AIMS Mathematics, 10(5): 11783–11812.
DOI: 10.3934/math.2025533
Received: 08 April 2025
Revised: 09 May 2025
Accepted: 16 May 2025
Published: 22 May 2025

https://www.aimspress.com/journal/Math

Research article

A multi-strategy upgraded Harris Hawk optimization algorithm for solving nonlinear inequality constrained optimization problems

Juhe Sun[1], **Guolin Huang**[1,*], **Li Wang**[1], **Chuanjun Yin**[1] **and Ning Ma**[2]

[1] School of Science, Shenyang Aerospace University, Shenyang 110136, China
[2] School of Mechatronics Engineering, Shenyang Aerospace University, Shenyang 110136, China

* **Correspondence:** Email: 15023912554@163.com.

Abstract: This study presented an upgraded version of the Harris Hawk optimization algorithm (UHHO) designed to overcome the inherent limitations of the original algorithm, especially in solving nonlinear constrained optimization problems that tend to converge prematurely and fall into local optima. First, the initial population generated in a random way was replaced by a good point set strategy. Second, we replaced the linear strategy with a nonlinear strategy in the intermediate stage in order to optimize the global search process. Furthermore, the sine-cosine strategy and L-C cascade chaos strategy were introduced in the development stage to perturb the population's position. This aimed to better explore the neighborhood of Harris Hawk optimal individuals in depth, enhance the local search ability of the algorithm, and avoid the algorithm falling into local optima. Some numerical experiments for solving nonlinear inequality constrained optimization problems are presented at the end of this paper. The simulation results show that the multi-strategy upgraded Harris Hawk algorithm can effectively avoid the problem of the standard Harris Hawk optimization algorithm falling into local optima.

Keywords: multi-strategy upgrade; local optimality; good point set strategy; nonlinear factor strategy; sine-cosine strategy; L-C chaos strategy; nonlinear constrained optimization problems
Mathematics Subject Classification: 68W50, 90C30

1. Introduction

So far, many practical application problems in mathematics-related fields can be attributed to optimization problems, which have attracted a lot of attention due to their complexity and wide range of applications. For example, many optimization problems in artificial intelligence are usually characterized by a variety of features such as constraints, nonlinearities, and discretization [1, 2], which pose a great challenge to solve and make the problems difficult to be solved by traditional

mathematical planning methods [3, 4].

For the moment, designing new algorithms to solve optimization problems is an extremely important research direction. These related optimization algorithms have important applications in the design of engineering structures, aerospace performance optimization, and mechanical engineering optimization, among others [5]. With the deepening of research, intelligent optimization algorithms, as a kind of integration of biology, mathematics, and other multidisciplinary knowledge of new optimization algorithms, have gradually become an effective method for the resolution of complex problems. For example, in the field of engineering optimization, algorithms for intelligent optimization have been used extensively in path planning, job shop scheduling problems, intelligent traffic control, power system optimization, etc., and have achieved significant breakthroughs and results [6–8]. At present, among the widely adopted metaheuristic approaches are the ant colony algorithm, particle swarm optimization, genetic algorithms, and the firefly method, as documented in prior studies [9–12]. In order to overcome these drawbacks, Wang et al.'s significant contribution [13] involved the fusion of quantum state updating mechanisms with quasi-contrastive learning frameworks, and combined with a dynamic Q-learning optimization process and an advanced variable spiral local domain approach, avoid premature convergence and improve the ability of global optimization.

Harris Hawk optimization (HHO) was proposed by Heidari, Mirjalili, and Faris et al. in 2019 as a new intelligent optimization algorithm [14]. It is inspired by the hunting behaviors of Hawks. The algorithm is designed to perform efficient and effective hunting procedures in real-world conditions. This is due to its small number of tuning parameters, localized search capability, and ease of implementation. It is widely used in the solution of various engineering problems and shows better parameter search performance [15, 16]. For example, QAIS et al. [17] came up with a new method combining computational and HHO algorithms for updating the PV panel parameters for optimizing the time-domain finite-difference method model.

As we all know, according to the "no free lunch" theorem in optimization, every algorithm, such as Harris Hawk optimization, exhibits specific performance boundaries and cannot address all problem domains effectively. The Harris Hawk optimization algorithm also has certain limitations, such as being prone to local optimization in the late search stage [18]. Therefore, upgrading the Harris Hawk optimization algorithm for its defects has important research value. To address the above problems, a multi-strategy upgraded version of the Harris Hawk algorithm is presented. First, population initialization is optimized through good point set theory, effectively improving the population distribution characteristics. Second, we introduce the nonlinear energy factor to improve the global search ability. In addition, a sine-cosine strategy is introduced to update the individual positions in the soft and hard attack phases of the HHO algorithm. Furthermore, an L-C chaos strategy is introduced in the progressive fast swooping soft attack and progressive fast swooping hard attack phases, which performs a positional perturbation search for inferior individuals to improve the algorithm's ability to search for neighborhoods around the elite solution while enhancing the performance of local exploitation. Finally, comprehensive numerical simulations were conducted to evaluate the algorithm's performance. The results demonstrate that the multi-strategy upgraded HHO algorithm exhibits superior optimization capability and solution stability, while effectively reducing the local optimization problem to a large extent.

