

Received July 15, 2020, accepted July 22, 2020, date of publication July 31, 2020, date of current version September 14, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3013332

Improved Harris Hawks Optimization Based on Adaptive Cooperative Foraging and Dispersed Foraging Strategies

XUNCAI ZHANG¹, (Member, IEEE), KAI ZHAO¹, AND YING NIU²

¹College of Electrical and Information Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China

²College of Architecture Environment Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China

Corresponding author: Ying Niu (niuying@zzuli.edu.cn)

This work was supported in part by the Key Research and Development Program of Henan Province under Grant 202102210177 and Grant 192102210134, and in part by the National Natural Science Foundation of China under Grant 61602424 and Grant U1804262.

ABSTRACT Harris hawks optimization (HHO) is a new swarm intelligence optimization technique. Because of its simple structure and easy to implement, HHO has attracted research interest from scholars in different fields. However, the low population diversity and the single search method in the exploration phase weakened the global search capability of the HHO algorithm. In response to these defects, this paper proposes an improved HHO algorithm based on adaptive cooperative foraging and dispersed foraging strategies. First, the adaptive cooperative foraging strategy uses three random individuals to guide the position update, which achieves cooperation between individuals. Then the cooperation behavior is embedded in the one-dimensional update operation framework, and the one-dimensional or total-dimensional update operation is adaptively selected. This way allows the algorithm to perform position update operations for a specific dimension of individual vectors with a certain probability, which improves the population diversity. Second, the dispersed foraging strategy is introduced into the HHO, forcing a part of Harris hawks to leave their current position to find more prey to obtain a better candidate solution. This way effectively avoids the algorithm falling into local optimum. Finally, a randomly shrinking exponential function is used to simulate the energy change of the prey, so that the algorithm maintains the exploration ability in the later exploitation process, effectively balancing the exploration and exploitation ability of the algorithm. The performance of the proposed ADHHO algorithm is evaluated using Wilcoxon's test on unimodal, multimodal and CEC 2014 benchmark functions. Numerical results and statistical experiments show that ADHHO provides better solution quality, convergence accuracy and stability compared with other state-of-the-art algorithms.

INDEX TERMS Harris hawks optimization, adaptive cooperative foraging, dispersed foraging, Wilcoxon's test, CEC 2014 benchmark functions.

I. INTRODUCTION

Optimization is the process of finding the best solution for all feasible solutions to a particular problem. The purpose of optimization is to consume the least cost and use limited resources to maximize profits, efficiency and performance. In recent years, as the optimization problems presented in different fields become more and more complex, the method of finding the optimal solution based on gradient information is challenging to adapt to the complex challenges of large-scale, multi-constraint, multi-modal, high-dimensional and

other characteristics [1]–[4]. As a result, the demand for more efficient algorithms is increasing. In the past few decades, swarm intelligence (SI) has shown high efficient and robust performance in solving modern nonlinear numerical global optimization problems [5], [6].

SI algorithm is a typical nature-inspired optimization algorithm, which simulates the social behavior of groups of animals [6]. Because of its simplicity, flexibility, no gradient information, and the ability to bypass local optimum, SI has been widely used in different disciplines and engineering fields [7]–[10]. In the past two decades, a large number of SI algorithms have been proposed. Particle Swarm Optimization (PSO) [11] algorithm simulates the social

The associate editor coordinating the review of this manuscript and approving it for publication was Yilun Shang¹.

behavior bird or fish swarm. It has the advantages of simple structure and few parameter settings. However, the PSO algorithm is easy to fall into local optimum [12]. Artificial Bee Colony algorithm (ABC) [13] is a simulation of a particular behavior of honeybees known as foraging behavior. The Ant Colony Optimization (ACO) algorithm is inspired by the behavior of finding the optimal path in the ant colony foraging process [14]. The pheromone matrix of the ant colony evolves continuously during the iteration of the candidate solution and eventually obtains the optimal solution. It is very effective in solving path planning problems [15]. Cuckoo Search algorithm (CS) [16] is another swarm intelligence algorithm, which is based on the obligate brood parasitic behavior of certain cuckoo species [17]. However, such algorithms are not very popular due to their high time complexity. Fruit fly optimization algorithm (FOA) [18] was proposed in 2011, inspired by the cooperative foraging behavior of fruit flies. Besides, many new SI algorithms appear every year, such as grey wolf optimization algorithm (GWO) [19], firefly algorithm (FF) [20], dragonfly algorithm (DA) [21], squirrel search algorithm (SSA) [22], [23]. It is worth mentioning that some recently proposed meta-heuristic algorithms, such as Henry gas solubility (HGS) [24], slime mould algorithm (SMA) [25], equilibrium optimizer (EO) [26] and quasi-affine transformation evolutionary (QUATRE) [27] show excellent performance in solving optimization problems. The HGS algorithm simulates Henry's law behavior and was recently developed by Hashim *et al.* In HGS, population agents were divided into clusters with the same number of gas types. The gas equilibrium state was updated by updating each cluster of gases' solubility at different temperatures and pressures. The resulting gas in the highest equilibrium state is the optimal solution. The advantage of HGS is that it has the search mechanism of the worst gas neighborhood, thereby improving the global search capability. SMA mainly simulates the behavior and morphological changes of slime mould during foraging. When the food source's quality is different, the slime mould can establish a moving route in the direction of a higher concentration, thereby ensuring the maximum concentration of nutritional requirements. SMA has an excellent balance between exploration and exploitation, so it could effectively avoid local optimization. The inspiration of the EO algorithm comes from the dynamic mass balance on the control volume. In EO, the mass balance equation is used to measure the number of mass items generated in the volume over time and seek to find the state that achieves the equilibrium of the system. EO is easy to implement and has a high population diversity. Therefore, it has been applied to many real-world optimization problems such as multi-thresholding image segmentation problems [28], optimal structural design problems [29]. The QUATRE is a co-evolution framework based on quasi-affine transformation. It has been proved that excellent optimization performance could be obtained on large-scale optimization problems [27], [30], [31]. Due to its simple structure and high flexibility, the QUATRE algorithm has been

successfully applied to text feature extraction and has achieved good results [32].

As a new bionic optimization algorithm, the swarm intelligence algorithm has developed rapidly in recent years. However, according to the no free lunch (NFL) theorem, one algorithm cannot be regarded as a general optimizer to solve all optimization problems [33]. NFL Theorem encourages scholars to propose new optimization algorithms or improve classical optimization algorithms to achieve better optimization performance. Therefore, Heidari *et al.* proposed a new swarm intelligence algorithm in 2019, Harris hawks Optimization (HHO) algorithm [34], by simulating the cooperative behavior of Harris hawks in the process of hunting prey. The simulation experiments of 29 benchmark functions and several engineering optimization problems proved the effectiveness of HHO in optimization problems. At the same time, the HHO algorithm has the advantages of simple operation, few adjustment parameters and easy to implement, so it has been applied to solve actual optimization problems in many disciplines. For example, image segmentation [35], structure optimization [36], image denoising [37], parameter identification [38], layout optimization [39], and power load distribution [40].

Although HHO has been successfully applied to various practical optimization problems, the algorithm itself still has some shortcomings. Since the HHO algorithm introduced four different attack strategies, it has a significant advantage in exploitation capability. However, the choice of these four attack strategies is based on random parameters, resulting in an imbalance between the exploration and exploitation capabilities of the algorithm. When dealing with multimodal or modern highly complex optimization tasks, the convergence accuracy is low, and it is easy to fall into local optimization. Besides, the HHO algorithm ignored the later global search capabilities. More specifically, the escape energy E in the later stage of the iteration is always less than 1, and the hawks are always in the stage of attacking prey. In this way, there is no guarantee that the population has gathered around the optimum at the end of the exploration phase, resulting in premature convergence. In response to these problems, some scholars have proposed improvement strategies from different perspectives. For example, literature [41] introduced long-term memory to the HHO algorithm, allowing individuals to exercise based on experience, increasing the population's diversity. However, it ignored the running time of the algorithm and was less effective in high-dimensional problems. Reference [35] used dynamic control parameters to reduce the probability of the HHO algorithm falling into a local optimum and used mutation operators to improve the global search capability further. Literature [42] added interference terms to the escape energy to control the location of the disturbance peaks, which increased the global search capability in the later stage. Besides, some scholars combined with the exploration ability of other algorithms to improve HHO, such as combining sine and cosine algorithm [43], simulated annealing

TABLE 1. Recent techniques to improve the SSA.

Algorithm	Improvement measures	Advantages	Disadvantages	Application
EHHO [36]	Introduce opposition-based learning (OBL) to explore the solution space; chaotic local search (CLS) is used to develop potential solutions.	Using OBL and CLS can prevent the algorithm from falling into the local optimum; improve solution accuracy.	Poor performance in multimodal problems; high time complexity.	Optimal processing parameter selection
CHHO [46]	Using Chaos for Population Initialization.	Increase search diversity; improve algorithm convergence speed.	It is easy to fall into local optimal in some high-dimensional problems.	Extracting Parameters of Proton Exchange Membrane Fuel Cell Stacks
MHHO [47]	Modify the position update formula in the exploration stage to achieve collaborative behavior between individuals.	Use local communication between Harris hawks to improve global search capabilities; strong robustness.	High time cost; poor convergence performance for high-dimensional problems.	Layout optimization
BHHO [48]	Introduce biological pollination and cross-pollination mechanisms into the exploration phase; combine non-biological pollination and self-pollination to attack prey.	Strong ability to deal with multimodal nonlinear optimization problems; convergence accuracy is better than HHO; strong robustness.	The algorithm is greatly affected by the parameters, and flexibility is not high.	Parameter identification of a single diode solar cell model

algorithm [44] and dragonfly algorithm [45]. However, the above improvements are generally aimed at improving exploration capabilities, and the lack of a balanced method between search capabilities makes the robustness and search results under multimodal or modern highly complex optimization tasks generally weak. In addition to the above improvement measures, TABLE 1 summarizes the advantages, disadvantages and applications of other HHO variants.

Given the above discussion, this paper proposes a Harris hawks optimization algorithm based on adaptive cooperative foraging and dispersed foraging strategies, and proposes three improvements to the above deficiencies. The main contributions of this article are summarized as follows:

1) Introduced an adaptive cooperative foraging strategy that randomly selected some Harris hawk individuals to forage cooperatively. Then this foraging strategy was embedded in the one-dimensional position update framework, and the one-dimensional update operation and the traditional total-dimensional update operation were adaptively selected according to the conversion factor. This strategy effectively improved the population diversity of the HHO algorithm.

2) Proposed a dispersed foraging strategy that randomly distributed some Harris hawk individuals to other areas for foraging, to avoid the algorithm falling into local optimum.

3) The randomly shrinking exponential function was used to simulate the energy consumption process of the prey, and effectively solved the problem of imbalance between exploration and exploitation in the later stage of the algorithm.

4) The proposed algorithm was compared with other famous metaheuristic algorithms based on the convergence curves and statistical measures (e.g., mean, best, worst, standard deviation).

The rest of the paper is organized as follows. Section II briefly introduced the HHO algorithm. Section III detailed the proposed ADHHO algorithm based on adaptive foraging and dispersed foraging strategies. Section IV gave the experimental results and analysis. The proposed method was applied to a practical problem in Section V. Finally, Section VI gave a conclusion.

II. AN OVERVIEW OF HARRIS HAWKS OPTIMIZATION

The HHO algorithm was inspired by Harris hawk’s foraging behavior and attack strategy. The search process consists of three parts: the exploration phase, the transformation of exploration and exploitation, and the exploitation phase. The first stage involves waiting, finding and discovering prey. The X_{prey} is regarded as the prey position, and then, the hawk’s position is defined as follows:

$$X_i^{t+1} = \begin{cases} X_{rand}^t - r_1 |X_{rand}^t - 2r_2 X_i^t| & q \geq 0.5 \\ X_{prey}^t - X_m^t - r_3 (lb + r_4 (ub - lb)) & q < 0.5 \end{cases} \quad (1)$$

where X_{rand}^t represents a random hawk in the current iteration. q is a random value in the range $[0, 1]$. r_1, r_2, r_3 and r_4 are random numbers in between $[0, 1]$. X_m^t denotes the average position of all hawks and can be calculated as follows.

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

where X_i^t is the position of the i^{th} Harris hawk in iteration t and N represents the size of population.

