals rely on other members of the population to update their positions, which is partly influenced by the initial population distribution. In the mathematical model when $q \geq 0.5$, the combination of random numbers and differential position vectors does not adequately improve the exploration efficiency of the algorithm. In order to improve the slow convergence

of HHO, the spiral motion strategy is introduced in the exploration stage. Spiral motion is a rotational motion around a fixed point at a constant angular velocity and gradually moving away from this point, as shown in Figure 2. Among many algorithms, the spiral strategy has been verified to be an effective strategy to improve the search ability of the algorithm [52–55], and the spiral movement is more in line with the hunting behavior of hawks in nature. The spiral motion mainly improves the mathematical model when $q \geq 0.5$, and Formula (2) is changed to Equation (21) after the introduction of the spiral motion:

$$U_{t+1}^{i,j} = \begin{cases} X_t^{rand,j} - e^{hf} \cdot \cos(2\pi f) \cdot |X_t^{rand,j} - 2r_2 X_t^{i,j}| & q \geq 0.5 \\ (X_t^{prey,j} - X_t^{av,j}) - r_3 [LB^j + r_4(UB^j - LB^j)] & q < 0.5 \end{cases} \quad (21)$$

where h is the limiting constant of the logarithmic spiral shape; f is the random number in the interval $[-1, 1]$; and $U_{t+1}^{i,j}$ is the new position of the next iteration of the i th hawk in the j th dimension.

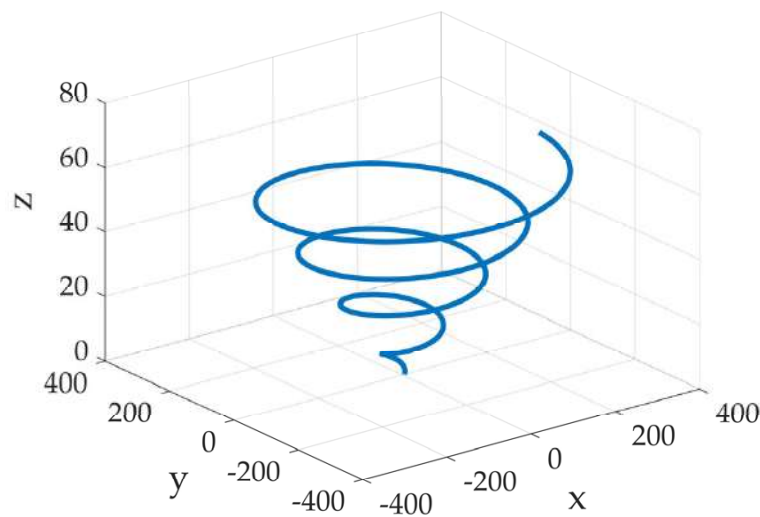


Figure 2. Schematic diagram of spiral motion.

Although the spiral motion strategy can improve the exploration efficiency of the algorithm, it cannot determine whether or not the new position obtained is better. To avoid missing the optimal solution, the greedy choice strategy is introduced in the exploration stage. The fitness values of the two positions before and after the update are compared to determine whether to update the positions in order to retain better individuals. The process is described as follows:

$$X_{t+1}^{i,j} = \begin{cases} U_{t+1}^{i,j}, & \text{if } f(U_{t+1}^{i,j}) \leq f(X_t^{i,j}) \\ X_t^{i,j}, & \text{if } f(U_{t+1}^{i,j}) > f(X_t^{i,j}) \end{cases} \quad (22)$$

where the expression of $U_{t+1}^{i,j}$ is Formula (21).

The pseudo-code of MEHHO proposed in this paper is shown in Algorithm 1.

Algorithm 1. Pseudo-code of Proposed MEHHO

Inputs: The population size N_{pop} and the maximum iterations T .

Outputs: The location of prey.

1: Initialize the random population $X^i (i = 1, 2, \dots, N_{pop})$ in a provided search space.

2: **While** $t < T$ **do**

3: Calculate the fitness values of each hawk.

4: Select the best individual position as the prey position.

5: Update the location using Equation (20) that incorporates the map-compass operator and the Cauchy mutation, calculate the individual fitness again and update X^{prey} .

6: **for** (each hawk) **do**

7: Update the initial energy E_0 and jump strength $Jump$.

8: Update the E using Equation (1).

9: **if** $|E| \geq 1$ **then**

10: Update the location of members using Equation (22).

11: **if** $|E| < 1$ **then**

12: **if** $|E| \geq 0.5, r \geq 0.5$ **then**

13: Update the location of members using Equation (4).

14: **else if** $|E| < 0.5, r \geq 0.5$ **then**

15: Update the location of members using Equation (7).

16: **else if** $|E| \geq 0.5, r < 0.5$ **then**

17: Update the location of members using Equation (8).

18: **else if** $|E| < 0.5, r < 0.5$ **then**

19: Update the location of members using Equation (14).

20: **Return** X^{prey} .

4. Experiment and Discussion: Global Optimization

To verify the performance of the MEHHO in global optimization problems, three experiments were designed: (1) MEHHO was compared with six meta-heuristic algorithms; (2) MEHHO and original HHO were compared in different dimensions; (3) MEHHO was compared with three improved HHOs. In the experiment, 28 test functions (as shown in Tables 1–3) were selected for comparative testing, including 10 unimodal benchmark functions (F1–F10), which have only one global optimal value point and are generally used to test the development ability of the algorithm. Ten multimodal benchmark functions (F11–F20) and eight fixed-dimension multimodal benchmark functions (F21–F28) were used to test the exploration ability of the algorithm and the ability to jump out of the local optimal value.

Table 1. Unimodal benchmark functions.

Function	Dim	Range	F_{min}
$F_1(x) = \sum_{i=1}^{Dim} x_i^2$	30, 50, 100, 500	$[-100, 100]$	0
$F_2(x) = \sum_{i=1}^{Dim} x_i + \prod_{i=1}^{Dim} x_i $	30, 50, 100, 500	$[-10, 10]$	0
$F_3(x) = \sum_{i=1}^{Dim} \left(\sum_{j=1}^i x_j \right)^2$	30, 50, 100, 500	$[-100, 100]$	0
$F_4(x) = \max x_i \{ x_i , 1 \leq i \leq Dim \}$	30, 50, 100, 500	$[-100, 100]$	0
$F_5(x) = \sum_{i=i}^{Dim-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$	30, 50, 100, 500	$[-30, 30]$	0
$F_6(x) = \sum_{i=i}^{Dim} ([x_i + 0.5])^2$	30, 50, 100, 500	$[-100, 100]$	0
$F_7(x) = \sum_{i=i}^{Dim} ix_i^4 + rand[0, 1)$	30, 50, 100, 500	$[-1.28, 1.28]$	0
$F_8(x) = x_1^2 + 10^6 \sum_{i=2}^{Dim} x_i^2$	30, 50, 100, 500	$[-100, 100]$	0
$F_9(x) = 10^6 x_1^2 + \sum_{i=2}^{Dim} x_i^2$	30, 50, 100, 500	$[-1, 1]$	0
$F_{10}(x) = \sum_{i=1}^{Dim} (10^6)^{\frac{i-1}{Dim-1}} x_i^2$	30, 50, 100, 500	$[-100, 100]$	0

Table 2. Multimodal benchmark functions.

Function	Dim	Range	F_{min}
$F_{11}(x) = \sum_{i=1}^{Dim} [x_i^2 - 10 \cos(2\pi x_i + 10)]$	30, 50, 100, 500	$[-5.12, 5.12]$	0
$F_{12}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{Dim} \sum_{i=1}^{Dim} x_i^2}\right) - \exp\left(\frac{1}{Dim} \sum_{i=1}^{Dim} \cos(2\pi x_i)\right) + 20 + e$	30, 50, 100, 500	$[-32, 32]$	0
$F_{13}(x) = \frac{1}{4000} \sum_{i=1}^{Dim} x_i^2 - \prod_{i=1}^{Dim} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30, 50, 100, 500	$[-600, 600]$	0
$F_{14}(x) = \frac{\pi}{Dim} \left(10 \sin^2(\pi y_1) + \sum_{i=1}^{Dim-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_{Dim} - 1)^2\right) + \sum_{i=1}^{Dim} u(x_i, 10, 100, 4)$	30, 50, 100, 500	$[-50, 50]$	0
$F_{15}(x) = 0.1 \left(\begin{matrix} \sin^2(3\pi x_1) + \\ \sum_{i=1}^{Dim} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1}) + (x_{Dim} - 1)^2 [1 + \sin^2(2\pi x_{Dim})]] \\ + \sum_{i=1}^{Dim} u(x_i, 5, 100, 4) \end{matrix} \right)$	30, 50, 100, 500	$[-50, 50]$	0
$F_{16}(x) = \sum_{i=1}^{Dim} x_i \sin(x_i) + 0.1x_i $	30, 50, 100, 500	$[-10, 10]$	0
$F_{17}(x) = 0.5 + \frac{\sin^2(\sum_{i=1}^{Dim} x_i^2) - 0.5}{1 + 0.001(\sum_{i=1}^{Dim} x_i^2)^2}$	30, 50, 100, 500	$[-100, 100]$	0
$F_{18}(x) = 1 - \cos\left(2\pi \sqrt{\sum_{i=1}^{Dim} x_i^2}\right) + 0.1 \sqrt{\sum_{i=1}^{Dim} x_i^2}$	30, 50, 100, 500	$[-100, 100]$	0
$F_{19}(x) = \sum_{i=1}^{Dim} x_i^6 \left[2 + \sin\left(\frac{1}{x_i}\right)\right]$	30, 50, 100, 500	$[-1, 1]$	0
$F_{20}(x) = \frac{\sum_{i=1}^{Dim} x_i }{e^{\sum_{i=1}^{Dim} \sin(x_i^2)}}$	30, 50, 100, 500	$[-100, 100]$	0

Table 3. Fixed-dimension multimodal benchmark functions.

Function	Dim	Range	F_{min}
$F_{21}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 + 4x_2^4$	2	$[-5, 5]$	-1.0316
$F_{22}(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right) \cos x_1 + 10$	2	$[-5, 5]$	0.398
$F_{23}(x) = \left[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1 - 14x_2 + 6x_1x_2 + 3x_2^2)\right] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	3	$[-2, 2]$	3
$F_{24}(x) = -\sum_{i=1}^4 cH_i \exp\left(-\sum_{j=1}^3 aH_{ij} (x_j - pH_{ij})^2\right)$	3	$[1, 3]$	-3.86
$F_{25}(x) = -\sum_{i=1}^4 cH_i \exp\left(-\sum_{j=1}^6 aH_{ij} (x_j - pH_{ij})^2\right)$	6	$[0, 1]$	-3.32
$F_{26}(x) = -\sum_{i=1}^5 [(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	$[0, 10]$	-10.1532
$F_{27}(x) = -\sum_{i=1}^7 [(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	$[0, 10]$	-10.4028
$F_{28}(x) = -\sum_{i=1}^{10} [(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	$[0, 10]$	-10.5363

4.1. Comparison with Other Meta-Heuristic Algorithms

To verify the effectiveness and significance of MEHHO, six meta-heuristic algorithms are selected for comparison in this section. They are Harris Hawks Optimization (HHO, 2019) [34], Whale Optimization Algorithm (WOA, 2016) [9], Marine Predators Algorithm (MPA, 2020) [12], Grey Wolf Optimizer (GWO, 2014) [56], Particle Swarm Optimization (PSO, 1995) [6], Butterfly Optimization Algorithm (BOA, 2019) [11]. To ensure the objectivity

of the results, the same parameters are set for each algorithm: the population size is all 30, the maximum number of iterations is 500, and the dimension of the test functions F1–F20 is 30. After each test function is run 30 times independently, its mean value, standard deviation, and computing time (unit: s) for each algorithm to reach their respective optimal value in Tables 4–6 are calculated. The experimental results are shown in Tables 4–6, where the bold fonts indicate better optimization results. To directly show the optimization performance of the MEHHO, the convergence curves of some benchmark test functions of the seven algorithms (F1–F9, F11–F19, F26–F28) are listed in this paper, as shown in Figures 3–5.

Table 4. Comparison of unimodal functions test results.