At the second stage, the transition from exploration and exploitation depends on the escape energy E of prey. The mathematical model based on prey escape energy behavior is as follows:

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

where E_0 represents the initial energy of the prey, which is a random value in the range $[-1, 1]$. T is the maximum iteration. During the iteration, If $|E| \geq 1$, the algorithm is in the exploration phase; otherwise, the exploitation phase gets started.

In this last phase, Harris hawks suddenly attack their prey, which is the exploitation phase of the algorithm. There are four attack strategies available. Here, r is regarded as the chance of the target's escaping probability.

When $E \geq 0.5$ and $r \geq 0.5$, Harris hawks use soft besiege strategy to surround prey slowly. The mathematical model is as follows:

$$X_i^{t+1} = \Delta X_i^t - E |JX_{prey} - X_i^t|, \quad \Delta X_i^t = X_{prey} - X_i^t \quad (4)$$

where ΔX_i^t represents the difference between the position vector of the prey and the current individual, J represents the jumping intensity of the prey during the escape, and it is a random value between $[0, 2]$.

When $E < 0.5$, $r \geq 0.5$, the prey cannot escape due to insufficient escape energy, in this case, the position of the Harris hawks can be updated as follows:

$$X_i^{t+1} = X_{prey} - E |\Delta X_i^t| \quad (5)$$

When $E \geq 0.5$, $r < 0.5$, the prey has enough energy to escape successfully so that the Harris hawks perform soft besiege with progressive rapid dive strategy to confuse prey. It can be mathematically modelled as:

$$X_i^{t+1} = \begin{cases} Y = X_{prey} - E |JX_{prey} - X_i^t| & \text{if } f(Y) < f(X_i^t) \\ Z = Y + S \times \text{Lévy}(d) & \text{if } f(Z) < f(X_i^t) \end{cases} \quad (6)$$

where d represents the dimension of the problem and S denotes a random vector of size $1 \times d$. The Lévy is the Levy Flight function [34].

When $E < 0.5$, $r < 0.5$, the prey does not have enough energy to escape. Harris hawks use the following methods to attack prey:

$$X_i^{t+1} = \begin{cases} X_{prey} - E |JX_{prey} - X_m^t| & \text{if } f(Y) < f(X_i^t) \\ Z = Y + S \times \text{Lévy}(d) & \text{if } f(Z) < f(X_i^t) \end{cases} \quad (7)$$

FIGURE 1 shows the predation process of Harris hawks. And the flow chart of original HHO has been shown in FIGURE 2.

III. THE PROPOSED ADHHO ALGORITHM

In this section, a new ADHHO algorithm was proposed to improve the performance of the original HHO algorithm. There are three main innovations. First, an adaptive cooperative foraging strategy was introduced into the original HHO algorithm and used the average distance of some randomly selected individuals to guide the search agent to update their position to realize the cooperative mechanism. Then, according to the conversion factor CF , one-dimensional and total-dimensional update operations were adaptively selected to improve the population diversity. Second, proposed a dispersed foraging strategy, distributing some Harris hawk individuals to other regions to realize the exploration of search

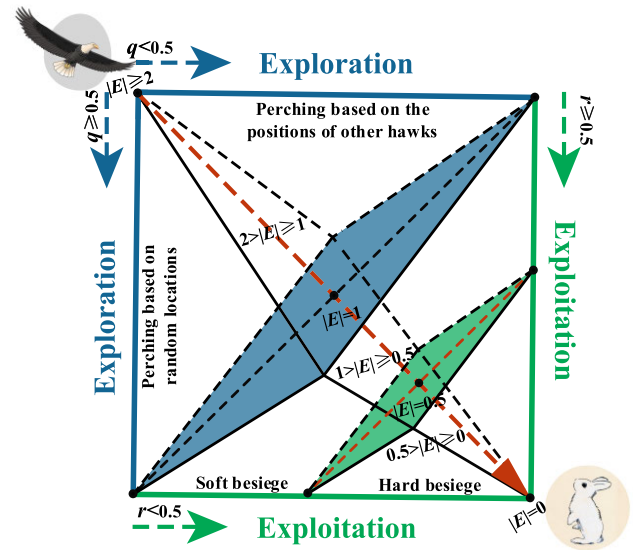


FIGURE 1. The predation process of Harris hawks.

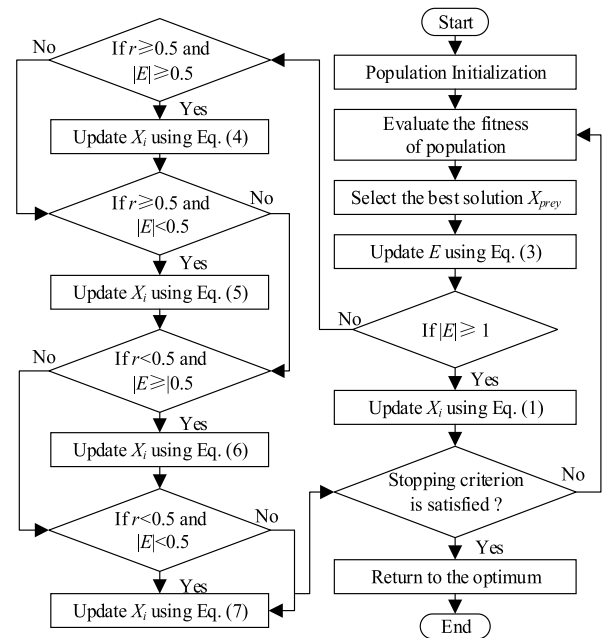


FIGURE 2. The flow chart of original HHO.

space and avoid the algorithm falling into local optimum. The third is to introduce a randomly shrinking exponential function to simulate the escape energy of the prey. And part of the exploration capability is retained in the exploitation stage of the late search, thereby achieving a balance between exploration and exploitation.

A. ADAPTIVE COOPERATIVE FORAGING STRATEGY

In the original HHO, a single global optimal position (X_{prey}) was used to guide the hawk's swarm to update the position. However, it ignores that there may be an optimal solution in the vicinity of individuals with poor fitness, leading to premature convergence of the algorithm [47], [49].

Algorithm 1 One-Dimensional Update Operation

Input: Randomly select one dimensional: $j_{rand} \in \{1, 2, \dots, D\}$

Output: Updated search agents population X^{t+1}

- 1: **for** $j = 1$ to D **do**
- 2: **if** $rand \leq CF$ || $j = j_{rand}$ **then**
- 3: $X_{i,d}^{t+1} = X_{i,j}^t$ %Update the position of individual by Eq. (8)
- 4: **else**
- 5: $X_{i,d}^{t+1} = X_{i,d}^t$
- 6: **end for**

Also, similar to many other meta-heuristic algorithms, HHO still uses a total-dimensional update operation to update all dimensions of individual vectors. This method is not easy to obtain high-quality solutions for multimodal and inseparable functions [50], [51]. In response to these shortcomings, this study introduced an adaptive cooperative foraging strategy in the exploration phase.

First, in the exploration stage, three Harris hawk individuals were randomly selected as the guiding particles in the search space, and the average distance between the search agent and guiding particles was used as the search step size to guide the search process jointly. This search process is considered to be the cooperative foraging process of the population. The new location update formula is as follows:

$$X_i^{t+1} = \begin{cases} X_{rand}^t - r_1 |X_{rand}^t - 2r_2 X_i^t| & q \geq 0.5 \\ r_3 \frac{((X_{rand_1}^t - X_i^t) + (X_{rand_2}^t - X_i^t) + (X_{rand_3}^t - X_i^t))}{3} + X_i^t & q < 0.5 \end{cases} \quad (8)$$

where $rand_1, rand_2, rand_3$ are random integers in $\{1, 2, 3, \dots, N\}$. q is a random value in the range $[0, 1]$. r_1, r_2, r_3 are random numbers between $[0, 1]$. The second line in the formula (8) is the proposed new search equation. Different from the single optimal position guidance of the original algorithm, the guiding particles of the new search equation are based on the random vector $(X_{rand_1}^t, X_{rand_2}^t, X_{rand_3}^t)$. In this way, the quality of the guiding particles may be good or bad, and this randomness improves the limitation that the original HHO ignores the optimal solution near the individual with poor fitness. At the same time, it avoids the premature convergence caused by a single global optimal position guidance, which helps the algorithm to explore further.

Then, the cooperative foraging process of the population was embedded in the one-dimensional update operation. It is different from the traditional total-dimensional update operation to update all dimensions of the individual, and in the one-dimensional update operation, each iteration only updated a certain dimension value of the Harris hawk individuals vector. FIGURE 3 shows the population diversity curve of one-dimensional update operation and total-dimensional update

Algorithm 2 Adaptable Cooperative Foraging Strategy

Input: The search agent population X^t , parameter CF

Output: Updated search agents population X^{t+1}

- 1: **for** $i = 1$ to N **do**
- 2: **if** $rand \leq CF$ **then**
- 3: Generate a rand integer $j_{rand} (j_{rand} \in \{1, 2, \dots, D\})$
- 4: $j = j_{rand}$
- 5: **else**
- 6: $j = 1$ to D
- 7: **end if**
- 8: **for** $d = j$ **do**
- 9: Update the position of search agent $X_{i,d}^{t+1}$ by Eq. (8)
- 10: **end for**
- 11: **end for**

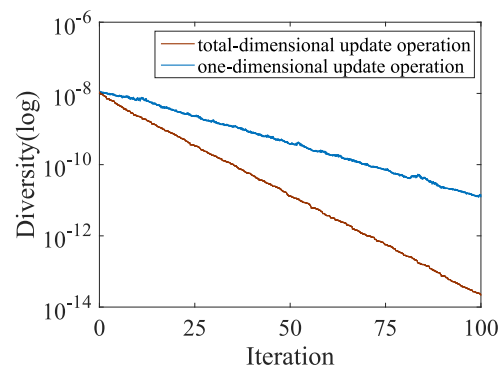


FIGURE 3. Comparison of one and total dimensional update operations on diversity variation.

operation on the Sphere function. It can be seen from the figure that the population diversity of one-dimensional renewal operation is higher than that of full-dimensional renewal operation, but the convergence speed is relatively slow. Algorithm 1 shows the one-dimensional update operation framework. When it is in the stage of the one-dimensional update operation, a dimension of the individual to be updated is randomly selected for the location update operation.

Finally, the adaptive conversion from one-dimensional update to traditional total-dimensional update was realized. At this stage, define the conversion factor (CF). CF is a logical value, and its initial value is 0, which represents a one-dimensional update operation. When CF is 1, it corresponds to a total-dimensional update operation. The conditions for triggering the change of the CF are: the population diversity is less than the predetermined value, and there is no change during five consecutive iterations. Algorithm 2 describes the framework of the proposed adaptive foraging strategy.

FIGURE 4 shows the population diversity curve with and without the proposed adaptive foraging strategy. The dotted line indicates that only the traditional total-dimensional update operation is used. The solid line indicates the one-dimensional and total-dimensional update operations using the adaptive cooperative foraging strategy. It can be seen from the figure that the total-dimensional update operation only

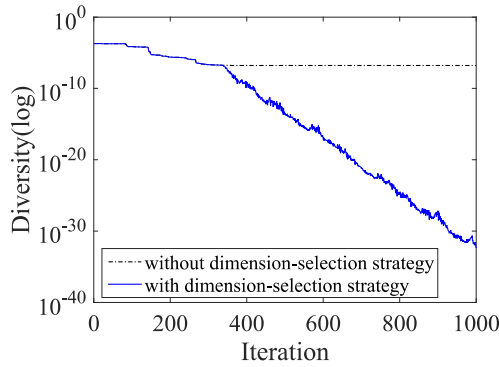


FIGURE 4. Diversity variation with or without the proposed adaptive cooperative foraging strategy.

focuses on the convergence of the algorithm and ignores the diversity of the population, thus causing premature convergence. At the beginning of the iteration, the proposed strategy maintains a high population diversity and maintains a high convergence rate in the later stage of the search. Besides, the phenomenon of turbulence appeared in the diversity in the later period, which enhanced the precision search capability.

B. DISPERSED FORAGING STRATEGY

The scarcity of food may force some Harris hawks to leave their current location, thereby effectively exploring more promising areas [52]. Based on this behavior, the dispersed foraging strategy was proposed. The process of dispersed foraging was determined by the dispersed factor ε . Only individuals who meet the dispersed conditions can perform position update operations. The position update equation during the dispersed foraging process is as follows:

$$X_i^{t+1} = X_i^t + \mu * \nabla_i^t * P_i^t \tag{9}$$

$$\nabla_i^t = (X_{r_1}^t - X_{r_2}^t) \tag{10}$$

where μ is the migration coefficient of the Harris hawks, $\mu \sim N(0.5, 0.1^2)$, and the parameter setting of μ is consistent with that in literature [34]. ∇_i^t represents the distance between any two search agents. n_1, n_2 represent random integers in $\{1, 2, \dots, N\}$, $n_1 \neq n_2 \neq i$. P_i^t is a logical value which is used to determine whether the Harris hawks implements the dispersed foraging strategy and be formulated as follows:

$$P_i^t = \begin{cases} 1, & r_5 > \varepsilon \\ 0, & r_5 \leq \varepsilon \end{cases} \tag{11}$$

where the dispersed factor ε is a parameter that decreases nonlinearly with iteration, and it is defined as follow.

$$\varepsilon = \varepsilon_0 e^{-\frac{t}{T}} \tag{12}$$

where ε_0 is constant 0.4, in formula (12), ε is a parameter that changes adaptively with the iteration. It selects some individuals to perform dispersed operations, while rest individuals remain in the original position. The advantage is that it is unnecessary for all individuals to explore unknown

Algorithm 3 Dispersed Foraging Strategy

Input: The search agent population X^t , parameter ε

Output: Updated search agents population X^{t+1}

- 1: Generate a logic matrix P_i^t by Eq. (11)
- 2: Generate the dispersed distance matrix ∇_i^t by Eq. (10)
- 3: Update the position of search agent X^{t+1} by Eq. (9)

areas, thus increasing the diversity of the population while retaining the exploitation of some areas. Therefore, the value of ε in the early stage is relatively large, and only a small part of the individuals perform the scattered foraging stage. These individuals help to improve the rate of convergence in the early stage. As the value of ε decreases, almost all individuals perform dispersed foraging, which is necessary to avoid local optimal. Algorithm 3 is the pseudo-code of dispersed foraging of ADHHO.

C. MODIFIED ESCAPE ENERGY

The escape energy E in HHO determines the transformation between exploration and exploitation [34]. If $|E| < 1$, the algorithm is in an exploitation stage. In formula (3), the parameter E decreases linearly from 2 to 0, which makes the value of $|E|$ completely less than 1 at the end of the iteration. In other words, the original HHO algorithm only performed local exploitation in the later stage, while completely ignored the global exploration.

In addition, prey takes action to avoid pursuit, which may increase its short-term survival probability. Some scholars have modelled the pursuit and escape behavior between predators and prey. According to the existing mathematical models, it can be concluded that the randomly shrinking exponential function is more suitable to simulate the energy changes of prey [53], [54].

Based on the above analysis, a new escape energy equation is proposed in this study, which is expressed as follows:

$$E = 2E_0 \times (2 \times rand \times e^{-(\delta \times \frac{t}{T})}) \tag{13}$$

where E represents the escape energy of the prey; E_0 represents the initial energy of the prey, and it is a random value between $[-1, 1]$; t and T are the current iteration number and the maximum iteration; δ is the attenuation factor.

In the formula (13), the attenuation factor δ represents the attenuation intensity of prey energy. The larger the δ value, the faster the escape energy E attenuates. The best δ value is obtained by rigorous experimental testing of the benchmark function. TABLE 2 shows some of the experimental results. It can be observed from TABLE 2 that the value of the attenuation factor δ changes from 1 to 2, and satisfactory performance is achieved for most benchmark functions when $\delta = 1.5$. It can also be observed from FIGURE 5 that when $\delta = 1.5$, the escape energy E can provide sufficient disturbance after 250 generations, which can be greater than 1 in some cases. This also shows that under a certain probability, the algorithm can maintain part of the exploration ability in

TABLE 2. The effect of attenuation factor (δ) on the performance of ADHHO algorithm on four benchmark functions.

Function	Parameter	$\delta=1.0$	$\delta=1.2$	$\delta=1.4$	$\delta=1.5$	$\delta=1.6$	$\delta=1.8$	$\delta=2.0$
F_3	Mean	1.64E-05	8.73E-10	7.84E-18	6.31E-72	3.56E-35	1.21E-21	5.98E-02
	Std	1.91E-04	3.91E-08	2.25E-17	6.39E-63	1.11E-20	1.14E-17	1.97E-02
F_8	Mean	4.97E+02	4.75E+01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.25E-03
	Std	3.49E+01	6.85E+01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	7.96E-04
F_{11}	Mean	6.86E-06	4.30E-11	1.92E-18	1.26E-34	6.89E-32	2.13E-14	9.84E-10
	Std	5.51E-08	9.97E-16	5.32E-15	5.93E-32	2.44E-26	1.04E-11	4.70E-13
F_{15}	Mean	3.78E+04	1.01E+03	9.97E+02	8.59E+02	9.20E+02	1.31E+03	3.15E+03
	Std	1.61E+04	6.18E+03	2.23E-06	4.94E+02	1.13E+04	1.01E+03	8.41E+03

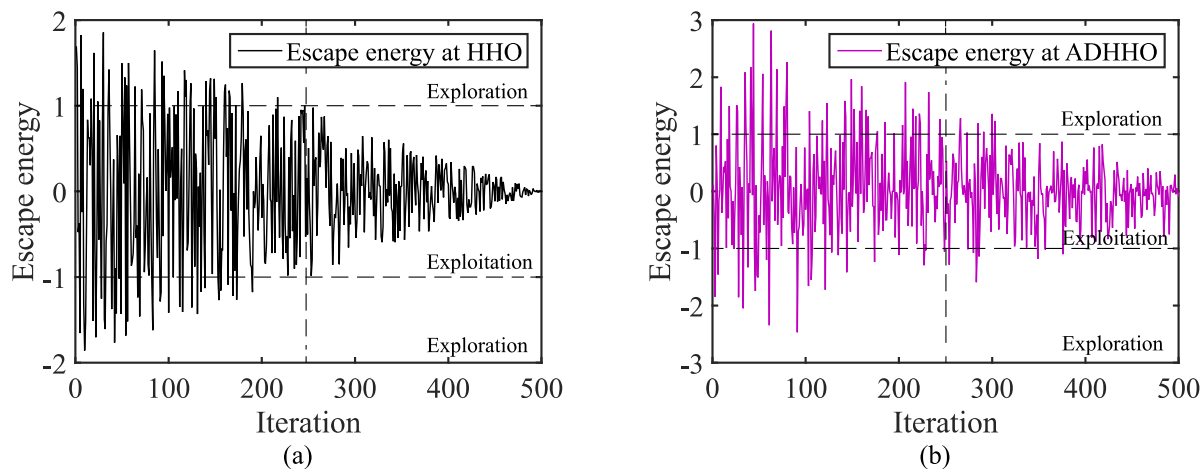


FIGURE 5. Comparison between original HHO and proposed ADHHO on escape energy, (a) The original escape energy at HHO, (b) The randomly shrinking exponential escape energy at ADHHO ($\delta = 1.5$).

the exploitation stage. Therefore, δ could help the algorithm to achieve a balance between exploration and exploitation and avoid falling into local optima.

D. ALGORITHM FLOW

The primary process of the ADHHO algorithm is shown in FIGURE 6. First, the exploration and exploitation phases were selected according to the escape energy of the prey in each iteration. The adaptive cooperation strategy was introduced into the exploration process, and the solution search equation is the modified location update equation (formula (8)). Then, one-dimensional or all-dimensional update operations were performed according to the conversion factor CF , which improved the diversity of Harris hawks' population and helps to find a better solution. Then, the fitness of the population were evaluated, and better individuals were selected to enter the dispersed foraging stage. The dispersed foraging strategy forces some individuals to leave the current optimal position to explore the target space further. This method prevented the algorithm from falling into local optima. In addition, as an auxiliary method, randomly shrinking exponential escape energy can not only ensure that the algorithm has strong exploration ability in the early iteration, but also retain the exploration capability in the later exploitation stage, which effectively balanced the exploration and exploitation of the algorithm. The pseudo-code of ADHHO algorithm is shown in Algorithm 4.

E. COMPUTATIONAL COMPLEXITY

To analyze the computational complexity of the proposed ADHHO algorithm and the original HHO algorithm, we gradually calculated the complexity of the two algorithms according to the worst-case complexity. In the initialization stage, after initializing the positions of N individuals, ADHHO and HHO had the same computational complexity $O(N)$. Then for the main loop of the HHO algorithm, the fitness of each search agent was first evaluated, so the computational complexity in this stage is $O(T \cdot N)$. Then in the population update stage, the position vector of each search agent was updated, so the computational complexity in this stage is $O(T \cdot N \cdot D)$. After comprehensive analysis, the computational complexity of HHO is calculated as follows:

$$O(\text{HHO}) = O(N) + O(T \cdot N) + O(T \cdot N \cdot D) = O(T \cdot N \cdot D) \tag{14}$$

In the main loop process of ADHHO, the proposed adaptive cooperative foraging strategy adaptively changed the one-dimensional update operation and the total-dimensional update operation according to the CF . In the one-dimensional update operation, only one dimension of the solution vector was changed. Therefore, the computational complexity is $O(T \cdot N)$. In the total-dimensional update operation, all dimensions of the solution vector were updated, so the computational complexity

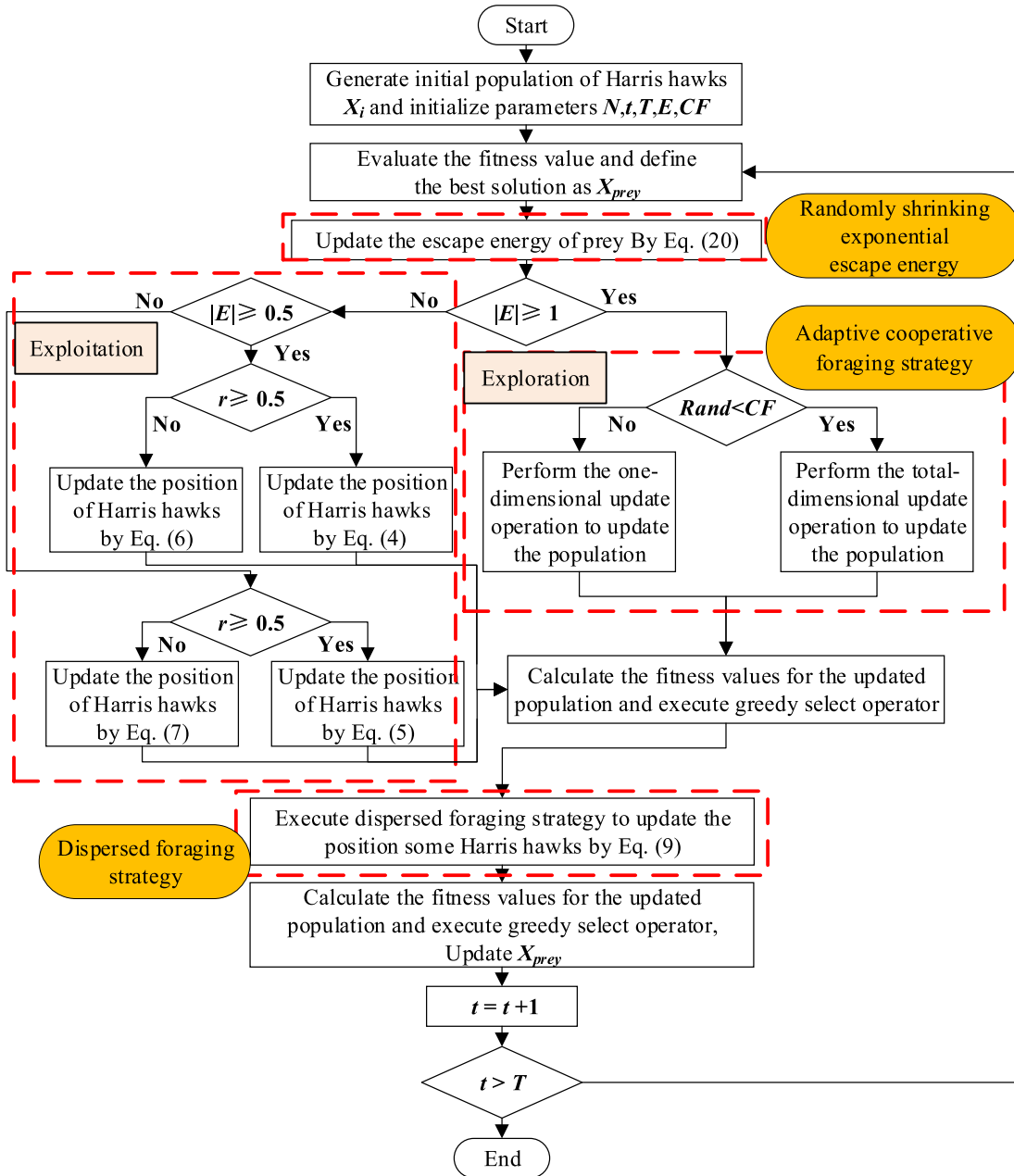


FIGURE 6. ADHHO algorithm flow chart.