Function	Metric	MEHHO	HHO	WOA	MPA	GWO	PSO	BOA
F1	Mean	0	1.36×10^{-96}	8.44×10^{-72}	5.98×10^{-23}	2.94×10^{-27}	2.69×10^3	1.27×10^{-11}
	Std	0	4.95×10^{-96}	4.60×10^{-71}	7.77×10^{-23}	1.21×10^{-26}	1.47×10^3	9.19×10^{-13}
	Time	0.038	0.080	0.041	0.102	0.069	0.007	0.054
F2	Mean	0	2.67×10^{-51}	1.68×10^{-50}	2.85×10^{-13}	1.12×10^{-16}	3.20×10^1	3.96×10^{-9}
	Std	0	9.90×10^{-51}	9.05×10^{-50}	2.75×10^{-13}	7.44×10^{-17}	1.33×10^1	1.84×10^{-9}
	Time	0.041	0.088	0.062	0.084	0.072	0.011	0.070
F3	Mean	0	1.88×10^{-72}	4.22×10^4	2.41×10^{-4}	7.18×10^{-6}	8.15×10^3	1.25×10^{-11}
	Std	0	1.03×10^{-71}	1.28×10^4	5.09×10^{-4}	1.43×10^{-5}	3.74×10^3	9.15×10^{-13}
	Time	0.121	0.394	0.012	0.240	0.221	0.014	0.330
F4	Mean	0	4.21×10^{-49}	5.14×10^1	3.79×10^{-9}	8.78×10^{-7}	2.51×10^1	6.14×10^{-9}
	Std	0	2.23×10^{-48}	2.50×10^1	2.56×10^{-9}	6.16×10^{-7}	4.83	3.95×10^{-10}
	Time	0.037	0.100	0.006	0.082	0.064	0.007	0.093
F5	Mean	6.90×10^{-3}	1.34×10^{-2}	2.80×10^1	2.52×10^1	2.71×10^1	4.29×10^5	2.90×10^1
	Std	7.80×10^{-3}	1.86×10^{-2}	4.66×10^{-1}	3.35×10^{-1}	7.35×10^{-1}	4.72×10^5	2.49×10^{-2}
	Time	0.014	0.029	0.009	0.021	0.017	0.004	0.016
F6	Mean	5.28×10^{-5}	1.19×10^{-4}	3.87×10^{-1}	8.93×10^{-8}	7.79×10^{-1}	2.13×10^3	6.17
	Std	1.42×10^{-4}	1.20×10^{-4}	2.57×10^{-1}	2.65×10^{-7}	4.09×10^{-1}	9.04×10^2	5.76×10^{-1}
	Time	0.053	0.063	0.015	0.361	0.029	0.012	0.026
F7	Mean	5.45×10^{-5}	1.40×10^{-4}	3.90×10^{-3}	1.50×10^{-3}	2.10×10^{-3}	1.33	1.00×10^{-3}
	Std	5.54×10^{-5}	1.49×10^{-4}	5.50×10^{-3}	5.97×10^{-4}	9.10×10^{-4}	5.76×10^{-1}	4.54×10^{-4}
	Time	0.033	0.035	0.053	0.086	0.024	0.011	0.060
F8	Mean	0	2.60×10^{-89}	4.00×10^{-65}	3.74×10^{-17}	5.91×10^{-22}	1.96×10^9	1.55×10^{-11}
	Std	0	1.40×10^{-88}	2.19×10^{-64}	4.11×10^{-17}	5.52×10^{-22}	8.19×10^8	1.15×10^{-12}
	Time	0.069	0.134	0.083	0.251	0.137	0.016	0.094
F9	Mean	0	4.21×10^{-97}	2.29×10^{-74}	3.67×10^{-26}	1.82×10^{-31}	3.99	7.87×10^{-12}
	Std	0	1.61×10^{-96}	1.17×10^{-73}	4.46×10^{-26}	2.75×10^{-31}	1.21	1.09×10^{-12}
	Time	0.057	0.157	0.072	0.259	0.140	0.022	0.104
F10	Mean	0	3.68×10^{-94}	7.58×10^{-71}	1.98×10^{-19}	5.62×10^{-24}	3.27×10^7	1.40×10^{-11}
	Std	0	1.06×10^{-93}	2.87×10^{-70}	2.37×10^{-19}	1.34×10^{-23}	1.84×10^7	1.20×10^{-12}
	Time	0.100	0.260	0.162	0.263	0.190	0.017	0.203

Table 5. Comparison of multimodal functions test results.

Function	Metric	MEHHO	HHO	WOA	MPA	GWO	PSO	BOA
F11	Mean	0	0	1.89×10^{-15}	0	2.01	1.44×10^2	2.79×10^{-13}
	Std	0	0	1.04×10^{-14}	0	3.58	3.01×10^1	6.73×10^{-13}
	Time	0.048	0.103	0.066	0.146	0.057	0.019	0.115
F12	Mean	8.88×10^{-16}	8.88×10^{-16}	5.15×10^{-15}	1.25×10^{-12}	9.68×10^{-14}	1.10×10^1	6.04×10^{-9}
	Std	0	0	1.59×10^{-15}	5.26×10^{-13}	1.53×10^{-14}	1.40	2.28×10^{-10}
	Time	0.033	0.084	0.064	0.128	0.115	0.007	0.140
F13	Mean	0	0	0	0	4.20×10^{-3}	2.08×10^1	8.08×10^{-12}
	Std	0	0	0	0	8.90×10^{-3}	9.54	3.42×10^{-12}
	Time	0.020	0.076	0.052	0.104	0.035	0.008	0.134

Table 5. Cont.

Function	Metric	MEHHO	HHO	WOA	MPA	GWO	PSO	BOA
F14	Mean	2.99×10^{-6}	8.14×10^{-6}	2.75×10^{-2}	6.53×10^{-6}	5.12×10^{-2}	7.74×10^2	7.27×10^{-1}
	Std	5.33×10^{-6}	9.83×10^{-6}	4.88×10^{-2}	2.7×10^{-5}	3.03×10^{-2}	1.69×10^3	1.75×10^{-1}
	Time	0.044	0.135	0.017	0.417	0.021	0.237	0.024
F15	Mean	4.6×10^{-5}	9.1×10^{-5}	6.03×10^{-1}	5.60×10^{-3}	6.05×10^{-1}	1.45×10^5	2.98
	Std	6.89×10^{-5}	1.16×10^{-4}	2.85×10^{-1}	1.40×10^{-2}	2.53×10^{-1}	2.26×10^5	3.83×10^{-2}
	Time	0.075	0.093	0.042	0.725	0.055	0.090	0.038
F16	Mean	0	3.47×10^{-51}	1.04×10^{-31}	1.09×10^{-13}	4.27×10^{-4}	1.62×10^1	4.07×10^{-9}
	Std	0	1.7×10^{-50}	5.72×10^{-31}	1.33×10^{-13}	4.23×10^{-4}	4.25	1.13×10^{-9}
	Time	0.039	0.096	0.041	0.102	0.031	0.005	0.124
F17	Mean	0	0	3.90×10^{-3}	4.90×10^{-3}	5.80×10^{-3}	5.00×10^{-1}	5.00×10^{-3}
	Std	0	0	2.00×10^{-3}	2.52×10^{-16}	3.60×10^{-3}	1.46×10^{-5}	2.46×10^{-4}
	Time	0.046	0.089	0.030	0.118	0.086	0.013	0.077
F18	Mean	0	3.43×10^{-48}	1.57×10^{-1}	1.40×10^{-1}	1.90×10^{-1}	6.55	1.86×10^{-1}
	Std	0	1.87×10^{-47}	8.97×10^{-2}	4.98×10^{-2}	4.81×10^{-2}	1.37	3.29×10^{-2}
	Time	0.067	0.133	0.014	0.066	0.014	0.012	0.025
F19	Mean	0	1.1×10^{-280}	5.8×10^{-115}	7.49×10^{-64}	9.18×10^{-66}	5.30×10^{-4}	5.32×10^{-12}
	Std	0	0	3.2×10^{-114}	1.91×10^{-63}	2.91×10^{-65}	4.70×10^{-4}	2.81×10^{-12}
	Time	0.040	0.232	0.133	0.163	0.184	0.038	0.111
F20	Mean	0	3.51×10^{-12}	4.41×10^{-12}	6.47×10^{-12}	3.65×10^{-8}	2.88×10^{-7}	2.36×10^{-7}
	Std	0	5.98×10^{-15}	1.47×10^{-12}	3.16×10^{-12}	9.40×10^{-8}	4.27×10^{-7}	2.43×10^{-7}
	Time	0.116	0.043	0.013	0.070	0.021	0.029	0.071

Table 6. Comparison of fixed-dimension multimodal functions test results.

Function	Metric	MEHHO	HHO	WOA	MPA	GWO	PSO	BOA
F21	Mean	-1.03	-1.03	-1.03	-1.03	-1.03	-1.03	-3.56
	Std	4.60×10^{-7}	2.47×10^{-9}	2.87×10^{-9}	4.34×10^{-16}	8.77×10^{-9}	7.14×10^{-5}	8.35×10^{-2}
	Time	0.041	0.048	0.014	0.042	0.011	0.015	0.037
F22	Mean	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	7.97×10^{-1}
	Std	2.99×10^{-5}	4.95×10^{-6}	6.86×10^{-6}	0	2.30×10^{-7}	8.10×10^{-6}	1.01
	Time	0.015	0.019	0.008	0.020	0.006	0.008	0.006
F23	Mean	3.00	3.00	3.00	3.00	3.00	3.00	5.82
	Std	3.29×10^{-5}	4.91×10^{-7}	1.40×10^{-4}	1.66×10^{-15}	4.25×10^{-5}	1.08×10^{-4}	5.33
	Time	0.017	0.027	0.009	0.033	0.009	0.011	0.008
F24	Mean	-3.00×10^{-1}	-3.00×10^{-1}	-3.00×10^{-1}	-3.00×10^{-1}	-3.00×10^{-1}	-3.86	6.14×10^{-9}
	Std	2.26×10^{-16}	2.26×10^{-16}	2.26×10^{-16}	2.26×10^{-16}	2.26×10^{-16}	6.34×10^{-5}	3.24×10^{-4}
	Time	0.013	0.016	0.007	0.018	0.005	0.008	0.011
F25	Mean	-3.07	-3.08	-3.21	-3.32	-3.27	-3.31	2.90×10^1
	Std	1.55×10^{-1}	1.36×10^{-1}	1.03×10^{-1}	5.20×10^{-12}	7.22×10^{-2}	3.16×10^{-2}	4.50×10^{-1}
	Time	0.024	0.043	0.009	0.032	0.010	0.010	0.009
F26	Mean	-1.00×10^1	-5.05	-7.89	-1.02×10^1	-9.65	-6.85	-4.26
	Std	1.73×10^{-1}	5.64×10^{-3}	2.53	2.43×10^{-11}	1.54	2.82	3.26×10^{-1}
	Time	0.026	0.028	0.009	0.040	0.034	0.011	0.110
F27	Mean	-1.03×10^1	-5.57	-7.90	-1.04×10^1	-1.02×10^1	-7.56	-3.85
	Std	1.70×10^{-1}	1.49	3.16	2.43×10^{-11}	9.70×10^{-1}	3.16	4.23×10^{-1}
	Time	0.025	0.028	0.011	0.046	0.036	0.012	0.088
F28	Mean	-1.05×10^1	-5.61	-7.63	-1.05×10^1	-1.05×10^1	-7.57	-3.78
	Std	1.12×10^{-1}	1.49	3.25	2.93×10^{-11}	4.51×10^{-4}	3.56	6.20×10^{-1}
	Time	0.029	0.031	0.009	0.038	0.030	0.020	0.082