is $O(T \cdot N \cdot D)$. According to the worst-case complexity rule, the computational complexity at this stage is $O(T \cdot N \cdot D)$. In the dispersed foraging strategy, the Harris hawk individuals that meet the decentralized conditions were allocated to other areas in the search space. The total number of individuals with position update does not change. Therefore, the computational complexity is $O(T \cdot N \cdot D)$. Comprehensive analysis, the computational complexity of ADHHO is calculated as follows:

$$O(\text{HHO}) = O(N) + O(T \cdot N \cdot D) + O(T \cdot N \cdot D) = O(T \cdot N \cdot D) \quad (15)$$

According to the above analysis, the ADHHO and HHO algorithms have the same computational complexity.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. BENCHMARK FUNCTION

In this study, 20 benchmark functions, including unimodal [55], multimodal [56] and CEC 2014 benchmark functions [57] were selected to evaluate the performance of ADHHO algorithm, and three experiments were carried out respectively. The expressions, dimensions, search ranges, and theoretical optimum of the benchmark functions are shown in TABLE 3. Among them, the unimodal benchmark

Algorithm 4 Pseudo-Code of ADHHO Algorithm

01: Initialize the number of iterations T , the number of hawks N , the effect of attenuation factor δ , the random jump strength J , conversion factor CF and the position of the hawks X_i ($i = 1, 2, \dots, N$)

02: Evaluate the fitness of each hawk then determine the locations of prey (best solution): X_{prey} .

03: **while** $Iter < T$ **do**

04: Check the boundary and calculate the fitness of each hawks X_i

$$E = 2E_0 \times (2 \times rand \times e^{-(\delta \times \frac{t}{T})}) \quad \% \text{update the } E \text{ using randomly shrinking exponential escape energy (Eq. (13))}$$

05: **for** $i = 1$ to N **do**

06: **if** $|E| \geq 1$ **then** % exploration stage

07: **if** $rand \leq CF$ **then** % adaptable cooperative foraging strategy

08: Generate a rand integer j_{rand} ($j_{rand} \in \{1, 2, \dots, D\}$)

09: $j = j_{rand}$ % one-dimensional position update operation

10: **else if**

11: $j = 1$ to D % total-dimensional position update operation

12: **end if**

13: **for** $d = j$ **do**

$$X_i^{t+1} = \begin{cases} Levy(D) (X_{prey}^t - X_i^t) + X_m^t & q \geq 0.5 \\ r_1 \frac{((X_{rand1}^t - X_i^t) + (X_{rand2}^t - X_i^t) + (X_{rand3}^t - X_i^t))}{3} & q < 0.5 \quad \% \text{update } X_i^{t+1} \text{ using Eq. (8)} \\ -r_2 |X_{rand4}^t - X_i^t| & \end{cases}$$

14: **end for**

15: **else if** ($|E| < 1$) % exploitation stage

16: **if** $r \geq 0.5$ and $|E| \geq 0.5$ **then** % soft besiege

17:

$$X_i^{t+1} = \Delta X_i^t - E |JX_{prey} - X_i^t| \quad \% \text{update the position of Harris hawks using Eq. (4)}$$

18: **else if** ($r \geq 0.5$ and $|E| < 0.5$) **then** % hard besiege

19:

$$X_i^{t+1} = X_{prey} - E |\Delta X_i^t| \quad \% \text{update the position of Harris hawks using Eq. (5)}$$

20: **else if** ($r < 0.5$ and $|E| \geq 0.5$) **then** % soft besiege with progressive rapid dives

21:

$$X_i^{t+1} = \begin{cases} Y = X_{prey} - E |JX_{prey} - X_i^t| & \text{if } f(Y) < f(X_i^t) \\ Z = Y + S \times Levy(d) & \text{if } f(Z) < f(X_i^t) \end{cases} \quad \% \text{update } X_i^{t+1} \text{ using Eq. (6)}$$

22: **else if** ($r < 0.5$ and $|E| < 0.5$) **then** % hard besiege with progressive rapid dives

$$X_i^{t+1} = \begin{cases} X_{prey} - E |JX_{prey} - X_m^t| & \text{if } f(Y) < f(X_i^t) \\ Z = Y + S \times Levy(d) & \text{if } f(Z) < f(X_i^t) \end{cases} \quad \% \text{update } X_i^{t+1} \text{ using Eq. (7)}$$

23: **end if**

24: **end if**

25: Generate a logic matrix P_i^t by Eq. (11) and the dispersed distance matrix ∇_i^t by Eq. (10)

$$X_i^{t+1} = X_i^t + \mu \times \nabla_i^t \times P_i^t \quad \% \text{update } X_i^{t+1} \text{ using dispersed foraging strategy (Eq. (9))}$$

26: Sort fitness and then update the locations of prey: X_{prey}

27: The location of prey X_{prey} is the final optimal solution

28: $Iter = Iter + 1$

29: **end for**

30: **End**

functions (F_1 - F_6) were selected to test the convergence rate and local exploitation ability of the algorithm. F_7 - F_{12} are multimodal benchmark functions and they contain multiple locally optimal solutions, and the number increases exponentially with the increase of dimension. Therefore, they were used to evaluate the exploration ability and global

optimization ability of the algorithm. F_{13} - F_{20} are modern numerical optimization problems proposed in IEEE CEC 2014 special session and competition on single-objective real parameter numerical optimization [57]. These benchmark functions have shifted, rotated, expanded and combined the most complicated mathematical optimization problems

TABLE 3. Description of the unimodal, multimodal and CEC 2014 benchmark functions.

Benchmark function	d	Range	f_{min}
$F_1(x) = \sum_{i=1}^n x_i^2$	50	[-100,100]	0
$F_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	50	[-10,10]	0
$F_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	50	[-100,100]	0
$F_4(x) = \max_i \{ x_i , 1 \leq i \leq n\}$	50	[-100,100]	0
$F_5(x) = \sum_{i=1}^n ([x_i + 0.5])^2$	50	[-100,100]	0
$F_6(x) = \sum_{i=1}^n ix_i^4 + random(0,1)$	50	[-1.28,1.28]	0
$F_7(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	50	[-500, 500]	-418.98 × Dim
$F_8(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	50	[-5.12, 5.12]	0
$F_9(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	50	[-32, 32]	0
$F_{10}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	50	[-600, 600]	0
$F_{11}(x) = \frac{\pi}{n} \{10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$	50	[-50, 50]	0
$y_i = 1 + \frac{x_i + 1}{4} u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$			
$F_{12}(x) = 0.1 \{ \sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	50	[-50, 50]	0
$F_{13}(x) =$ (CEC1: Rotated High Conditioned Elliptic Function)	50	[-100, 100]	100
$F_{14}(x) =$ (CEC2: Rotated Bent Cigar Function)	50	[-100, 100]	200
$F_{15}(x) =$ (CEC8: Shifted and Rastrigin's Function)	50	[-100, 100]	800
$F_{16}(x) =$ (CEC19: Hybrid Function 3 (N=4))	50	[-100, 100]	1900
$F_{17}(x) =$ (CEC20: Hybrid Function 4 (N=4))	50	[-100, 100]	2000
$F_{18}(x) =$ (CEC21: Hybrid Function 5 (N=5))	50	[-100, 100]	2100
$F_{19}(x) =$ (CEC24: Composition Function 2 (N=3))	50	[-100, 100]	2400
$F_{20}(x) =$ (CEC25: Composition Function 3 (N=3))	50	[-100, 100]	2500

TABLE 4. Parameter setting of algorithms.

Algorithm	Parameter setting
LMHHO	Population size $N=50$, iterations $T=1000/6000$, random jump strength $J \in [0, 2]$, long-term memory limit $ML = 10$
DHHO/M	Population size $N=50$, iterations $T=1000/6000$, random jump strength $J \in [0, 2]$, disturbance peak constant $\alpha = 2.5$, scaling factor $SF = 0.5$
HHO	Population size $N=50$, iterations $T=1000/6000$, random jump strength $J \in [0, 2]$
PSO	Population size $N=50$, iterations $T=1000/6000$, learning factors $C_1 = C_2 = 2$, inertia weight $w = 0.9$
FF	Population size $N=50$, iterations $T=1000/6000$, step factor of disturbance $\alpha = 0.25$, attractiveness $\beta = 0.20$, light absorption coefficient $\gamma = 1$
ADHHO	Population size $N=50$, iterations $T=1000/6000$, random jump strength $J \in [0, 2]$, conversion factor $CF = 0$, attenuation factor $\delta = 1.5$

presented in the literature. Therefore, they are usually used to evaluate the comprehensive abilities of an algorithm.

B. PARAMETER SETTING

In each experimental test, the performance of the improved ADHHO algorithm was compared with classical SI algorithm and other improved versions of HHO, including the original HHO [34], PSO [11], FF [20], Long-term memory Harris hawks optimization (LMHHO) [41] and Dynamic Harris

hawks optimization with mutation mechanism (DHHO/M) [35]. TABLE 4 lists the parameter settings of all algorithms. For the fairness of the experiment, the maximum iterations (T) and population size (N) were set to 1000 and 50, respectively, in the first two experiments (benchmark problems). In the third experiment (CEC 2014 problems), the maximum iterations (T) was set to 6000 to obtain 300000 number of functional evaluations (NFEs). For each benchmark function, all algorithms were run 30 times independently. In addition,

TABLE 5. Statistical results obtained by PSO, FF, HHO, DHHO/M, LMHHO and ADHHO through 30 independent runs on functions $F_1 - F_6$ with unimodal benchmark functions.

<i>Fun</i>		PSO	FF	HHO	DHHO/M	LMHHO	ADHHO
F_1	Best	2.00E-33	1.72E-15	1.37E-83	1.73E-56	6.18E-137	7.23E-182
	Worst	2.78E-31	1.98E-14	1.51E-82	2.10E-54	6.66E-134	8.48E-180
	Mean	3.56E-31	2.23E-14	1.64E-82	2.46E-55	7.15E-136	9.73E-180
	Std	1.11E-23	3.64E-15	1.91E-80	5.13E-55	6.87E-131	1.77E-179
F_2	Best	1.20E-11	7.54E-16	1.87E-84	6.32E-41	1.49E-38	3.95E-133
	Worst	1.96E-10	9.66E-14	2.09E-82	1.00E-40	6.21E-31	4.39E-132
	Mean	2.71E-10	1.18E-15	4.12E-83	1.37E-40	1.24E-32	4.84E-132
	Std	1.07E-09	3.01E-11	2.69E-62	5.22E-32	8.78E-27	6.26E-123
F_3	Best	3.28E-14	4.97E-19	4.97E-24	1.02E-27	3.40E-38	3.08E-85
	Worst	3.12E-12	4.97E-19	4.99E-23	1.27E-26	3.29E-35	3.01E-83
	Mean	2.97E-13	4.97E-19	5.46E-23	1.52E-26	3.18E-37	2.94E-84
	Std	2.23E-10	0.00E+00	1.47E-21	3.54E-27	1.54E-28	9.97E-68
F_4	Best	6.08E-04	3.45E-06	3.28E-10	3.20E-11	3.08E-11	2.67E-19
	Worst	6.47E-04	5.17E-06	3.12E-10	6.52E-09	3.01E-10	2.29E-17
	Mean	6.86E-04	6.89E-06	2.97E-10	9.84E-10	2.94E-10	1.92E-18
	Std	5.51E-06	2.44E-07	2.23E-06	4.70E-13	9.97E-16	5.32E-13
F_5	Best	1.76E+00	4.97E+00	1.86E-04	9.95E-01	9.84E-04	9.95E-04
	Worst	2.71E+00	4.97E+00	2.62E-04	3.48E+00	5.22E-04	3.48E-04
	Mean	3.65E+00	4.97E+00	3.38E-04	5.97E+00	1.03E-04	5.97E-04
	Std	1.33E-02	7.24E-02	1.07E-03	3.52E+00	3.52E-12	0.00E+00
F_6	Best	1.02E-02	6.08E-01	3.28E-03	3.20E-03	6.08E-04	9.90E-03
	Worst	1.27E-02	6.47E-01	3.12E-03	6.52E-03	6.47E-04	1.15E-03
	Mean	1.52E-02	6.86E-01	2.97E-03	9.84E-03	6.86E-04	1.51E-03
	Std	3.54E-01	5.51E-03	2.23E-04	4.70E-05	5.51E-07	5.02E-15

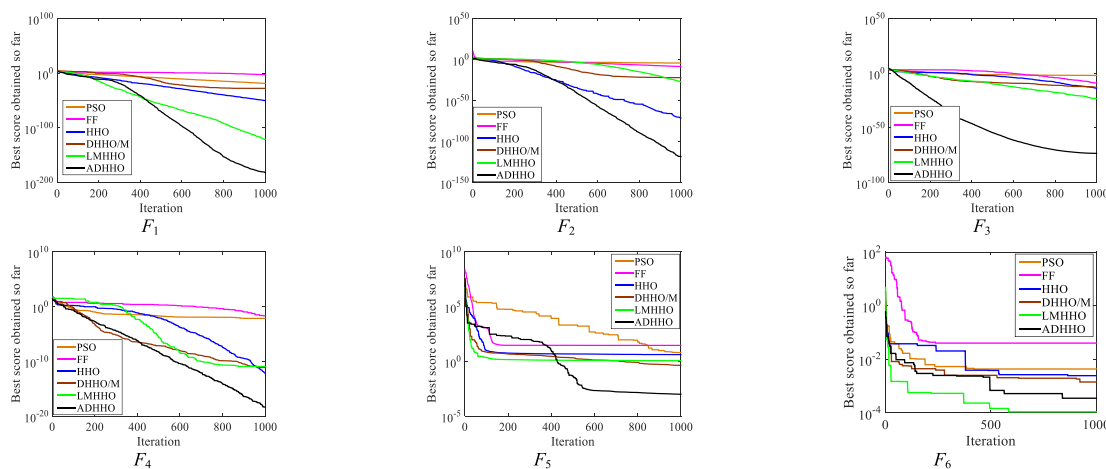


FIGURE 7. Comparison of the convergence curves of ADHHO and the other algorithms on functions $F_1 - F_6$.

all the algorithms were implemented in MATLAB 2014a. All computations were run on a CPU: Intel Core i5-4200 M, 2.5 GHz, 8G RAM, and Windows 7 (64-bit) operating system.

C. EXPERIMENTAL RESULTS

1) EXPERIMENTAL TEST 1: UNIMODAL BENCHMARK FUNCTIONS

Unimodal functions have only one global optimum, and they are often used to evaluate the convergence rate and exploitation ability of algorithms. It can be seen from the TABLE 5 that the proposed ADHHO algorithm can achieve better results than other algorithms on most unimodal functions. For example, on F_1, F_2, F_3, F_4 and F_6 , the average and standard deviation of ADHHO are better than LMHHO, DHHO/M, PSO and FF. For F_5 , the performance of ADHHO algorithm

is the same as that of LMHHO and HHO, but much better than other algorithms. FIGURE 7 shows the comparison of the convergence rates of the six algorithms. From FIGURE 7, the ADHHO algorithm has a faster convergence rate than other algorithms for most unimodal functions. Although the search steps of LMHHO and DHHO/M have been improved compared with HHO, they still can not exceed the proposed ADHHO. According to the characteristics of unimodal function, it can be proved that ADHHO has good exploitation ability.

2) EXPERIMENTAL TEST 2: MULTIMODAL BENCHMARK FUNCTIONS

The multimodal benchmark function contains multiple local minimum points and is often used to evaluate the exploratory

TABLE 6. Statistical results obtained by PSO, FF, HHO, DHHO/M, LMHHO and ADHHO through 30 independent runs on functions F_7 - F_{12} with multimodal benchmark functions.

<i>Fun</i>		PSO	FF	HHO	DHHO/M	LMHHO	ADHHO
F_7	Best	-1.62E+04	-1.46E+04	-1.71E+04	-1.68E+04	-1.61E+04	-2.09E+04
	Worst	-1.57E+04	-1.43E+04	-1.69E+04	-1.57E+04	-1.58E+04	-2.09E+04
	Mean	-1.61E+04	-1.45E+04	-1.70E+04	-1.65E+04	-1.60E+04	-2.09E+04
	Std	1.01E+04	4.73E+03	8.23E+02	2.44E+02	3.84E+02	1.03E+02
F_8	Best	4.71E+01	3.13E+00	5.11E+00	7.23E+00	0.00E+00	0.00E+00
	Worst	4.76E+01	3.16E+00	5.14E+00	7.54E+00	0.00E+00	0.00E+00
	Mean	4.75E+01	3.15E+00	5.13E+00	7.16E+00	0.00E+00	0.00E+00
	Std	6.85E+01	3.84E-02	1.20E-02	4.30E-05	0.00E+00	0.00E+00
F_9	Best	1.94E+00	3.01E-03	3.74E-07	3.28E-09	1.86E-15	5.05E-18
	Worst	2.08E+00	2.66E-03	4.08E-06	3.12E-08	2.62E-15	8.01E-18
	Mean	2.23E+00	2.31E-03	4.43E-07	2.97E-09	3.38E-15	1.55E-18
	Std	2.04E-02	4.89E-03	4.85E-05	2.23E-05	1.07E-09	1.06E-30
F_{10}	Best	1.45E-02	7.54E-02	4.34E-03	5.98E-02	3.65E-02	4.42E-03
	Worst	6.22E-01	9.66E-01	2.15E-02	1.97E-02	4.33E-02	4.56E-03
	Mean	1.64E-02	1.18E-02	4.29E-03	3.89E-02	8.29E-02	4.69E-03
	Std	8.79E-02	3.01E-05	3.04E-02	2.71E-01	5.61E-01	1.92E-02
F_{11}	Best	3.64E-12	7.54E-05	5.98E-08	4.34E-16	4.42E-38	3.65E-18
	Worst	6.22E-10	1.18E-04	9.89E-07	2.15E-15	4.56E-36	4.33E-19
	Mean	1.24E-11	9.66E-05	1.97E-07	4.29E-16	4.69E-38	8.29E-18
	Std	8.79E-10	3.01E-05	2.71E-06	3.04E-11	1.92E-33	5.61E-20
F_{12}	Best	1.37E-06	8.73E-10	7.84E-18	6.18E-04	2.00E-35	1.66E-36
	Worst	1.51E-05	3.91E-08	2.25E-17	6.66E-04	2.78E-34	2.36E-35
	Mean	1.64E-05	7.73E-08	3.71E-18	1.59E-04	3.56E-35	4.71E-35
	Std	1.91E-04	5.41E-08	2.07E-10	6.87E-03	1.11E-20	3.33E-32

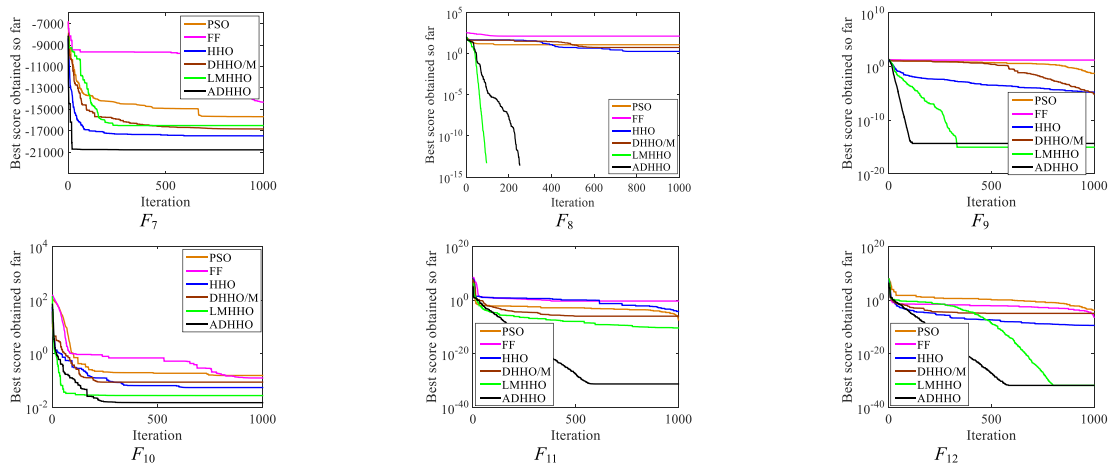


FIGURE 8. Comparison of the convergence curves of ADHHO and the other algorithms on functions F_7 - F_{12} .

ability of the algorithm. It can be observed from TABLE 6 that the mean and standard deviation of ADHHO on F_7 , F_8 , F_{10} , and F_{12} are obviously better than other algorithms. It is worth noting that the ADHHO algorithm can obtain the global optimum on F_7 and F_8 , while other algorithms cannot get the global optimum. On F_{11} , the ADHHO algorithm ranks second among all algorithms. FIGURE 8 shows the convergence curves of all algorithms for solving high-dimensional multimodal functions. From this figure, compared with the other six algorithms, ADHHO has better convergence rate and sufficient accuracy. The LMHHO algorithm only defeats ADHHO on F_9 . The experiment shows that the ADHHO algorithm still has an excellent performance in exploration ability, which is attributed to the high population diversity

provided by the cooperative relationship of Harris hawks and dispersed foraging behavior.

3) EXPERIMENTAL TEST 3: CEC 2014 BENCHMARK FUNCTIONS

In this experiment, the CEC 2014 benchmark functions are selected to evaluate the comprehensive performance of ADHHO. From TABLE 7, it can be seen that among all the 8 CEC 2014 test functions, the performance of ADHHO on seven functions is better than that of other algorithms. The average value of ADHHO is second only to LMHHO algorithm on F_{14} . Also, as can be seen from FIGURE 9, ADHHO has the best convergence performance among all

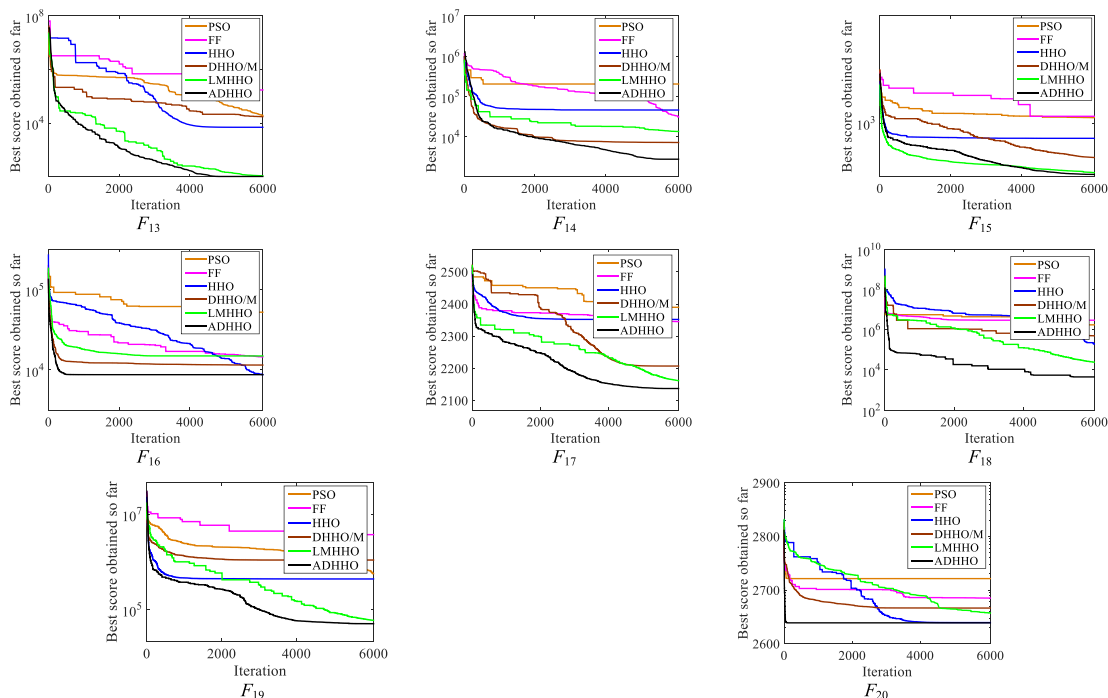


FIGURE 9. Comparison of the convergence curves of ADHHO and the other algorithms on functions F_{13} - F_{20} .

algorithms. Through these comparisons, it is proved that ADHHO algorithm is still the best algorithm to solve these complex optimization problems. The modified escape energy equation effectively balances the global search and local search of the ADHHO algorithm. Therefore, ADHHO shows stability and effectiveness in such a complex benchmark function.