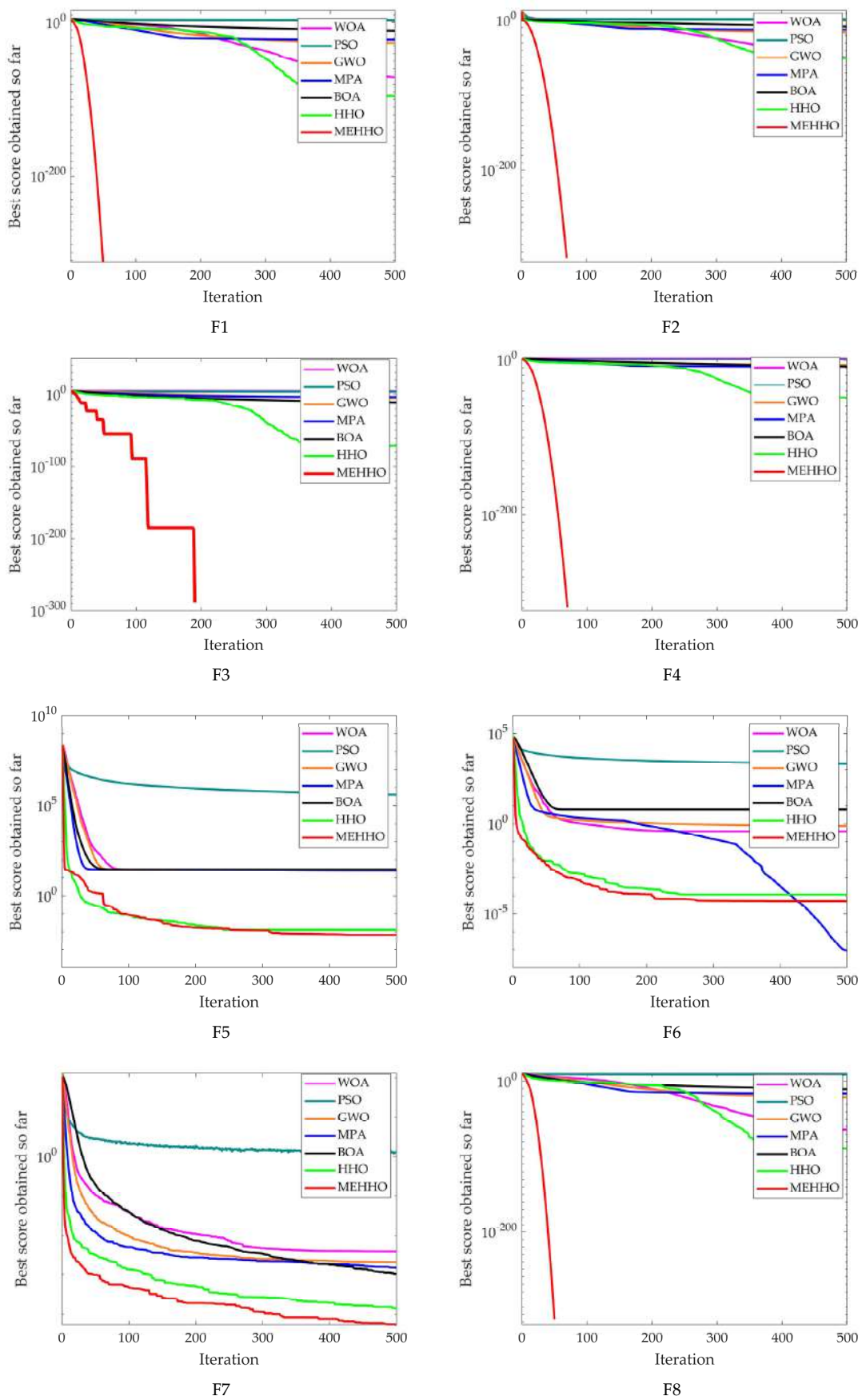


Figure 3. Cont.

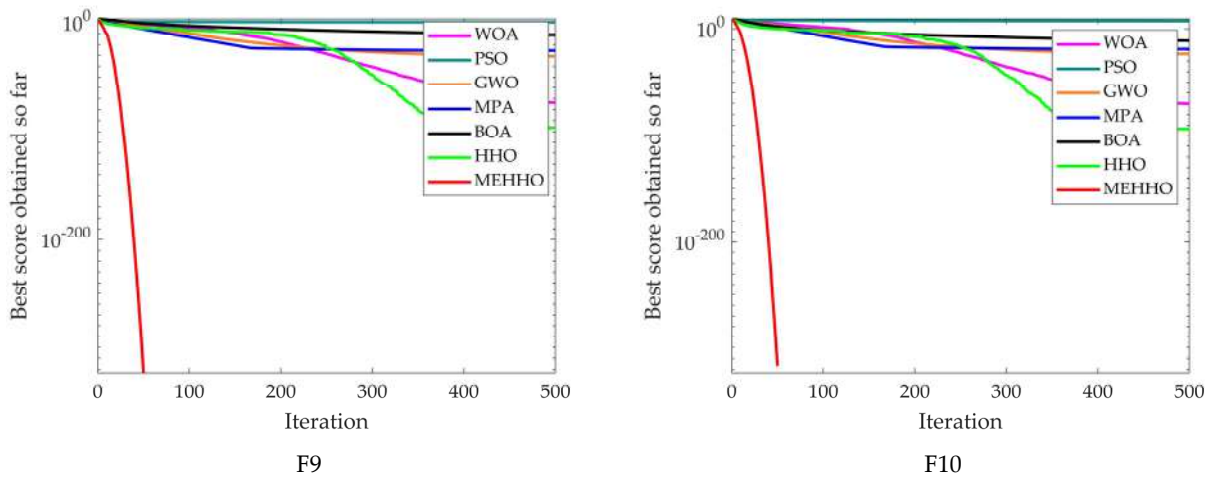


Figure 3. Convergence curves of unimodal functions (F1–F10).

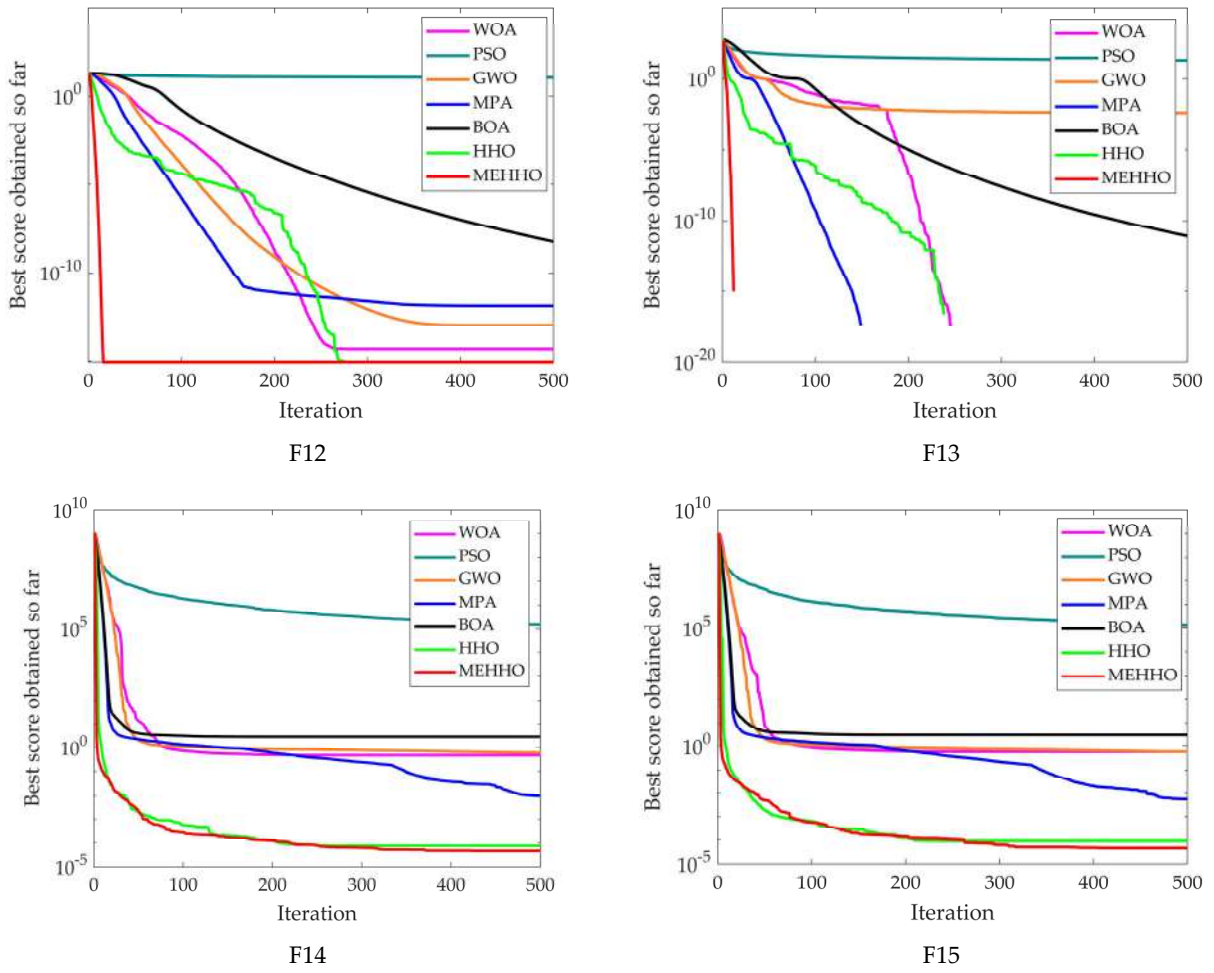


Figure 4. Cont.

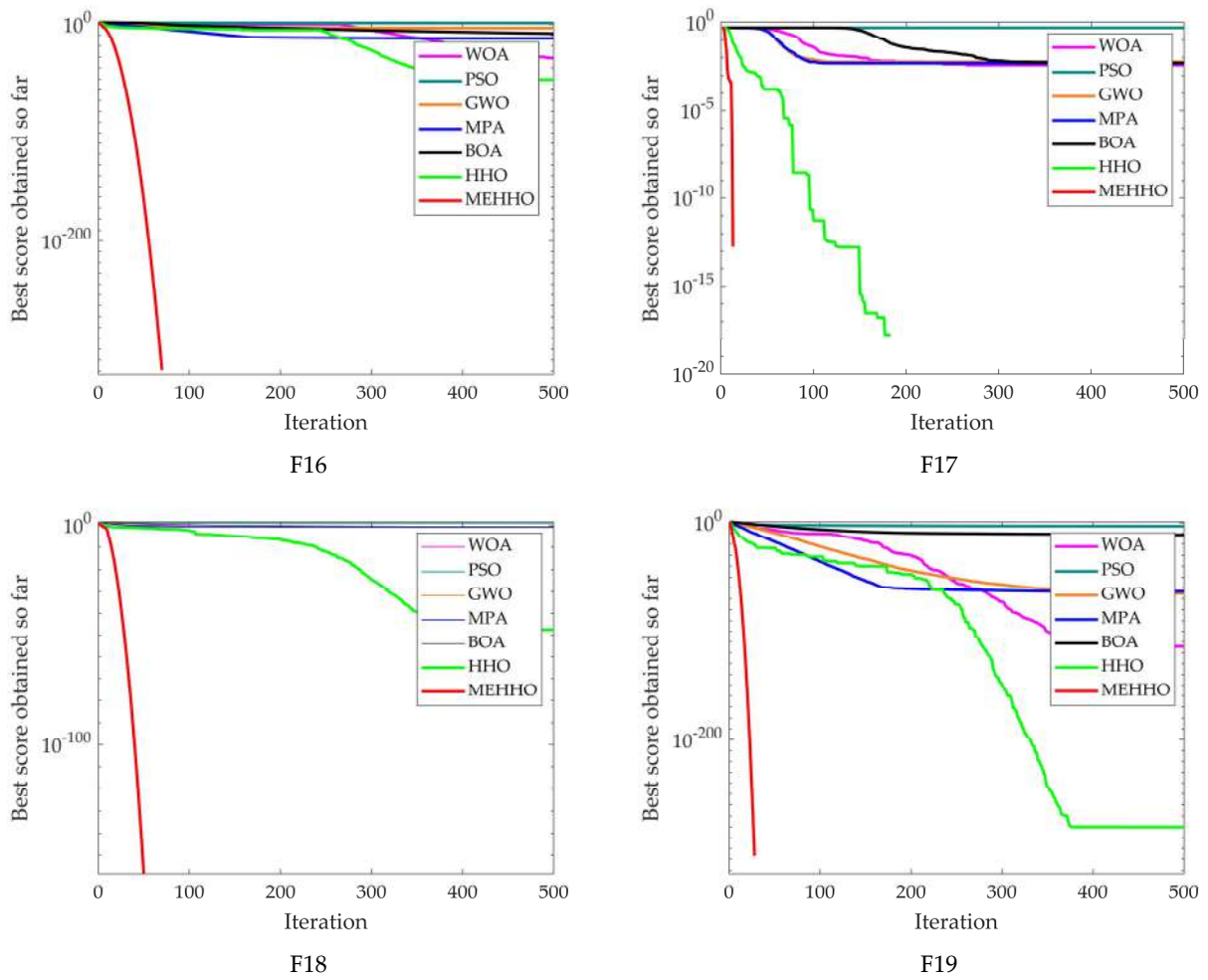


Figure 4. Convergence curve of partial multimodal functions (F12–F19).