4) STATISTICAL ANALYSIS

In this section, Wilcoxon’s rank-sum test [58], a popular non-parametric statistical test method was applied to the statistics of the experimental results to prove the significance performance of ADHHO. The P obtained by the Wilcoxon’s rank-sum test are recorded in TABLE 8. In this table, the ‘+’ sign indicates that the reference algorithm has a better performance than the compared one, and ‘-’ signs indicate that the reference algorithm is worse than the compared algorithm. The value on the last line indicates the number of ‘+’ and ‘-’ symbols. It corresponds to the number of P under a 95% confidence interval ($\alpha = 0.05$). As can be seen from the table, ADHHO has more ‘+’ signs than the other algorithms, which proves that ADHHO has a significant difference from the other five algorithms. As a result, the ADHHO has a better performance than the other algorithms.

Finally, the mean absolute error (MAE) of various algorithms are calculated and sorted to analyze the performance of all algorithm. MAE is an effective statistical method to show the gap between the result and the actual value. The MAE formula is expressed as follows:

$$MAE = \frac{\sum_{i=1}^N |o_i - m_i|}{N} \tag{16}$$

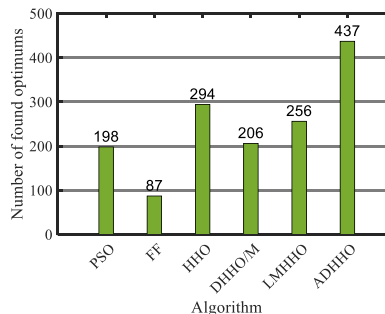


FIGURE 10. Comparison of the algorithms to obtain the number of global optimal solutions in 600 runs.

where o_i represents the actual result of an algorithm, m_i is the theoretical optimum of a benchmark function, and N is the number of samples.

TABLE 9 lists the MAE and ranking of all algorithms. From this table, ADHHO provides the best performance compared to the other five algorithms. Besides, FIGURE 10 shows the comparison results of 600 runs of each algorithm (30 runs for each benchmark function), and the number of times that ADHHO reaches the optimal solution is 437. This once again proves the excellent performance of ADHHO algorithm.

5) PERFORMANCE ANALYSIS OF IMPROVED STRATEGIES

This paper proposes three strategies to improve the HHO algorithm. They are adaptive cooperative foraging strategy, dispersed foraging strategy, and randomly shrinking exponential escape energy strategy. This section mainly discusses

TABLE 7. Statistical results obtained by PSO, FF, HHO, DHHO/M, LMHHO and ADHHO through 30 independent runs on functions F_{13} - F_{20} with the CEC 2014 benchmark functions.

<i>Fun</i>		PSO	FF	HHO	DHHO/M	LMHHO	ADHHO
F_{13}	Best	2.31E+04	5.15E+04	7.30E+03	9.98E+02	5.61E+02	3.10E+02
	Worst	4.02E+05	5.62E+04	2.55E+03	5.90E+03	9.26E+03	1.75E+03
	Mean	1.25E+04	2.13E+05	1.01E+04	2.90E+03	1.78E+03	8.19E+02
	Std	7.96E+05	1.04E+04	6.18E+08	1.13E+04	1.61E+03	4.02E+02
F_{14}	Best	5.32E+06	4.47E+05	3.22E+05	5.46E+03	2.16E+02	1.82E+03
	Worst	6.12E+06	3.55E+06	9.28E+05	3.90E+04	2.82E+04	2.90E+04
	Mean	5.76E+06	1.50E+05	6.63E+05	1.30E+04	1.00E+04	1.79E+04
	Std	1.96E+05	8.46E+06	1.52E+05	9.02E+03	9.83E+03	1.06E+04
F_{15}	Best	9.67E+02	6.06E+03	8.68E+02	4.63E+03	4.04E+03	8.00E+02
	Worst	4.73E+03	2.52E+04	9.04E+03	5.64E+03	5.58E+03	9.44E+03
	Mean	1.64E+03	1.26E+04	8.77E+02	4.96E+03	4.97E+03	8.57E+02
	Std	9.11E+02	4.94E+03	8.87E+00	3.12E+01	3.49E+04	3.99E+02
F_{16}	Best	1.14E+05	1.66E+06	1.07E+04	2.47E+04	2.64E+05	7.05E+03
	Worst	4.38E+06	1.66E+08	4.36E+05	4.98E+05	3.16E+06	6.01E+04
	Mean	9.87E+05	5.04E+07	1.24E+05	1.84E+05	1.32E+06	2.62E+04
	Std	1.03E+06	4.55E+07	9.58E+04	1.29E+05	7.19E+05	1.52E+04
F_{17}	Best	1.58E+05	3.59E+04	1.06E+04	1.46E+04	1.33E+04	1.17E+04
	Worst	1.70E+08	2.33E+05	5.46E+05	3.25E+06	2.03E+05	8.45E+04
	Mean	2.18E+07	1.03E+05	1.12E+05	1.84E+05	6.04E+04	4.05E+04
	Std	3.82E+07	5.00E+04	1.27E+05	5.74E+05	3.85E+04	1.63E+04
F_{18}	Best	4.47E+04	1.56E+05	5.43E+04	8.84E+04	9.18E+04	9.70E+04
	Worst	3.37E+07	3.21E+06	8.70E+06	1.72E+06	3.43E+06	1.22E+06
	Mean	1.99E+06	1.37E+06	2.01E+06	6.58E+05	7.70E+05	4.46E+05
	Std	5.93E+06	8.17E+05	1.83E+06	4.15E+05	7.81E+05	2.88E+05
F_{19}	Best	2.64E+03	2.82E+03	2.62E+03	2.62E+03	2.62E+03	2.50E+03
	Worst	2.74E+03	3.88E+03	2.62E+03	2.62E+03	2.62E+03	2.50E+03
	Mean	2.68E+03	3.06E+03	2.62E+03	2.62E+03	2.62E+03	2.50E+03
	Std	2.76E+01	2.24E+02	2.92E-02	1.20E-01	1.78E+00	8.41E-10
F_{20}	Best	2.64E+03	2.71E+03	2.60E+03	2.60E+03	2.60E+03	2.60E+03
	Worst	2.70E+03	2.86E+03	2.63E+03	2.64E+03	2.63E+03	2.60E+03
	Mean	2.66E+03	2.78E+03	2.61E+03	2.62E+03	2.62E+03	2.60E+03
	Std	1.31E+01	3.70E+01	1.10E+01	1.37E+01	6.51E+00	8.37E-02

TABLE 8. Results of Wilcoxon’s test for ADHHO against the other five algorithms for 20 benchmark functions.

<i>Fun</i>	PSO vs ADHHO		FF vs ADHHO		HHO vs ADHHO		DHHO/M vs ADHHO		LMHHO vs ADHHO	
	<i>p</i> -value	win	<i>p</i> -value	win	<i>p</i> -value	win	<i>p</i> -value	win	<i>p</i> -value	win
F_1	1.73E-19	+	1.73E-19	+	1.73E-19	+	1.73E-19	+	1.73E-19	+
F_2	4.95E-14	+	1.69E-17	+	1.73E-19	+	1.73E-19	+	1.73E-19	+
F_3	7.87E-05	+	1.69E-17	+	3.55E-14	+	1.86E-08	+	1.06E-07	+
F_4	1.74E-10	+	1.69E-17	+	1.73E-19	+	1.73E-19	+	1.73E-19	+
F_5	1.73E-19	+	1.69E-17	+	1.73E-19	+	1.73E-19	+	1.73E-19	+
F_6	1.73E-19	+	1.69E-17	+	1.73E-19	+	1.73E-19	+	1.73E-19	+
F_7	4.99E-01	-	1.69E-17	+	1.73E-19	+	1.73E-19	+	1.73E-19	+
F_8	7.62E-14	+	1.69E-17	+	1.73E-19	+	1.73E-19	+	1.73E-19	+
F_9	2.24E-06	+	3.56E-09	+	5.04E-01	-	8.20E-04	+	5.32E-02	+
F_{10}	1.73E-19	+	1.73E-19	+	1.73E-19	+	1.73E-19	+	1.73E-19	+
F_{11}	6.44E-01	-	5.96E-08	+	9.16E-08	+	9.36E-01	-	5.43E-04	+
F_{12}	1.73E-19	+	1.73E-19	+	1.00E+00	-	1.24E-04	+	1.00E+00	-
F_{13}	1.73E-19	+	1.73E-19	+	1.69E-17	+	1.73E-19	+	1.73E-19	+
F_{14}	1.69E-17	+	1.73E-19	+	1.69E-17	+	1.73E-19	+	1.73E-19	+
F_{15}	5.96E-08	+	1.73E-19	+	9.30E-11	+	1.24E-04	+	1.73E-19	+
F_{16}	1.69E-17	+	1.73E-19	+	5.96E-08	+	1.69E-17	+	5.96E-08	+
F_{17}	1.73E-19	+	1.69E-17	+	6.33E-01	-	8.20E-04	+	7.50E-09	+
F_{18}	1.73E-19	+	1.69E-17	+	5.91E-02	+	1.42E-02	+	4.65E-07	+
F_{19}	1.73E-19	+	6.13E-01	-	9.30E-11	+	1.69E-17	+	5.96E-08	+
F_{20}	1.73E-19	+	5.96E-08	+	9.30E-11	+	1.69E-17	+	5.32E-02	+
+/-		18/2		19/1		17/3		19/1		19/1

the impact of three strategies on algorithm performance. Since the third strategy is the auxiliary strategy of the first two strategies, in short, the third strategy is a part of the first two

strategies. Therefore, we no longer compare the third strategy separately. Based on this, the improved algorithm (ADHHO), the HHO that only introduces adaptive cooperative foraging

TABLE 9. Average ranking of the six algorithms using MAE for 24 benchmark functions.

Algorithm	MAE	Rank
ADHHO	1.1815E+02	1
LMHHO	4.3472E+03	2
HHO	2.1562E+04	3
DHHO/M	1.9748E+04	4
PSO	2.5696E+05	5
FF	8.2977E+06	6

strategy (ADHHO), the HHO that only introduces dispersed foraging strategy (DHHO), and the original HHO algorithm were selected for comparison experiments. The parameter settings of each algorithm are shown in TABLE 4. Selected functions F_1 - F_{12} from TABLE 3 for experimental simulation. Each algorithm runs independently 30 times on each function. The experimental results are shown in TABLE 10, and the results with the best performance are shown in bold. From this table, the performance of ADHHO algorithm on functions F_1 , F_2 , F_3 , F_4 , F_7 , and F_9 is significantly better than the other three algorithms. For F_8 , ADHHO and AHHO obtained the same performance. For F_5 , F_6 , and F_{11} , ADHHO performs worse than AHHO but ranks second among all algorithms. It can also be seen from the convergence curves of the partial functions shown in FIGURE 11 that ADHHO is worse than AHHO only on F_6 . For F_8 , there is no apparent difference between ADHHO and AHHO in optimization accuracy. For F_{10} , the convergence rate of ADHHO is inferior to HHO. For other functions, ADHHO has the best performance. Combined with average accuracy and stability performance, it can be seen that ADHHO has better global convergence ability and robust performance than AHHO, DHHO, and HHO. The experimental results show that adaptive cooperative foraging strategy and dispersed foraging strategy have a synergistic effect in improving the performance of HHO, and verify the effectiveness of the multi-strategy integration algorithm proposed in this paper.

V. ADHHO FOR PRESSURE VESSEL DESIGN PROBLEM

In recent years, stochastic optimization techniques to solve structural design problems have become a research hotspot in structural design [59], [60]. To further verify the effectiveness of ADHHO in structural design problems, this section optimizes pressure vessels' parameters. The pressure vessel design problem's goal is to minimize the total cost of materials, forming, and welding of cylindrical vessels. The pressure vessel and parameters of this study are shown in FIGURE 12. In this figure, T_h represents the head's thickness, T_s represents the thickness of the housing, and L represents the length of the cylindrical cross-section without considering the head. R represents the inner diameter. The pressure vessel design problem is constrained by the above four parameters, and its mathematical model is as follows:

Consider $\vec{x} = [x_1 \ x_2 \ x_3 \ x_4] = [T_s \ T_h \ R \ L]$,
 Minimize $f(\vec{x}) = 0.6224 \ x_1 x_3 x_4 + 1.7781 \ x_2 x_3^2$

TABLE 10. Simulation comparison of improved strategies.