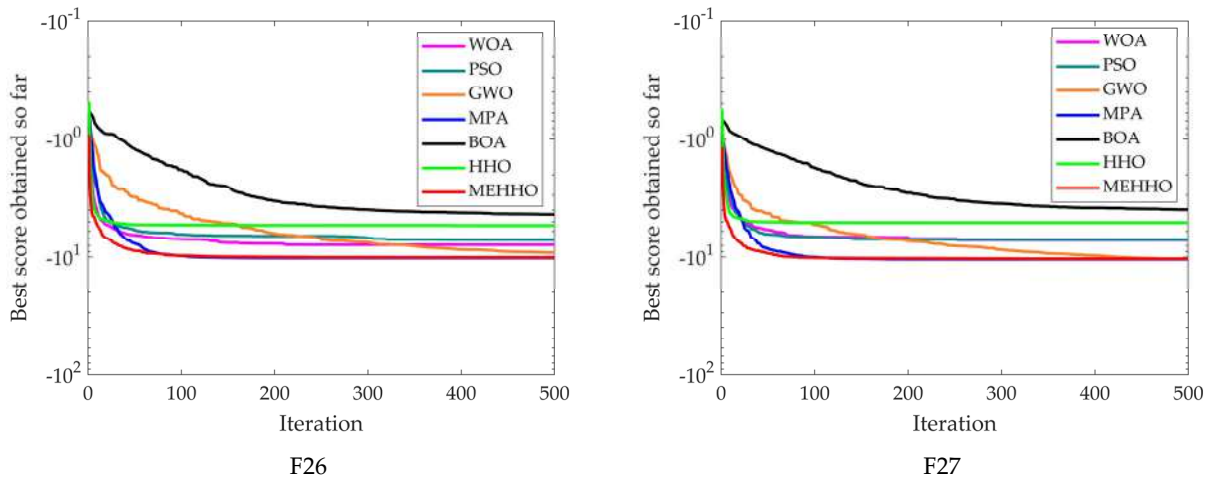


Figure 5. Convergence curve of partial fixed-dimension multimodal functions (F26, F27).

As can be seen from Table 4, for the unimodal functions, the proposed MEHHO has the strongest optimization ability, which is superior to the other six meta-heuristic algorithms. In the problem of the unimodal benchmark test functions, for functions F1–F4 and F8–F11, the optimal solution can be close to the theoretical optimal value. For functions F5 and F7, under the condition that the optimization accuracy of the other meta-heuristic algorithms is generally poor, the MEHHO cannot find the theoretical optimal value, but

its solving accuracy is the highest. For function F6, the MEHHO is second only to MPA and superior to other comparison algorithms. The standard deviations of F1–F4 and F8–F10 both reach 0, indicating that the MEHHO has high robustness and stronger stability than HHO for solving such unimodal problems. The computation time also proves the validity of MEHHO. In most test functions, the computation time of PSO is short, because the statistical time in tables is the time for algorithms to reach their respective optimal values, while PSO tends to fall into local optimal. In most unimodal functions, MEHHO takes a shorter computation time by a factor of 1 than HHO while maintaining the highest solution accuracy.

As can be seen from Table 5, in the multimodal benchmark functions (F11–F20) problem, for function F11, the proposed MEHHO, HHO, and MPA can all reach the global optimal value. For functions F12, F13, and F17, the performance of the improved algorithm is similar to that of the original HHO, and all of them can well approach the theoretical optimal value. For functions F14 and F15, the MEHHO has the highest search accuracy. For functions F16 and F18–F20, the MEHHO is superior to all the comparison algorithms, and the optimal solution is very close to the theoretical optimal value. A lower standard deviation for most problems indicates that the MEHHO is robust. In summary, compared with the other six algorithms, the proposed MEHHO has a better optimization performance and stronger stability in the test functions. For function F20, MEHHO lost some execution time in order to become closer to the theoretical optimal value. Among other multimodal test functions, the MEHHO takes less time than HHO while maintaining the highest accuracy. For functions F11 and F13, the MEHHO takes less time to execute than MPA with the same precision.

Fixed-dimension multimodal functions can be used to test the algorithm's ability to balance exploration and exploitation in the search process. In the fixed dimension optimization problem, the number of optimal solutions is not unique, but because the dimension is fixed, it requires sufficient exploration and development of the algorithm to solve this kind of problem. The results in Table 6 show that the MEHHO can reach the theoretical optimal value of the problem on functions F21–F23. In F26–F28, the MEHHO is closer to the theoretical optimal value than HHO, and is second only to MPA, indicating that the proposed MEHHO can achieve a more stable balance between exploration and development. For functions F24 and F25, the MEHHO is unable to reach the theoretical optimal value, and the standard deviation of the MEHHO is not dominant in most of the fixed-dimension multimodal functions. Further research will be conducted on how to improve the search accuracy and robustness of the algorithm on such functions. MEHHO takes less time to compute than HHO. Compared with other algorithms, the MEHHO needs further research in relation to computing time.

In the convergence diagram shown in Figures 3–5, the horizontal axis represents the number of iterations and the vertical axis represents the best fitness value. As can be seen from Figures 3–5, the convergence curve of the MEHHO decreases faster than the other six comparison algorithms, and the convergence accuracy can reach a higher level faster. As shown in the convergence curve of F1–F10 in Figure 3, the convergence speed and accuracy of the improved MEHHO are greatly improved except for functions F5 and F6, and there is an inflection point in the convergence process of the algorithm, indicating that the ability to jump out of the local extreme value is better than other comparison algorithms. It can be seen from Figure 4 that for functions F12, F13, and F17, HHO and MEHHO have a similar optimization accuracy and are better than other algorithms. However, it can be seen from the convergence curve that the MEHHO has a faster convergence speed than the original algorithm. In addition to the relatively small improvement in the optimization accuracy of functions F14 and F15, the convergence performance of the proposed MEHHO for functions F16, F18, and F19 has been significantly improved. As can be seen from Figure 5, there is little performance gap between the several algorithms in fixed-dimension multimodal functions, but the MEHHO still achieves better results. The convergence rate of F26 and F27 is faster than that of the original HHO, and it is closer to the global optimal

value of the target function. In conclusion, the convergence speed and precision of the MEHHO are superior.

4.2. Comparison and Significance Verification with Original Harris Hawks Optimization in Different Dimensions

To further verify the optimization ability of the proposed MEHHO in different dimensions (50/100/500), the performance of the proposed MEHHO and the original HHO is tested in different dimensions. Set the population size of both algorithms as 30, the maximum number of iterations as 500, and calculate the mean, standard deviation, maximum and minimum values of each test function after running it 30 times independently. The results are shown in Tables 7–9, where the bold fonts indicate better optimization results.

Table 7. Results comparison of variable-dimensional functions (50 dimensions).

Function	Optimizer	Mean	Std	Best	Worst
F1	HHO	5.38×10^{-95}	2.28×10^{-94}	2.8×10^{-112}	1.24×10^{-93}
	MEHHO	0	0	0	0
F2	HHO	9.2×10^{-49}	4.87×10^{-48}	3.32×10^{-61}	2.67×10^{-47}
	MEHHO	0	0	0	0
F3	HHO	5.07×10^{-63}	2.21×10^{-62}	2.35×10^{-92}	1.17×10^{-61}
	MEHHO	0	0	0	0
F4	HHO	1.43×10^{-48}	5.9×10^{-48}	9.29×10^{-59}	3.2×10^{-47}
	MEHHO	0	0	0	0
F5	HHO	1.89×10^{-2}	2.95×10^{-2}	5.48×10^{-5}	1.22×10^{-1}
	MEHHO	1.80×10^{-2}	2.10×10^{-2}	2.89×10^{-6}	9.28×10^{-2}
F6	HHO	2.85×10^{-4}	4.02×10^{-4}	6.20×10^{-8}	1.70×10^{-3}
	MEHHO	1.32×10^{-4}	2.12×10^{-4}	1.69×10^{-8}	8.55×10^{-4}
F7	HHO	1.62×10^{-4}	1.99×10^{-4}	1.10×10^{-5}	1.10×10^{-3}
	MEHHO	5.54×10^{-5}	5.47×10^{-5}	2.32×10^{-6}	2.61×10^{-4}
F8	HHO	2.61×10^{-4}	1.51×10^{-86}	1.54×10^{-105}	8.27×10^{-86}
	MEHHO	0	0	0	0
F9	HHO	7.00×10^{-98}	3.41×10^{-97}	3.36×10^{-114}	1.87×10^{-96}
	MEHHO	0	0	0	0
F10	HHO	9.4×10^{-88}	5.15×10^{-87}	1.9×10^{-110}	2.82×10^{-86}
	MEHHO	0	0	0	0
F11	HHO	0	0	0	0
	MEHHO	0	0	0	0
F12	HHO	8.88×10^{-16}	0	8.88×10^{-16}	8.88×10^{-16}
	MEHHO	8.88×10^{-16}	0	8.88×10^{-16}	8.88×10^{-16}
F13	HHO	0	0	0	0
	MEHHO	0	0	0	0
F14	HHO	5.75×10^{-6}	8.57×10^{-6}	5.02×10^{-8}	3.42×10^{-5}
	MEHHO	1.80×10^{-6}	2.31×10^{-6}	8.82×10^{-11}	8.65×10^{-6}
F15	HHO	1.06×10^{-4}	1.32×10^{-4}	2.71×10^{-7}	4.74×10^{-4}
	MEHHO	6.03×10^{-5}	7.24×10^{-5}	1.51×10^{-9}	2.39×10^{-4}
F16	HHO	6.23×10^{-50}	3.24×10^{-49}	3×10^{-62}	1.78×10^{-48}
	MEHHO	0	0	0	0
F17	HHO	0	0	0	0
	MEHHO	0	0	0	0

Table 7. Cont.

Function	Optimizer	Mean	Std	Best	Worst
F18	HHO	3.28×10^{-48}	1.67×10^{-47}	5.39×10^{-55}	9.14×10^{-47}
	MEHHO	0	0	0	0
F19	HHO	1×10^{-293}	0	0	3×10^{-292}
	MEHHO	0	0	0	0
F20	HHO	1.21×10^{-20}	9.22×10^{-24}	1.21×10^{-20}	1.21×10^{-20}
	MEHHO	0	0	0	0

Table 8. Results comparison of variable-dimensional functions (100 dimensions).

Function	Optimizer	Mean	Std	Best	Worst
F1	HHO	5.65×10^{-95}	3.07×10^{-94}	2.1×10^{-110}	1.68×10^{-93}
	MEHHO	0	0	0	0
F2	HHO	2.45×10^{-50}	1.11×10^{-49}	7.88×10^{-60}	6.09×10^{-49}
	MEHHO	0	0	0	0
F3	HHO	2.62×10^{-59}	9.96×10^{-59}	2.79×10^{-92}	4.11×10^{-58}
	MEHHO	0	0	0	0
F4	HHO	7.75×10^{-48}	4.16×10^{-47}	2.27×10^{-55}	2.28×10^{-46}
	MEHHO	0	0	0	0
F5	HHO	4.21×10^{-2}	7.05×10^{-2}	1.65×10^{-4}	3.50×10^{-1}
	MEHHO	2.37 $\times 10^{-2}$	2.94 $\times 10^{-2}$	6.65 $\times 10^{-5}$	1.00 $\times 10^{-1}$
F6	HHO	8.52×10^{-4}	1.30×10^{-3}	1.66×10^{-8}	4.70×10^{-3}
	MEHHO	1.82 $\times 10^{-4}$	2.52 $\times 10^{-4}$	5.89 $\times 10^{-7}$	1.20 $\times 10^{-3}$
F7	HHO	2.04×10^{-4}	2.16×10^{-4}	5.36×10^{-7}	8.43×10^{-4}
	MEHHO	5.88 $\times 10^{-5}$	6.43 $\times 10^{-5}$	3.56 $\times 10^{-7}$	2.22 $\times 10^{-4}$
F8	HHO	3.63×10^{-87}	1.98×10^{-86}	2.96×10^{-106}	1.08×10^{-85}
	MEHHO	0	0	0	0
F9	HHO	8.06×10^{-96}	4.17×10^{-95}	4.03×10^{-113}	2.29×10^{-94}
	MEHHO	0	0	0	0
F10	HHO	3.12×10^{-90}	1.71×10^{-89}	1.9×10^{-109}	9.35×10^{-89}
	MEHHO	0	0	0	0
F11	HHO	0	0	0	0
	MEHHO	0	0	0	0
F12	HHO	8.88 $\times 10^{-16}$	0	8.88 $\times 10^{-16}$	8.88 $\times 10^{-16}$
	MEHHO	8.88 $\times 10^{-16}$	0	8.88 $\times 10^{-16}$	8.88 $\times 10^{-16}$
F13	HHO	0	0	0	0
	MEHHO	0	0	0	0
F14	HHO	3.13×10^{-6}	4.40×10^{-6}	7.29×10^{-9}	1.59×10^{-5}
	MEHHO	1.57 $\times 10^{-6}$	2.31 $\times 10^{-6}$	2.58 $\times 10^{-11}$	1.11 $\times 10^{-5}$
F15	HHO	2.79×10^{-4}	5.00×10^{-4}	6.34×10^{-7}	2.10×10^{-3}
	MEHHO	5.83 $\times 10^{-5}$	7.05 $\times 10^{-5}$	1.14 $\times 10^{-9}$	2.87 $\times 10^{-4}$
F16	HHO	2.55×10^{-6}	1.4×10^{-5}	9.63×10^{-60}	7.65×10^{-5}
	MEHHO	0	0	0	0
F17	HHO	0	0	0	0
	MEHHO	0	0	0	0
F18	HHO	1.5×10^{-49}	6.76×10^{-49}	1.32×10^{-55}	3.68×10^{-48}
	MEHHO	0	0	0	0

Table 8. Cont.

Function	Optimizer	Mean	Std	Best	Worst
F19	HHO	2.8×10^{-298}	0	0	8.4×10^{-297}
	MEHHO	0	0	0	0
F20	HHO	4.68×10^{-42}	6.21×10^{-44}	4.66×10^{-42}	5.00×10^{-42}
	MEHHO	4.66×10^{-43}	1.42×10^{-42}	0	4.66×10^{-42}

Table 9. Results comparison of variable-dimensional functions (500 dimensions).

Function	Optimizer	Mean	Std	Best	Worst
F1	HHO	1.97×10^{-96}	6.88×10^{-96}	5.9×10^{-114}	3.63×10^{-95}
	MEHHO	0	0	0	0
F2	HHO	1.59×10^{-49}	3.83×10^{-49}	8.98×10^{-59}	1.69×10^{-48}
	MEHHO	0	0	0	0
F3	HHO	1.09×10^{-37}	5.96×10^{-37}	1.26×10^{-74}	3.27×10^{-36}
	MEHHO	0	0	0	0
F4	HHO	2.56×10^{-47}	8.68×10^{-47}	4.67×10^{-56}	4.04×10^{-46}
	MEHHO	0	0	0	0
F5	HHO	2.31×10^{-1}	3.76×10^{-1}	2.00×10^{-3}	1.80
	MEHHO	1.53×10^{-1}	1.73×10^{-1}	1.35×10^{-4}	6.49×10^{-1}
F6	HHO	3.60×10^{-3}	6.00×10^{-3}	1.22×10^{-5}	2.29×10^{-2}
	MEHHO	1.40×10^{-3}	2.10×10^{-3}	3.00×10^{-7}	7.60×10^{-3}
F7	HHO	2.08×10^{-4}	2.96×10^{-4}	1.59×10^{-5}	1.30×10^{-3}
	MEHHO	5.58×10^{-5}	5.40×10^{-5}	6.12×10^{-7}	2.33×10^{-4}
F8	HHO	1.04×10^{-89}	2.66×10^{-89}	1.05×10^{-104}	1.03×10^{-88}
	MEHHO	0	0	0	0
F9	HHO	5.01×10^{-99}	2.06×10^{-98}	3.54×10^{-110}	1.13×10^{-97}
	MEHHO	0	0	0	0
F10	HHO	1.62×10^{-88}	8.84×10^{-88}	4.1×10^{-116}	4.84×10^{-87}
	MEHHO	0	0	0	0
F11	HHO	0	0	0	0
	MEHHO	0	0	0	0
F12	HHO	8.88×10^{-16}	0	8.88×10^{-16}	8.88×10^{-16}
	MEHHO	8.88×10^{-16}	0	8.88×10^{-16}	8.88×10^{-16}
F13	HHO	0	0	0	0
	MEHHO	0	0	0	0
F14	HHO	2.13×10^{-6}	3.37×10^{-6}	4.33×10^{-8}	1.24×10^{-5}
	MEHHO	7.06×10^{-7}	1.71×10^{-6}	6.05×10^{-10}	9.20×10^{-6}
F15	HHO	6.12×10^{-4}	7.75×10^{-4}	3.00×10^{-6}	2.90×10^{-3}
	MEHHO	2.64×10^{-4}	3.69×10^{-4}	7.17×10^{-7}	1.40×10^{-3}
F16	HHO	3.51×10^{-5}	1.92×10^{-4}	3.64×10^{-57}	1.10×10^{-3}
	MEHHO	0	0	0	0
F17	HHO	0	0	0	0
	MEHHO	0	0	0	0
F18	HHO	9.02×10^{-48}	4.45×10^{-47}	7.89×10^{-57}	2.44×10^{-46}
	MEHHO	0	0	0	0
F19	HHO	1.3×10^{-288}	0	0	3.8×10^{-287}
	MEHHO	0	0	0	0
F20	HHO	4.51×10^{-215}	0	4.46×10^{-215}	4.72×10^{-215}
	MEHHO	4.47×10^{-215}	0	4.46×10^{-215}	4.50×10^{-215}

The results in Tables 7–9 show that for the unimodal benchmark test functions, in the functions F1–F4 and F8–F10, the proposed MEHHO can be very close to the theoretical optimal value in 500 iterations of 50, 100, and 500 dimensions, and the standard deviation value is 0, which indicates that the proposed MEHHO has strong robustness. Although the MEHHO does not reach the theoretical optimal value in functions F5–F7, its search accuracy is better than the original HHO in all dimensions. For the multimodal benchmark test functions, in the functions F11, F12, F13, and F17, the proposed MEHHO and the original HHO can reach the theoretical optimal value as far as possible. In functions F16, F18, and F19, the proposed MEHHO is superior to the original HHO and can be closer to the global optimal value. Although the MEHHO is superior to the original algorithm in F14, F15, and F20, it cannot reach the theoretical optimal value. From the perspective of the overall analysis, compared with the original HHO, the MEHHO has excellent optimization ability and can provide a better solution for the objective function. Moreover, the dimensional test results verify that the proposed MEHHO has stronger competitiveness in higher dimensions.

Due to the contingency in the operation process of the algorithm, although it may have a good performance on the mean value, it is impossible to accurately evaluate the results of each operation. Therefore, to judge the significance of the results of the improved algorithm, Wilcoxon signed-rank test is performed in this section with a significance level of 0.05. Tables 10 and 11 list the test results of the MEHHO and the original HHO under different dimensions of unimodal and multimodal benchmark functions, and Table 12 lists the test results under the fixed-dimension multimodal benchmark functions. When the test *p*-value is less than 0.05, it is considered that the comparison algorithm has a significant difference; otherwise, it is considered that the comparison algorithm has little difference in performance. The +/–/= in the conclusion indicate that the MEHHO is better than/inferior to/equal to the original algorithm, respectively. According to the statistical results in Tables 10–12, for 28 test functions, the MEHHO has significant differences compared with traditional HHO, indicating that the MEHHO has a better optimization performance and a stable improvement of the optimization ability.

Table 10. Statistical results of Wilcoxon signed-rank test on unimodal functions.

Function	Value	Dim=30	Dim=50	Dim=100	Dim=500
F1	<i>p</i> -value	2.00×10^{-6}	2.00×10^{-6}	2.00×10^{-6}	2.00×10^{-6}
	conclusion	+	+	+	+
F2	<i>p</i> -value	2.00×10^{-6}	2.00×10^{-6}	2.00×10^{-6}	2.00×10^{-6}
	conclusion	+	+	+	+
F3	<i>p</i> -value	2.00×10^{-6}	2.00×10^{-6}	2.00×10^{-6}	2.00×10^{-6}
	conclusion	+	+	+	+
F4	<i>p</i> -value	2.00×10^{-6}	2.00×10^{-6}	2.00×10^{-6}	2.00×10^{-6}
	conclusion	+	+	+	+
F5	<i>p</i> -value	0.405	0.453	0.066	0.781
	conclusion	-	-	-	-
F6	<i>p</i> -value	0.004	0.01	0.014	0.465
	conclusion	+	+	+	-
F7	<i>p</i> -value	0.005	0.004	0.004	1.60×10^{-4}
	conclusion	+	+	+	+
F8	<i>p</i> -value	2.00×10^{-6}	2.00×10^{-6}	2.00×10^{-6}	2.00×10^{-6}
	conclusion	+	+	+	+
F9	<i>p</i> -value	2.00×10^{-6}	2.00×10^{-6}	2.00×10^{-6}	2.00×10^{-6}
	conclusion	+	+	+	+
F10	<i>p</i> -value	2.00×10^{-6}	2.00×10^{-6}	2.00×10^{-6}	2.00×10^{-6}
	conclusion	+	+	+	+

Table 11. Statistical results of Wilcoxon signed-rank test on multimodal functions.

Function	Value	Dim=30	Dim=50	Dim=100	Dim=500
F11	<i>p</i> -value conclusion	NA =	NA =	NA =	NA =
F12	<i>p</i> -value conclusion	NA =	NA =	NA =	NA =
F13	<i>p</i> -value conclusion	NA =	NA =	NA =	NA =
F14	<i>p</i> -value conclusion	0.002 +	5.30×10^{-5} +	0.329 -	0.141 -
F15	<i>p</i> -value conclusion	0.766 -	1.20×10^{-5} +	0.237 -	0.006 +
F16	<i>p</i> -value conclusion	2.00×10^{-6} +	2.00×10^{-6} +	2.00×10^{-6} +	2.00×10^{-6} +
F17	<i>p</i> -value conclusion	NA =	NA =	NA =	NA =
F18	<i>p</i> -value conclusion	2.00×10^{-6} +	2.00×10^{-6} +	2.00×10^{-6} +	2.00×10^{-6} +
F19	<i>p</i> -value conclusion	8.70×10^{-5} +	1.32×10^{-4} +	1.32×10^{-4} +	2.92×10^{-4} +
F20	<i>p</i> -value conclusion	2.00×10^{-6} +	2.00×10^{-6} +	2.00×10^{-6} +	4.90×10^{-5} +

Table 12. Statistical results of Wilcoxon signed-rank test on fixed-dimension multimodal functions.

Function	F21	F22	F23	F24	F25	F26	F27	F28
<i>p</i> -value	2.00×10^{-6}	0.001	5.00×10^{-6}	NA	0.079	2.00×10^{-6}	2.00×10^{-6}	2.00×10^{-6}
conclusion	+	+	+	=	-	+	+	+

4.3. Comparison with Other Improved Harris Hawks Optimization

To further verify the effectiveness and significance of the MEHHO, three improved HHO algorithms are selected in this section for comparison. They are HHO combined with Particle Swarm Optimization (hHHO-PSO, 2021) [57], HHO combined with Grey Wolf Optimizer (hHHO-GWO, 2021) [58], and HHO combined with Sine-Cosine Algorithm (hHHO-SCA, 2020) [59]. To ensure the objectivity of the results, the same parameters are set for each algorithm: the population size is all 30, the maximum number of iterations is 500, and the dimension of the unimodal functions and multimodal functions is 30. After each test function is run 30 times independently, its mean value and standard deviation are calculated. The data of the three improved algorithms are all from the original articles. Because some selected test functions are different, only the comparison results of common functions are listed here, as shown in Table 13, where the bold fonts indicate better optimization results.