Fun		HHO	AHHO	DHHO	ADHHO
F_1	Best	3.31E-79	5.41E-112	5.15E-88	9.40E-186
	Worst	4.02E-75	7.43E-106	5.62E-85	2.17E-183
	Mean	4.15E-76	2.12E-110	2.13E-87	3.50E-181
	Std	5.96E-73	1.18E-102	1.04E-80	2.86E-180
F_2	Best	4.32E-80	1.74E-113	4.47E-93	1.94E-136
	Worst	6.12E-76	5.73E-108	3.55E-90	2.52E-132
	Mean	5.76E-79	2.35E-112	1.50E-91	6.14E-133
	Std	1.96E-79	2.11E-110	8.46E-89	2.95E-133
F_3	Best	9.67E-22	3.68E-63	6.06E-51	1.62E-83
	Worst	4.73E-19	1.59E-60	2.52E-48	1.11E-80
	Mean	1.64E-20	5.28E-62	1.26E-50	7.35E-82
	Std	9.11E-20	4.68E-59	4.94E-50	1.94E-81
F_4	Best	1.14E-11	3.17E-18	1.66E-13	1.57E-21
	Worst	4.38E-10	7.61E-15	1.66E-11	5.98E-18
	Mean	1.87E-10	2.38E-16	5.04E-12	2.68E-20
	Std	1.03E-10	3.66E-12	4.55E-10	2.58E-16
F_5	Best	9.55E-04	5.29E-06	3.59E-05	7.31E-04
	Worst	7.05E-03	9.81E-05	2.33E-04	1.94E-03
	Mean	3.28E-03	3.28E-05	1.03E-05	8.42E-04
	Std	3.00E-01	4.55E-05	5.00E-03	1.48E-02
F_6	Best	4.23E-04	5.73E-05	1.56E-03	2.36E-03
	Worst	5.15E-03	2.34E-04	3.21E-01	1.49E-01
	Mean	3.06E-03	2.00E-04	1.37E-02	7.72E-02
	Std	5.93E-01	1.04E-03	8.17E-00	1.01E-01
F_7	Best	2.64E-03	1.70E-04	5.43E-03	2.19E-05
	Worst	2.74E-02	2.18E-03	8.70E-01	2.33E-04
	Mean	1.68E-02	1.82E-03	2.01E-02	8.68E-05
	Std	2.76E-02	4.47E-02	1.83E-01	1.94E-03
F_8	Best	2.64E-01	0.00E-00	2.71E-03	0.00E-00
	Worst	1.84E-00	0.00E-00	2.86E-01	0.00E-00
	Mean	6.54E-01	0.00E-00	2.78E-02	0.00E-00
	Std	3.07E-00	0.00E-00	9.11E-01	0.00E-00
F_9	Best	4.74E-06	1.37E-14	1.14E-03	1.06E-16
	Worst	2.70E-05	6.78E-12	4.38E-02	2.24E-15
	Mean	2.66E-05	1.70E-12	9.87E-03	3.76E-16
	Std	1.31E-05	1.10E-09	2.64E-01	1.37E-16
F_{10}	Best	2.64E-04	3.18E-04	4.70E-03	1.31E-04
	Worst	8.70E-03	6.63E-03	8.66E-02	2.64E-03
	Mean	2.66E-03	2.61E-03	1.31E-02	2.70E-04
	Std	1.31E-01	1.10E-02	2.64E-02	1.37E-04
F_{11}	Best	2.64E-09	1.24E-20	2.71E-13	2.66E-17
	Worst	2.70E-07	2.12E-18	5.86E-11	9.31E-16
	Mean	4.26E-08	2.36E-19	2.78E-11	2.64E-16
	Std	1.31E-07	1.10E-16	3.70E-10	2.70E-15
F_{12}	Best	2.64E-19	3.58E-39	1.71E-35	5.24E-40
	Worst	7.70E-18	5.21E-37	2.86E-34	2.78E-39
	Mean	2.66E-18	2.61E-38	2.78E-35	3.70E-38
	Std	1.31E-16	1.10E-36	3.70E-31	2.71E-32

+ 3.1661x₁²x₄ + 19.84x₁²x₃
 Subject to $g_1(\vec{x}) = -x_1 + 0.0193x_3 \leq 0$,
 $g_2(\vec{x}) = -x_2 + 0.00954 \ x_3 \leq 0$,

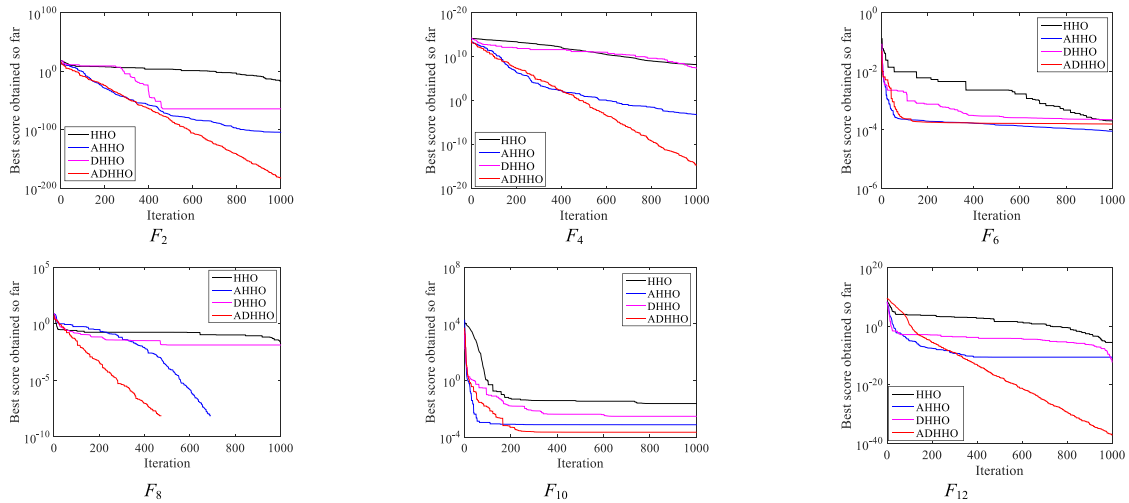


FIGURE 11. Comparison of the convergence curves of improved strategies on some benchmark functions.

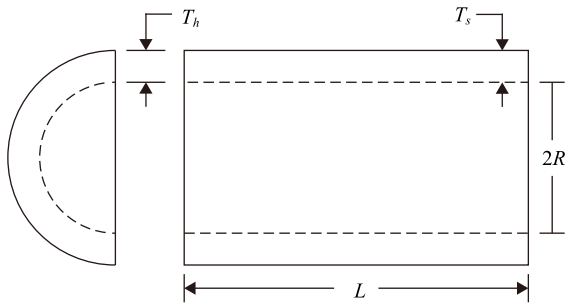


FIGURE 12. Pressure vessels and parameters.

TABLE 11. Parameters comparison results of pressure vessel design problems.

Algorithms	T_s	T_h	R	L	Cost
PSO	1.24014	0.70245	64.17954	14.7153	7903.73
FF	1.32865	0.65601	69.20148	53.89452	11395.15
HHO	1.26154	0.62125	65.2120	10.05178	7322.12
DHHO/M	0.91354	0.48756	47.16417	122.6145	6321.56
LMHHO	1.02971	0.49872	53.72954	71.24853	6382.10
ADHHO	0.87015	0.43114	45.01254	143.5317	6072.56

$$g_3(\vec{x}) = -\pi x_3^2 x_4 - 4/3\pi x_3^3 + 1296000 \leq 0,$$

$$g_4(\vec{x}) = x_4 - 240 \leq 0,$$

Variable range: $0 \leq x_1 \leq 99$; $0 \leq x_2 \leq 99$;
 $10 \leq x_3 \leq 200$; $10 \leq x_4 \leq 200$.

The algorithms used to test this type of problem are the same as the algorithms in Section IV. The parameter settings of the algorithms are shown in TABLE 4. The optimal solutions and corresponding variable results are shown in TABLE 11. From this table, the optimization result of ADHHO is the best among all algorithms. Besides, to test the stability of the algorithm, the experimental results of

TABLE 12. Performance comparison results of pressure vessel design problems.

Algorithms	Best	Worst	Mean	Std
PSO	7061.542	91042.63	23145.702	41521.215
FF	5904.53	11584.21	8074.26	1873.518
HHO	6011.741	7854.428	6845.216	510.183
DHHO/M	6013.253	7748.752	6483.156	482.271
LMHHO	6143.842	6617.583	6548.532	147.5279
ADHHO	5908.4511	6157.4825	6061.5283	86.4125

each algorithm running independently 30 times are shown in TABLE 12. It can be seen from the table that ADHHO obtains better mean and variance than other algorithms, and the robustness of this algorithm is superior to other algorithms. Since this structural design problem’s search space is unknown, these results provide strong evidence for the adaptability of ADHHO in solving real-world problems.

VI. CONCLUSION

This paper proposed three strategies to improve the global search performance of the HHO algorithm. First, an adaptive cooperative foraging strategy was introduced, and some Harris hawk individuals were randomly selected for cooperative foraging. Simultaneously, the method was embedded in the one-dimensional update operation, and the one-dimensional cooperative foraging and traditional total-dimensional cooperative foraging was adaptively selected. Second, a dispersed foraging strategy was introduced, forcing some Harris hawks to scatter into different search spaces to find potential prey. Finally, a randomly shrinking exponential function was used to simulate the energy decay process of the prey. Twenty benchmark functions, including unimodal, multimodal, and CEC 2014 benchmark functions, were used to test the proposed ADHHO algorithm’s robustness. The algorithm’s performance was compared with other advanced swarm

intelligence algorithms and other improved variants of the HHO algorithm. Wilcoxon's rank-sum test and MAE were used to analyze the experimental results. Statistical results show that ADHHO outperforms HHO and other advanced algorithms in solution quality, stability, and local optimum avoidance. Besides, this paper also applied the ADHHO algorithm to the design of pressure vessels. Experimental results have shown that ADHHO has a good performance in solving real-world optimization problems of unknown search spaces.

However, on some convergence curves, it can be seen that the convergence speed of ADHHO is slow. Therefore, in future research, it is necessary to study the convergence rate of the algorithm further. Also, this work only provides the basic framework of ADHHO for low dimension optimization problems, and the performance in high-dimensional optimization problems is not yet clear. It can be further extended to high-dimensional optimization problems in future research. Besides, we also intend to use the ADHHO algorithm to solve multi-objective optimization, constrained optimization, and NP-hard problems in future work.