As can be seen from Table 13, for the functions F1–F4, the MEHHO proposed in this paper is significantly better than the other three improved HHOS. MEHHO can converge all to the theoretical optimal value of 0 in the first 500 iterations, and all the standard deviations reach 0, which indicates that the algorithm has excellent optimization accuracy and robustness. For the functions F5–F7, F14, and F15 with difficult convergence, the MEHHO can achieve the best convergence accuracy although it cannot find the theoretical optimal value, while the other three improved algorithms generally have poor optimization accuracy. For functions F21–F23, all four algorithms can reach the theoretical optimal value. For functions F26–F28, the MEHHO is closer to the theoretical optimal value than the other three improved algorithms. In conclusion, compared with the above three improved

algorithms, the proposed MEHHO has the best convergence accuracy, indicating that the MEHHO effectively improves the performance of HHO.

Table 13. Comparison of optimization results of four improved algorithms on common functions.

Function	Metric	MEHHO	hHHO-PSO	hHHO-GWO	hHHO-SCA
F1	Mean	0	7.62×10^{-98}	4.61×10^{-96}	1.86×10^{-91}
	Std	0	3.11×10^{-97}	2.50×10^{-95}	9.49×10^{-91}
F2	Mean	0	2.49×10^{-51}	1.55×10^{-48}	2.46×10^{-51}
	Std	0	1.20×10^{-50}	8.42×10^{-48}	1.11×10^{-50}
F3	Mean	0	7.38×10^{-75}	1.53×10^{-68}	8.88×10^{-72}
	Std	0	4.02×10^{-74}	8.36×10^{-68}	4.86×10^{-71}
F4	Mean	0	1.23×10^{-47}	3.30×10^{-49}	8.02×10^{-49}
	Std	0	6.73×10^{-47}	1.21×10^{-48}	2.83×10^{-48}
F5	Mean	6.90×10^{-3}	7.32×10^{-3}	1.70×10^{-2}	1.43×10^{-2}
	Std	7.80×10^{-3}	8.53×10^{-3}	2.16×10^{-2}	2.02×10^{-2}
F6	Mean	5.28×10^{-5}	1.44×10^{-4}	1.64×10^{-4}	2.24×10^{-4}
	Std	1.42×10^{-4}	2.5×10^{-4}	3.08×10^{-4}	3.38×10^{-4}
F7	Mean	5.45×10^{-5}	1.77×10^{-4}	1.49×10^{-4}	1.22×10^{-4}
	Std	5.54×10^{-5}	1.74×10^{-4}	1.15×10^{-4}	1.10×10^{-4}
F11	Mean	0	0	0	0
	Std	0	0	0	0
F12	Mean	8.88×10^{-16}	8.88×10^{-16}	8.88×10^{-16}	8.88×10^{-16}
	Std	0	0	0	0
F13	Mean	0	0	0	0
	Std	0	0	0	0
F14	Mean	2.99×10^{-6}	1.13×10^{-5}	1.13×10^{-5}	1.13×10^{-5}
	Std	5.33×10^{-6}	1.50×10^{-5}	1.5×10^{-5}	1.5×10^{-5}
F15	Mean	4.6×10^{-5}	1.13×10^{-4}	1.13×10^{-4}	1.13×10^{-4}
	Std	6.89×10^{-5}	1.66×10^{-4}	1.66×10^{-4}	1.66×10^{-4}
F21	Mean	-1.03	-1.03	-1.03	-1.03
	Std	4.60×10^{-7}	4.80×10^{-9}	3.97×10^{-10}	1.80×10^{-9}
F22	Mean	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}	3.98×10^{-1}
	Std	2.99×10^{-5}	3.49×10^{-5}	1.86×10^{-5}	2.15×10^{-5}
F23	Mean	3.00	3.00	3.00	3.00
	Std	3.29×10^{-5}	5.28×10^{-7}	2.65×10^{-7}	9.98×10^{-7}
F24	Mean	-3.00×10^{-1}	-3.86	-3.86	-3.86
	Std	2.26×10^{-16}	3.41×10^{-3}	4.30×10^{-3}	3.00×10^{-3}
F25	Mean	-3.07	-3.10	-3.11	-3.09
	Std	1.55×10^{-1}	1.03×10^{-1}	1.21×10^{-1}	1.09×10^{-1}
F26	Mean	-1.00×10^1	-5.05	-5.22	-5.21
	Std	1.73×10^{-1}	6.60×10^{-3}	8.97×10^{-1}	8.95×10^{-1}
F27	Mean	-1.03×10^1	-5.08	-5.14	-5.25
	Std	1.70×10^{-1}	4.12×10^{-3}	1.10	9.26×10^{-1}
F28	Mean	-1.05×10^1	-5.12	-5.30	-5.28
	Std	1.12×10^{-1}	5.77×10^{-3}	9.55×10^{-1}	8.65×10^{-1}

5. Application in Channel Estimation

Channel estimation and signal detection are key tasks in the field of wireless communication. In the real world, the signal will be affected by Doppler shift, phase noise, and fading phenomenon in the process of high-speed transmission, and the received signal

will be polluted. The purpose of channel estimation and signal detection is to recover the transmission symbol while preserving the key information of the signal as much as possible. With the development of computing science, deep learning has also been applied to channel estimation and signal detection [60]. Deep learning can realize end-to-end learning and replace traditional channel estimation and equalization. Compared with traditional methods, when using fewer training pilots and omitting cyclic prefixes, the deep learning method can still retain more signal key information and has a stronger robustness. However, deep learning also has some disadvantages. For example, the setting of initial key parameters is the method that obtains the optimal value through several manual experiments, which is inefficient and cannot guarantee the global optimal. Therefore, this paper intends to use the MEHHO to solve the low efficiency of network initialization parameter setting in the application of deep learning channel-estimation, and further verify the effectiveness and practicability of the MEHHO.

5.1. Channel Estimation and Signal Detection Model

In this paper, the proposed MEHHO is used to solve the problems of the low efficiency of manual parameter setting in deep learning, because the performance is greatly affected by manual parameter setting. The optimized deep learning algorithm is used to improve the channel estimation and signal detection capabilities of the Orthogonal Frequency-Division Multiplexing (OFDM) wireless communication system. In this section, we chose to optimize a Long-Short Term Memory (LSTM) network, because it has the function of long-term memory and is good at processing long sequence data, so it is more suitable for processing the transmission symbol sequence in communication. Since the channel estimation method based on a LSTM network takes a long time to train, and a large number of parameters need to be adjusted during the learning process, researchers have developed offline training and online deployment technologies as illustrated in references [61–63]. This technology uses the data generated by the channel model simulation to train the LSTM network model offline, and then directly uses the trained model for online deployment to recover the data transmitted online. This section uses offline training and online deployment technology to design experiments for channel estimation and signal detection based on a LSTM network. Firstly, the OFDM system and LSTM network model are established. Secondly, the key initialization parameters of the LSTM network are optimized in the offline training stage, namely, initial learning rate, training times, and batch size. Finally, the optimized LSTM network is used for OFDM channel estimation and signal detection during online deployment to obtain the minimum Symbol Error Rate (SER). In this experiment, the total symbol error rate of the prediction results of the training samples and the prediction results of the verification samples are selected as the fitness function. The description is shown in Equation (23):

Minimize

$$SER(I, N, M) = \left(1 - \frac{\hat{W}}{W}\right) + \left(1 - \frac{\hat{V}}{V}\right) \tag{23}$$

Subject to

$$I \in [I_{\min}, I_{\max}], N \in [N_{\min}, N_{\max}], M \in [M_{\min}, M_{\max}]$$

where \hat{W} is the number of correctly predicted training samples; W is the number of total training samples; \hat{V} is the number of correctly predicted verification samples; V is the number of total verification samples; I is the initial learning rate; N is the number of training; and M is the batch size.

The process of implementing the MEHHO-LSTM for channel estimation and detection is shown in Figure 6. The specific steps are as follows:

Step1: Establish the mathematical model of the OFDM system, and generate the training set, verification set, and test set required by LSTM model under the 3GPP TR38.901 channel model;

- Step2: Establish the LSTM channel estimation and signal detection model;
- Step3: Initialize the MEHHO, take the initial learning rate I , training times N , and batch size M in the LSTM model as optimization objectives, and establish the MEHHO-LSTM model corresponding to each dimension in the HHO;
- Step4: Calculate the fitness value of each individual according to Formula (23), and update the individual position according to the fitness value;
- Step5: Determine whether the maximum number of iterations is reached. If so, output the optimal solution position, namely the best parameter of LSTM; Otherwise, return to Step4.
- Step6: Substitute the optimal parameters into the LSTM network model for OFDM channel estimation and signal detection.

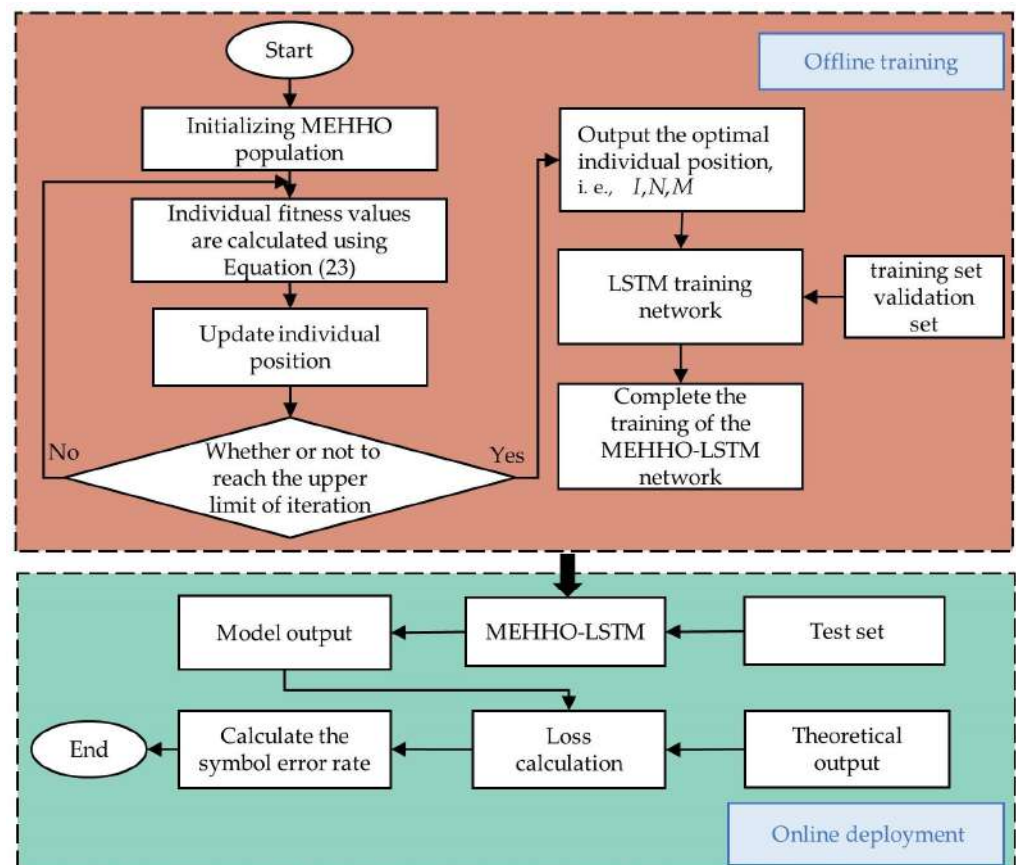


Figure 6. MEHHO-LSTM model channel estimation and signal detection process.

5.2. Experimental Parameter Setting

In this experiment, quadrature phase shift-keying (QPSK) modulation is adopted for the OFDM system. An OFDM wireless communication system with 64 subcarriers, 8 pilots, and a length of 16 cyclic prefixes is adopted. The channel model is 3GPP TR38.901, and the number of paths is 20. The ratio of the training set to the verification set is 8:2. The LSTM model used is composed of five layers, namely, the input layer, LSTM layer, fully connected layer, softmax layer, and classification layer. The input layer contains 256 neurons, the LSTM layer contains 16 neurons, and the fully connected layer contains 4 neurons. The Adam algorithm is used to train the internal parameters. The initialization parameters of the meta-heuristic algorithm used for testing are set identically. The fitness function is SER. The value ranges of the initial learning rate, training times, and batch size are $[0.005, 0.02]$, $[80, 100]$, and $[800, 1000]$, respectively. Two groups of LSTM networks are selected for comparison. The first group is LSTM1 with parameters set to non-empirical optimal values, and the accuracy is about 78% in the training set and 53% in the validation set. The second group is named LSTM2, and the parameters are set to empirical optimal

values, with 100% accuracy in both the training and validation sets. The parameters are set as follows: the initial learning rate of LSTM1 is 0.005, the number of training is 5, and the batch size is 500; the initial learning rate of LSTM2 is 0.02, the number of training is 100, and the batch size is 1000.

5.3. Results and Discussion

Firstly, the MEHHO-LSTM network is compared with LSTM1 and LSTM2 networks in channel estimation and signal detection performance. As can be seen from Figure 7, the performance of the LSTM1 network is the worst and cannot effectively reduce the SER, while the LSTM2 network is far superior to the LSTM1 network, indicating that the selection of key initialization parameters has a great impact on the network estimation and detection performance. When the signal-to-noise ratio (SNR) is greater than 10 dB, the MEHHO-LSTM method is better than LSTM2. When the SER is 10^{-2} , the performance is improved by 2–3 dB. The results show that the proposed algorithm can effectively optimize the key initial parameters of the LSTM network, to improve the accuracy of channel estimation and signal detection, and overcome the shortcomings of manually selecting the initial parameters of the LSTM network.

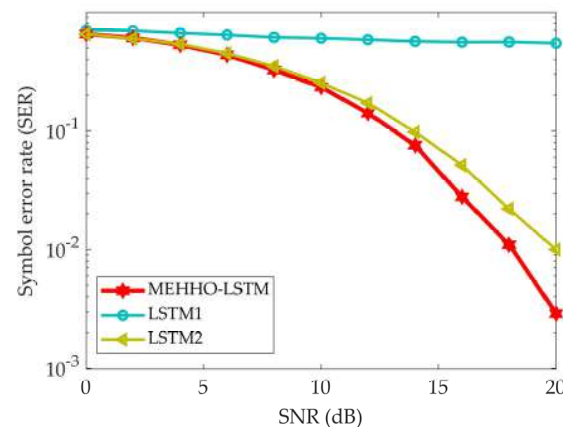


Figure 7. Performance comparison of LSTM1, LSTM2, and MEHHO-LSTM methods.

Figure 8 shows the MEHHO proposed in this paper and the original Harris Hawks Optimization (HHO, 2019) [34], Whale Optimization Algorithm (WOA, 2016) [9], Grey Wolf Optimizer (GWO, 2014) [54], Sine Cosine Algorithm (SCA, 2016) [64], performance comparison of the optimized LSTM network model in OFDM channel estimation and signal detection. As can be seen from Figure 8, with the increase in SNR, the SER of each method decreases. When the SNR of the MEHHO-LSTM is greater than 10 dB, the SER of the MEHHO-LSTM model obviously accelerates, and it is better than the other optimized network models. It is proved that the initial parameters of the model found by the proposed MEHHO are better, which can help the LSTM model learn the characteristics of the wireless channel better.

Figure 9 shows the comparison curve of the SER performance of the proposed MEHHO-LSTM algorithm, least squares (LS) algorithm, and minimum mean-square error (MMSE) algorithm with the change of SNR. It can be seen from the figure that the scheme using MEHHO-LSTM has the best performance, while the traditional LS and MMSE methods have a very slow rate of SER decline. When the SNR of the proposed model is greater than 10 dB, the rate of SER decline increases significantly. When the SER is 10^{-1} , the performance of the proposed algorithm is improved by 6–7 dB compared with the traditional LS and MMSE methods. This is because the traditional methods rely heavily on the number of pilots, but the MEHHO-LSTM proposed in this paper is robust to the number of pilots used for channel estimation. It can be shown that the MEHHO-LSTM network model, compared with traditional methods, can significantly reduce the SER of the OFDM system transmission signal, and has good channel estimation and signal detection performance.

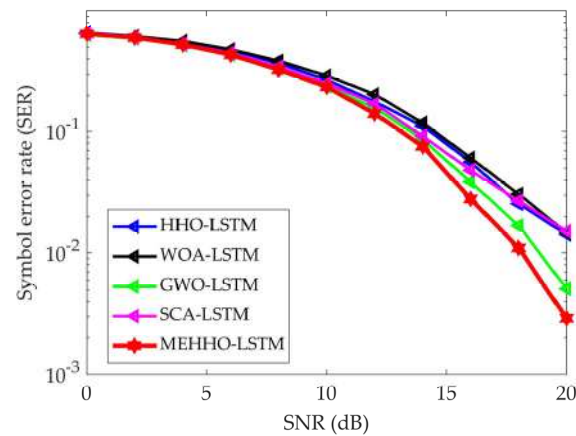


Figure 8. Comparison of the performance of the network model optimized by five algorithms.

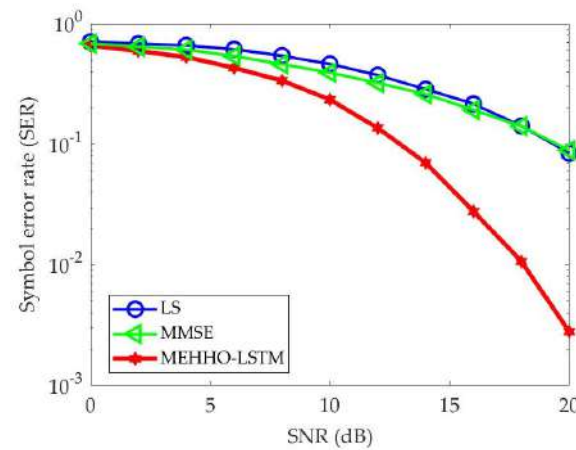


Figure 9. Performance comparison between traditional LS and MMSE methods and MEHHO-LSTM methods.

6. Conclusions and Prospect

This paper presents an improved version of the Harris Hawks Optimization, Multi-strategy Enhanced Harris Hawks Optimization. Firstly, the map-compass operator and the Cauchy mutation are introduced to enhance the population diversity and improve the ability of the algorithm to jump out of the local optimal. Secondly, the spiral motion strategy is used to improve the exploration stage and to retain the dominant individuals by greedy selection to improve the convergence speed and accuracy of the algorithm. The performance of the proposed MEHHO is compared with the original HHO, other meta-heuristic algorithms, and other improved HHOs in 28 benchmark test functions, and the significance level of the algorithm is evaluated using Wilcoxon signed-rank test. The experimental results show that the multi-strategy fusion design can better simulate the hunting behavior of Harris hawks, effectively improving the convergence speed and accuracy of the algorithm, and solving the problem of easy to fall into the local optimal. When solving unimodal, multimodal, low-dimensional, and high-dimensional functions, the MEHHO can obtain better optimization results, indicating that the algorithm has better accuracy, reliability, and universality in solving global optimization problems. Finally, the application ability of the MEHHO is further verified through the deep learning-based channel estimation and signal detection problems in the field of wireless communication. It shows that the algorithm has the ability to solve engineering application problems, and can adequately solve the optimization problems of parameter selection in engineering, providing efficient and reliable solutions.

MEHHO has greater development potential in the future, and there are many aspects to continue to study. First of all, although the strategy proposed in this paper effectively improves the optimization performance of HHO, the calculation time of individual test functions is long, and the performance of some of the fixed-dimension multimodal test functions still has room for improvement. It should be considered whether new strategies can be introduced to further improve the shortcomings of the algorithm. Secondly, in the multi-objective optimization algorithm, whether the improved strategy can achieve good results also needs further research. Finally, what other engineering application optimization problems the MEHHO can solve in the field of communication still need further consideration and experimentation.

Author Contributions: Conceptualization, Y.S. and Q.H.; methodology, Y.S. and Q.H.; software, Y.S. and Q.H.; validation, Q.H. and T.L.; formal analysis, Y.S. Q.H., and T.L.; investigation, Y.C. and Y.L.; resources, T.L. and Y.C.; data curation, Q.H. and Y.L.; writing—original draft preparation, Y.S. and Q.H.; writing—review and editing, Y.S. and T.L.; visualization, Y.S. and T.L.; supervision, Q.H. and T.L.; project administration, Y.S. and T.L.; funding acquisition, Y.S. and T.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Arufe, L.; González, M.A.; Oddi, A.; Rasconi, R.; Varela, R. Quantum circuit compilation by genetic algorithm for quantum approximate optimization algorithm applied to maxcut problem. *Swarm Evol. Comput.* **2022**, *69*, 101030. [[CrossRef](#)]
2. Wang, M.; Ma, Y.; Wang, P. Parameter and strategy adaptive differential evolution algorithm based on accompanying evolution. *Inf. Sci.* **2022**, *607*, 1136–1157. [[CrossRef](#)]
3. Alnowibet, K.A.; Mahdi, S.; El-Alem, M.; Abdelawwad, M.; Mohamed, A.W. Guided Hybrid Modified Simulated Annealing Algorithm for Solving Constrained Global Optimization Problems. *Mathematics* **2022**, *10*, 1312. [[CrossRef](#)]
4. Hashim, F.A.; Houssein, E.H.; Mabrouk, M.S.; Al-Atabany, W.; Mirjalili, S. Henry gas solubility optimization: A novel physics-based algorithm. *Futur. Gener. Comp. Syst.* **2019**, *101*, 646–667. [[CrossRef](#)]
5. Zhao, W.; Wang, L.; Zhang, Z. A novel atom search optimization for dispersion coefficient estimation in groundwater. *Futur. Gener. Comp. Syst.* **2019**, *91*, 601–610. [[CrossRef](#)]
6. Kassoul, K.; Zufferey, N.; Cheikhrouhou, N.; Belhaouari, S.B. Exponential particle swarm optimization for global optimization. *IEEE Access* **2022**, *10*, 78320–78344. [[CrossRef](#)]
7. Li, S.; Wei, Y.; Liu, X.; Zhu, H.; Yu, Z. A New Fast Ant Colony Optimization Algorithm: The Saltatory Evolution Ant Colony Optimization Algorithm. *Mathematics* **2022**, *10*, 925. [[CrossRef](#)]
8. Kaya, E. A New Neural Network Training Algorithm Based on Artificial Bee Colony Algorithm for Nonlinear System Identification. *Mathematics* **2022**, *10*, 3487. [[CrossRef](#)]
9. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [[CrossRef](#)]
10. Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **2017**, *114*, 163–191. [[CrossRef](#)]
11. Arora, S.; Singh, S. Butterfly optimization algorithm: A novel approach for global optimization. *Soft Comput.* **2019**, *23*, 715–734. [[CrossRef](#)]
12. Faramarzi, A.; Heidarinejad, M.; Mirjalili, S.; Gandomi, A.H. Marine Predators Algorithm: A nature-inspired metaheuristic. *Expert Syst. Appl.* **2020**, *152*, 113377. [[CrossRef](#)]
13. Zervoudakis, K.; Tsafarakis, S. A mayfly optimization algorithm. *Comput. Ind. Eng.* **2020**, *145*, 106559. [[CrossRef](#)]
14. Abualigah, L.; Yousefi, D.; Abd Elaziz, M.; Ewees, A.A.; Al-Qaness, M.A.; Gandomi, A.H. Aquila optimizer: A novel meta-heuristic optimization algorithm. *Comput. Ind. Eng.* **2021**, *157*, 107250. [[CrossRef](#)]
15. Abualigah, L.; Diabat, A.; Mirjalili, S.; Abd Elaziz, M.; Gandomi, A.H. The arithmetic optimization algorithm. *Comput. Meth. Appl. Mech. Eng.* **2021**, *376*, 113609. [[CrossRef](#)]
16. Seyyedabbasi, A.; Kiani, F. Sand Cat swarm optimization: A nature-inspired algorithm to solve global optimization problems. *Eng. Comput.* **2022**, *38*, 1–25. [[CrossRef](#)]
17. Ahmadianfar, I.; Heidari, A.A.; Noshadian, S.; Chen, H.; Gandomi, A.H. INFO: An efficient optimization algorithm based on weighted mean of vectors. *Expert Syst. Appl.* **2022**, *195*, 116516. [[CrossRef](#)]
18. Abd El-Sattar, H.; Kamel, S.; Hassan, M.H.; Jurado, F. Optimal sizing of an off-grid hybrid photovoltaic/biomass gasifier/battery system using a quantum model of Runge Kutta algorithm. *Energy Conv. Manag.* **2022**, *258*, 115539. [[CrossRef](#)]