REFERENCES

- [1] M. N. Omidvar, X. Li, Y. Mei, and X. Yao, "Cooperative co-evolution with differential grouping for large scale optimization," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 378–393, Jun. 2014.
- [2] S. Bubeck, "Convex optimization: Algorithms and complexity," *Found. Trends Mach. Learn.*, vol. 8, no. 4, pp. 231–357, May 2014. [Online]. Available: <https://arxiv.org/abs/1405.4980>
- [3] Y. Zhong, L. Wang, M. Lin, and H. Zhang, "Discrete pigeon-inspired optimization algorithm with metropolis acceptance criterion for large-scale traveling salesman problem," *Swarm Evol. Comput.*, vol. 48, pp. 134–144, Aug. 2019.
- [4] J. Liang, W. Xu, C. Yue, K. Yu, H. Song, O. D. Crisalle, and B. Qu, "Multimodal multiobjective optimization with differential evolution," *Swarm Evol. Comput.*, vol. 48, pp. 1028–1059, Feb. 2019.
- [5] A. Slowik and H. Kwasnicka, "Nature inspired methods and their industry applications—Swarm intelligence algorithms," *IEEE Trans. Ind. Inform.*, vol. 14, no. 3, pp. 1004–1015, Mar. 2018.
- [6] M. Mavrouniotis, C. Li, and S. Yang, "A survey of swarm intelligence for dynamic optimization: Algorithms and applications," *Swarm Evol. Comput.*, vol. 33, pp. 1–17, Apr. 2017.
- [7] H. Li, F. He, Y. Liang, and Q. Quan, "A dividing-based many-objective evolutionary algorithm for large-scale feature selection," *Soft Comput.*, vol. 24, no. 9, pp. 6851–6870, May 2020.
- [8] N. Hou, F. He, Y. Zhou, and Y. Chen, "An efficient GPU-based parallel tabu search algorithm for hardware/software co-design," *Frontiers Comput. Sci.*, vol. 14, no. 5, Mar. 2020, Art. no. 145316.
- [9] M. Le Quiniou, P. Mandel, and L. Monier, "Optimization of drinking water and sewer hydraulic management: Coupling of a genetic algorithm and two network hydraulic tools," *Procedia Eng.*, vol. 89, pp. 710–718, Jan. 2014.
- [10] Z. Sai, Y. Fan, T. Yuliang, X. Lei, and Z. Yifong, "Optimized algorithm of sensor node deployment for intelligent agricultural monitoring," *Comput. Electron. Agricult.*, vol. 127, pp. 76–86, Sep. 2016.
- [11] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, Perth, WA, Australia, Nov./Dec. 1995, pp. 1942–1948.
- [12] B. Yan, Z. Zhao, Y. Zhou, W. Yuan, J. Li, J. Wu, and D. Cheng, "A particle swarm optimization algorithm with random learning mechanism and levy flight for optimization of atomic clusters," *Comput. Phys. Commun.*, vol. 219, pp. 79–86, Oct. 2017.
- [13] K. Z. Gao, P. N. Suganthan, T. J. Chua, C. S. Chong, T. X. Cai, and Q. K. Pan, "A two-stage artificial bee colony algorithm scheduling flexible job-shop scheduling problem with new job insertion," *Expert Syst. Appl.*, vol. 42, no. 21, pp. 7652–7663, Nov. 2015.
- [14] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, Nov. 2006.
- [15] H. Wang, F. Guo, H. Yao, S. He, and X. Xu, "Collision avoidance planning method of USV based on improved ant colony optimization algorithm," *IEEE Access*, vol. 7, pp. 52964–52975, Mar. 2019.
- [16] X. S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proc. World Congr. Nature Biol. Inspired Comput.*, Dec. 2009, pp. 210–214.
- [17] L. Wang, Y. Zhong, and Y. Yin, "Nearest neighbour cuckoo search algorithm with probabilistic mutation," *Appl. Soft Comput.*, vol. 49, pp. 498–509, Dec. 2016.
- [18] W.-T. Pan, "A new fruit fly optimization algorithm: Taking the financial distress model as an example," *Knowl.-Based Syst.*, vol. 26, pp. 69–74, Feb. 2012.
- [19] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.
- [20] X. S. Yang, "Firefly algorithm, stochastic test functions and design optimization," *Int. J. Bio-Inspired Comput.*, vol. 2, no. 2, pp. 78–84, Mar. 2010. [Online]. Available: <http://arxiv.org/abs/1003.1409>
- [21] S. Mirjalili, "Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Comput. Appl.*, vol. 27, no. 4, pp. 1053–1073, May 2016.
- [22] M. Jain, V. Singh, and A. Rani, "A novel nature-inspired algorithm for optimization: Squirrel search algorithm," *Swarm Evol. Comput.*, vol. 44, pp. 148–175, Feb. 2019.
- [23] X. Zhang, K. Zhao, L. Wang, Y. Wang, and Y. Niu, "An improved squirrel search algorithm with reproductive behavior," *IEEE Access*, vol. 8, pp. 101118–101132, May 2020.
- [24] F. A. Hashim, E. H. Houssein, M. S. Mabrouk, W. Al-Atabany, and S. Mirjalili, "Henry gas solubility optimization: A novel physics-based algorithm," *Future Gener. Comput. Syst.*, vol. 101, pp. 646–667, Dec. 2019.
- [25] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," *Future Gener. Comput. Syst.*, vol. 111, pp. 300–323, Oct. 2020.
- [26] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, "Equilibrium optimizer: A novel optimization algorithm," *Knowl.-Based Syst.*, vol. 191, Mar. 2020, Art. no. 105190.
- [27] Z. Meng, J.-S. Pan, and H. Xu, "QUasi-affine transformation evolutionary (QUATRE) algorithm: A cooperative swarm based algorithm for global optimization," *Knowl.-Based Syst.*, vol. 109, pp. 104–121, Oct. 2016.
- [28] M. Abdel-Basset, V. Chang, and R. Mohamed, "A novel equilibrium optimization algorithm for multi-thresholding image segmentation problems," *Neural Computing and Applications*. Mar. 2020, pp. 1–34, doi: [10.1007/s00521-020-04820-y](https://doi.org/10.1007/s00521-020-04820-y).
- [29] H. Özkaya, M. Yıldız, A. R. Yıldız, S. Bureerat, B. S. Yıldız, and S. M. Sait, "The equilibrium optimization algorithm and the response surface-based metamodel for optimal structural design of vehicle components," *Mater. Test.*, vol. 62, no. 5, pp. 492–496, May 2020.
- [30] J.-S. Pan, Z. Meng, H. Xu, and X. Li, "QUasi-affine transformation evolutionary (QUATRE) algorithm: A new simple and accurate structure for global optimization," in *Proc. Int. Conf. Ind., Eng. Appl. Appl. Intell. Syst.*, Morioka, Japan, Aug. 2016, pp. 657–667.
- [31] Z. Meng and J.-S. Pan, "QUasi-affine TRansformation evolution with external ARchive (QUATRE-EAR): An enhanced structure for differential evolution," *Knowl.-Based Syst.*, vol. 155, pp. 35–53, Sep. 2018.
- [32] K. Matsumoto, F. Ren, M. Matsuoka, M. Yoshida, and K. Kita, "Slang feature extraction by analysing topic change on social media," *CAAI Trans. Intell. Technol.*, vol. 4, no. 1, pp. 64–71, Mar. 2019.
- [33] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.
- [34] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019.
- [35] H. Jia, C. Lang, D. Oliva, W. Song, and X. Peng, "Dynamic Harris hawks optimization with mutation mechanism for satellite image segmentation," *Remote Sens.*, vol. 11, no. 12, p. 1421, Jun. 2019.
- [36] A. R. Yıldız, B. S. Yıldız, S. M. Sait, and X. Li, "The Harris hawks, grasshopper and multi-verse optimization algorithms for the selection of optimal machining parameters in manufacturing operations," *Mater. Test.*, vol. 61, no. 8, pp. 725–733, Aug. 2019.
- [37] N. Amiri Golilarz, H. Gao, and H. Demirel, "Satellite image de-noising with Harris hawks meta heuristic optimization algorithm and improved adaptive generalized Gaussian distribution threshold function," *IEEE Access*, vol. 7, pp. 57459–57468, Apr. 2019.

- [38] 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," *J. Cleaner Prod.*, vol. 244, Jan. 2020, Art. no. 118778.
- [39] E. H. Houssein, M. R. Saad, K. Hussain, W. Zhu, H. Shaban, and M. Hassaballah, "Optimal sink node placement in large scale wireless sensor networks based on Harris' hawk optimization algorithm," *IEEE Access*, vol. 8, pp. 19381–19397, Jan. 2020.
- [40] M. S. Mehta, M. B. Singh, and M. Gagandeep, "Harris Hawks optimization for solving optimum load dispatch problem in power system," *Int. J. Eng. Res. Technol.*, vol. 8, no. 6, pp. 962–968, 2019.
- [41] K. Hussain, W. Zhu, and M. N. M. Salleh, "Long-term memory Harris' hawk optimization for high dimensional and optimal power flow problems," *IEEE Access*, vol. 7, pp. 147596–147616, Oct. 2019.
- [42] Q. Fan, Z. Chen, and Z. Xia, "A novel quasi-reflected Harris hawks optimization algorithm for global optimization problems," *Soft Comput.*, vol. 10, no. 2, pp. 1–19, Mar. 2020.
- [43] V. K. Kamboj, A. Nandi, A. Bhadoria, and S. Sehgal, "An intensify Harris hawks optimizer for numerical and engineering optimization problems," *Appl. Soft Comput.*, vol. 89, Apr. 2020, Art. no. 106018.
- [44] E. Kurtuluş, A. R. Yıldız, S. M. Sait, and S. Bureerat, "A novel hybrid Harris hawks-simulated annealing algorithm and RBF-based metamodel for design optimization of highway guardrails," *Mater. Test.*, vol. 62, no. 3, pp. 251–260, Mar. 2020.
- [45] H. Moayedi, M. M. Abdullahi, H. Nguyen, and A. S. A. Rashid, "Comparison of dragonfly algorithm and Harris hawks optimization evolutionary data mining techniques for the assessment of bearing capacity of footings over two-layer foundation soils," *Eng. Comput.*, pp. 1–11, Aug. 2019, doi: [10.1007/s00366-019-00834-w](https://doi.org/10.1007/s00366-019-00834-w).
- [46] A. S. Menesy, H. M. Sultan, A. Selim, M. G. Ashmawy, and S. Kamel, "Developing and applying chaotic Harris hawks optimization technique for extracting parameters of several proton exchange membrane fuel cell stacks," *IEEE Access*, vol. 8, pp. 1146–1159, Dec. 2020.
- [47] D. Yousri, D. Allam, and M. B. Eteiba, "Optimal photovoltaic array reconfiguration for alleviating the partial shading influence based on a modified Harris hawks optimizer," *Energy Convers. Manage.*, vol. 206, Feb. 2020, Art. no. 112470.
- [48] H. M. Ridha, A. A. Heidari, M. Wang, and H. Chen, "Boosted mutation-based Harris hawks optimizer for parameters identification of single-diode solar cell models," *Energy Convers. Manage.*, vol. 209, Apr. 2020, Art. no. 112660.
- [49] F. Zhong, H. Li, and S. Zhong, "A modified ABC algorithm based on improved-global-best-guided approach and adaptive-limit strategy for global optimization," *Appl. Soft Comput.*, vol. 46, pp. 469–486, Sep. 2016.
- [50] M. S. Kiran, H. Hakli, M. Gunduz, and H. Uguz, "Artificial bee colony algorithm with variable search strategy for continuous optimization," *Inf. Sci.*, vol. 300, pp. 140–157, Apr. 2015.
- [51] D. Karaboga and B. Akay, "A comparative study of artificial bee colony algorithm," *Appl. Math. Comput.*, vol. 214, no. 1, pp. 108–132, Aug. 2009.
- [52] J. C. Bednarz, "Cooperative hunting Harris' hawks (parabuteo unicinctus)," *Science*, vol. 239, no. 4847, pp. 1525–1527, Mar. 1988.
- [53] X. Wang and X. Zou, "Modeling the fear effect in predator–prey interactions with adaptive avoidance of predators," *Bull. Math. Biol.*, vol. 79, no. 6, pp. 1325–1359, Jun. 2017.
- [54] M. Moustafa, M. H. Mohd, A. I. Ismail, and F. A. Abdullah, "Dynamical analysis of a fractional-order Rosenzweig–MacArthur model incorporating a prey refuge," *Chaos, Solitons Fractals*, vol. 109, pp. 1–13, Apr. 2018.
- [55] M. Jamil and X. S. Yang, "A literature survey of benchmark functions for global optimization problems," *J. Math. Model. Numer. Optim.*, vol. 4, no. 2, pp. 150–194, Aug. 2013. [Online]. Available: <http://arxiv.org/abs/1308.4008>
- [56] F. Wang, H. Zhang, K. Li, Z. Lin, J. Yang, and X.-L. Shen, "A hybrid particle swarm optimization algorithm using adaptive learning strategy," *Inf. Sci.*, vols. 436–437, pp. 162–177, Apr. 2018.
- [57] J. J. Liang, B. Y. Qu, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization," *Comput. Intell. Lab., Zhengzhou Univ., Zhengzhou, China, Nanyang Technol. Univ., Singapore, Tech. Rep. 201311*, 2013.
- [58] S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behavior: A case study on the CEC'2005 special session on real parameter optimization," *J. Heuristics*, vol. 15, no. 6, p. 617, May 2008.
- [59] A. Kaveh and V. R. Mahdavi, "Colliding bodies optimization: A novel meta-heuristic method," *Comput. Struct.*, vol. 139, pp. 18–27, Jul. 2014.
- [60] A. Kaveh, T. Bakhshpoori, and E. Afshari, "An efficient hybrid particle swarm and swallow swarm optimization algorithm," *Comput. Struct.*, vol. 143, pp. 40–59, Sep. 2014.

• • •