19. Shao, K.; Fu, W.; Tan, J.; Wang, K. Coordinated approach fusing time-shift multiscale dispersion entropy and vibrational Harris hawks optimization-based SVM for fault diagnosis of rolling bearing. *Measurement* **2021**, *173*, 108580. [[CrossRef](#)]
20. Zhang, A.; Yu, D.; Zhang, Z. TLSCA-SVM Fault Diagnosis Optimization Method Based on Transfer Learning. *Processes* **2022**, *10*, 362. [[CrossRef](#)]
21. Dokeroglu, T.; Deniz, A.; Kiziloz, H.E. A Comprehensive Survey on Recent Metaheuristics for Feature Selection. *Neurocomputing* **2022**, *494*, 269. [[CrossRef](#)]
22. Ewees, A.A.; Ismail, F.H.; Ghoniem, R.M.; Gaheen, M.A. Enhanced Marine Predators Algorithm for Solving Global Optimization and Feature Selection Problems. *Mathematics* **2022**, *10*, 4154. [[CrossRef](#)]
23. Abbasi, A.; Firouzi, B.; Sendur, P. On the application of Harris hawks optimization (HHO) algorithm to the design of microchannel heat sinks. *Eng. Comput.* **2021**, *37*, 1409–1428. [[CrossRef](#)]
24. Chacko, A.; Chacko, S. Deep learning-based robust medical image watermarking exploiting DCT and Harris hawks optimization. *Int. J. Intell. Syst.* **2022**, *37*, 4810–4844. [[CrossRef](#)]
25. Badashah, S.J.; Basha, S.S.; Ahamed, S.R.; Subba Rao, S.P.V. Fractional-Harris hawks optimization-based generative adversarial network for osteosarcoma detection using Renyi entropy-hybrid fusion. *Int. J. Intell. Syst.* **2021**, *36*, 6007–6031. [[CrossRef](#)]
26. Bandyopadhyay, R.; Kundu, R.; Oliva, D.; Sarkar, R. Segmentation of brain MRI using an altruistic Harris Hawks' Optimization algorithm. *Knowl. -Based Syst.* **2021**, *232*, 107468. [[CrossRef](#)]
27. Kaur, N.; Kaur, L.; Cheema, S.S. An enhanced version of Harris Hawks optimization by dimension learning-based hunting for breast cancer detection. *Sci. Rep.* **2021**, *11*, 21933. [[CrossRef](#)]
28. Alweshah, M.; Almiani, M.; Almansour, N.; Al Khalailah, S.; Aldabbas, H.; Alomoush, W.; Alshareef, A. Vehicle routing problems based on Harris Hawks optimization. *J. Big Data* **2022**, *9*, 42. [[CrossRef](#)]
29. Zhang, R.; Li, S.; Ding, Y.; Qin, X.; Xia, Q. UAV Path Planning Algorithm Based on Improved Harris Hawks Optimization. *Sensors* **2022**, *22*, 5232. [[CrossRef](#)]
30. Golilarz, N.A.; Addeh, A.; Gao, H.; Ali, L.; Roshandeh, A.M.; Munir, H.M.; Khan, R.U. A new automatic method for control chart patterns recognition based on ConvNet and harris hawks meta heuristic optimization algorithm. *IEEE Access* **2019**, *7*, 149398–149405. [[CrossRef](#)]
31. Abd Elaziz, M.; Abualigah, L.; Ibrahim, R.A.; Attiya, I. IoT workflow scheduling using intelligent arithmetic optimization algorithm in fog computing. *Comput. Intell. Neurosci.* **2021**, *2021*, 9114113. [[CrossRef](#)] [[PubMed](#)]
32. Seyfollahi, A.; Ghaffari, A. Reliable data dissemination for the Internet of Things using Harris hawks optimization. *Peer Peer Netw. Appl.* **2020**, *13*, 1886–1902. [[CrossRef](#)]
33. Saravanan, G.; Ibrahim, A.M.; Kumar, D.S.; Vanitha, U.; Chandrika, V.S. Iot based speed control of bldc motor with Harris hawks optimization controller. *Int. J. Grid Distrib. Comput.* **2020**, *13*, 1902–1915.
34. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Futur. Gener. Comp. Syst.* **2019**, *97*, 849–872. [[CrossRef](#)]
35. Ramadan, A.; Kamel, S.; Korashy, A.; Almalaq, A.; Domínguez-García, J.L. An enhanced Harris Hawk optimization algorithm for parameter estimation of single, double and triple diode photovoltaic models. *Soft Comput.* **2022**, *26*, 7233–7257. [[CrossRef](#)]
36. Naeijian, M.; Rahimnejad, A.; Ebrahimi, S.M.; Pourmousa, N.; Gadsden, S.A. Parameter estimation of PV solar cells and modules using Whippy Harris Hawks Optimization Algorithm. *Energy Rep.* **2021**, *7*, 4047–4063. [[CrossRef](#)]
37. Chen, H.; Jiao, S.; Wang, M.; Heidari, A.A.; Zhao, X. Parameters identification of photovoltaic cells and modules using diversification-enriched Harris hawks optimization with chaotic drifts. *J. Clean Prod.* **2020**, *244*, 118778. [[CrossRef](#)]
38. Jia, H.; Lang, C.; Oliva, D.; Song, W.; Peng, X. Dynamic harris hawks optimization with mutation mechanism for satellite image segmentation. *Remote Sens.* **2019**, *11*, 1421. [[CrossRef](#)]
39. Wunnava, A.; Naik, M.K.; Panda, R.; Jena, B.; Abraham, A. A differential evolutionary adaptive Harris hawks optimization for two dimensional practical Masi entropy-based multilevel image thresholding. *J. King Saud Univ.-Comput. Inf. Sci.* **2020**, *34*, 3011–3024. [[CrossRef](#)]
40. Rodríguez-Esparza, E.; Zanella-Calzada, L.A.; Oliva, D.; Heidari, A.A.; Zaldivar, D.; Pérez-Cisneros, M.; Foong, L.K. An efficient Harris hawks-inspired image segmentation method. *Expert Syst. Appl.* **2020**, *155*, 113428. [[CrossRef](#)]
41. Srinivas, M.; Amgoth, T. EE-hHHSS: Energy-efficient wireless sensor network with mobile sink strategy using hybrid Harris hawk-salp swarm optimization algorithm. *Int. J. Commun. Syst.* **2020**, *33*, 4569. [[CrossRef](#)]
42. Bhat, S.J.; Venkata, S.K. An optimization based localization with area minimization for heterogeneous wireless sensor networks in anisotropic fields. *Comput. Netw.* **2020**, *179*, 107371. [[CrossRef](#)]
43. Kaveh, A.; Rahmani, P.; Eslamlou, A.D. An efficient hybrid approach based on Harris Hawks optimization and imperialist competitive algorithm for structural optimization. *Eng. Comput.* **2022**, *38*, 1555–1583. [[CrossRef](#)]
44. Dehkordi, A.A.; Sadiq, A.S.; Mirjalili, S.; Ghafoor, K.Z. Nonlinear-based chaotic harris hawks optimizer: Algorithm and internet of vehicles application. *Appl. Soft. Comput.* **2021**, *109*, 107574. [[CrossRef](#)]
45. Gupta, S.; Deep, K.; Heidari, A.A.; Moayedi, H.; Wang, M. Opposition-based learning Harris hawks optimization with advanced transition rules: Principles and analysis. *Expert Syst. Appl.* **2020**, *158*, 113510. [[CrossRef](#)]
46. Fan, Q.; Chen, Z.; Xia, Z. A novel quasi-reflected Harris hawks optimization algorithm for global optimization problems. *Soft Comput.* **2020**, *24*, 14825–14843. [[CrossRef](#)]
47. Liu, C. An improved Harris hawks optimizer for job-shop scheduling problem. *J. Supercomput.* **2021**, *77*, 14090–14129. [[CrossRef](#)]

48. Abualigah, L.; Diabat, A.; Altalhi, M.; Elaziz, M.A. Improved gradual change-based Harris Hawks optimization for real-world engineering design problems. *Eng. Comput.* **2022**, *38*, 1–41. [[CrossRef](#)]
49. Duan, H.; Qiao, P. Pigeon-inspired optimization: A new swarm intelligence optimizer for air robot path planning. *Int. J. Intell. Comput. Cybern.* **2014**, *7*, 24–37. [[CrossRef](#)]
50. Luo, J.; Chen, H.; Heidari, A.A.; Xu, Y.; Zhang, Q.; Li, C. Multi-strategy boosted mutative whale-inspired optimization approaches. *Appl. Math. Model.* **2019**, *73*, 109–123. [[CrossRef](#)]
51. Ou, X.; Wu, M.; Pu, Y.; Tu, B.; Zhang, G.; Xu, Z. Cuckoo search algorithm with fuzzy logic and Gauss–Cauchy for minimizing localization error of WSN. *Appl. Soft Comput.* **2022**, *125*, 109211. [[CrossRef](#)]
52. Jia, H.; Peng, X.; Lang, C. Remora optimization algorithm. *Expert Syst. Appl.* **2021**, *185*, 115665. [[CrossRef](#)]
53. Hayyolalam, V.; Kazem, A.A.P. Black widow optimization algorithm: A novel meta-heuristic approach for solving engineering optimization problems. *Eng. Appl. Artif. Intell.* **2020**, *87*, 103249. [[CrossRef](#)]
54. Mohammadi-Balani, A.; Nayeri, M.D.; Azar, A.; Taghizadeh-Yazdi, M. Golden eagle optimizer: A nature-inspired metaheuristic algorithm. *Comput. Ind. Eng.* **2021**, *152*, 107050. [[CrossRef](#)]
55. Zhao, W.; Zhang, Z.; Wang, L. Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications. *Eng. Appl. Artif. Intell.* **2020**, *87*, 103300. [[CrossRef](#)]
56. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [[CrossRef](#)]
57. Sarma, R.; Bhargava, C.; Jain, S.; Kamboj, V.K. Application of ameliorated Harris Hawks optimizer for designing of low-power signed floating-point MAC architecture. *Neural Comput. Appl.* **2021**, *33*, 8893–8922. [[CrossRef](#)]
58. Nandi, A.; Kamboj, V.K. A Canis lupus inspired upgraded Harris hawks optimizer for nonlinear, constrained, continuous, and discrete engineering design problem. *Int. J. Numer. Methods Eng.* **2021**, *122*, 1051–1088. [[CrossRef](#)]
59. Kamboj, V.K.; Nandi, A.; Bhadoria, A.; Sehgal, S. An intensify Harris Hawks optimizer for numerical and engineering optimization problems. *Appl. Soft. Comput.* **2020**, *89*, 106018. [[CrossRef](#)]
60. Ye, H.; Li, G.Y.; Juang, B.H. Power of deep learning for channel estimation and signal detection in OFDM systems. *IEEE Wirel. Commun. Lett.* **2017**, *7*, 114–117. [[CrossRef](#)]
61. Essai Ali, M.H. Deep learning-based pilot-assisted channel state estimator for OFDM systems. *IET Commun.* **2021**, *15*, 257–264. [[CrossRef](#)]
62. Ali, M.H.E.; Taha, I.B. Channel state information estimation for 5G wireless communication systems: Recurrent neural networks approach. *PeerJ Comput. Sci.* **2021**, *7*, e682.
63. Mai, Z.; Chen, Y.; Zhao, H.; Du, L.; Hao, C. A UAV Air-to-Ground Channel Estimation Algorithm Based on Deep Learning. *Wirel. Pers. Commun.* **2022**, *124*, 2247–2260. [[CrossRef](#)]
64. Mirjalili, S. SCA: A sine cosine algorithm for solving optimization problems. *Knowl. -Based Syst.* **2016**, *96*, 120–133. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.