This study is organized into six components: Section 1 presents the research background and

motivation for Harris Hawk optimization; Section 2 details the algorithm's fundamental framework and operational mechanisms; Section 3 gives the multi-strategy upgrade methods; Section 4 conducts comprehensive performance assessments; and Section 5 validates the approach through numerical experimentation. The last section gives relevant conclusions.

2. HHO rationale and steps

The HHO algorithm's mathematical framework consists of two key phases: exploration and exploitation. The corresponding tree structure is depicted in Figure 1.

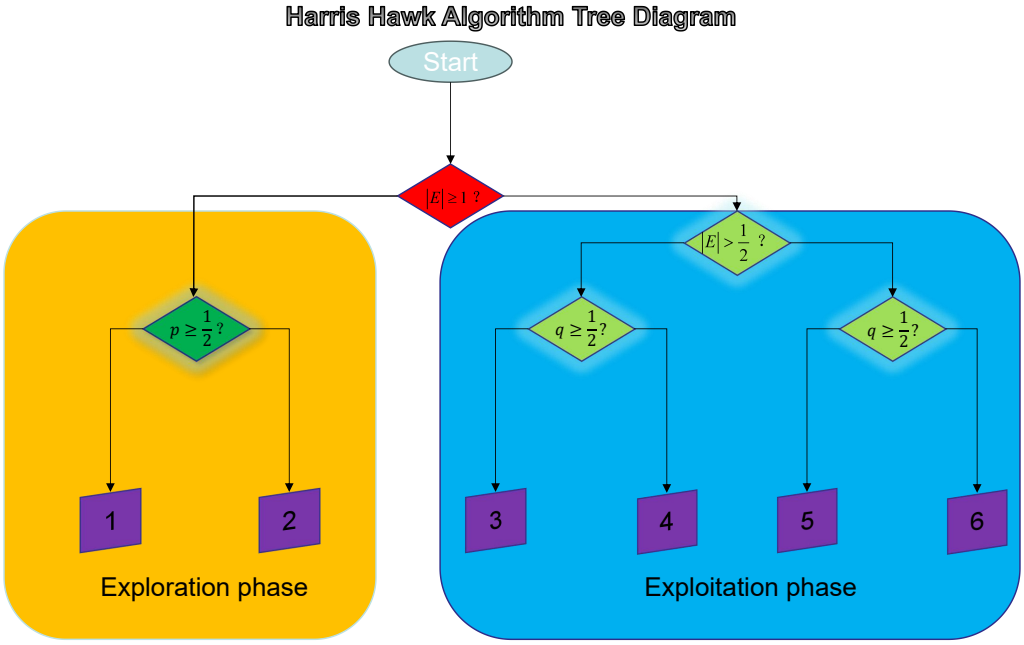

Figure 1. Harris Hawk algorithm tree diagram.

2.1. Principles of HHO

2.1.1. Initial population generation mode

During the population initialization stage, randomized position vectors are created, and their fitness value is calculated based on the corresponding location. Set the population size $N = 400$. Figures 2 and 3 show the 2D and 3D spatial distributions of the initialized population.

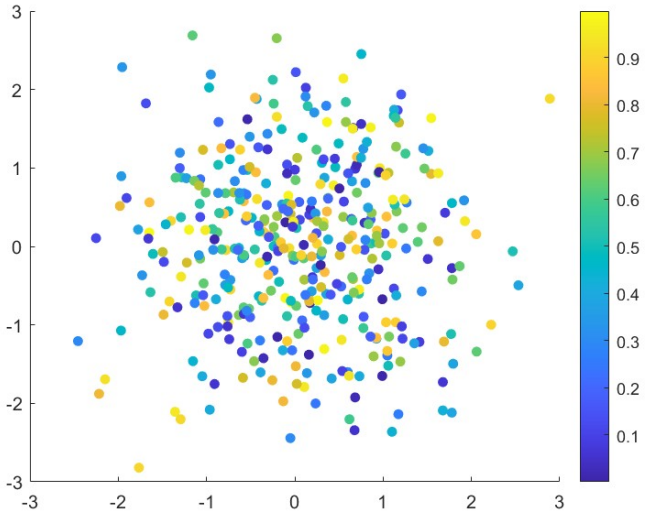

Figure 2. Two-dimensional distribution map.

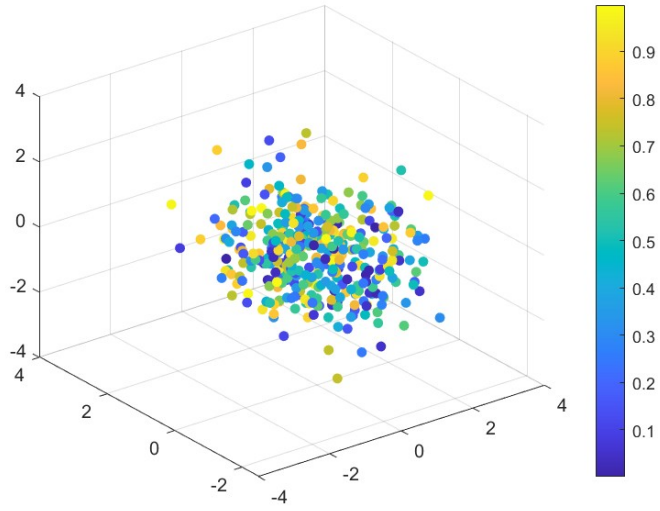

Figure 3. Three-dimensional distribution map.

2.1.2. Algorithm exploration phase

Harris's Hawks utilize a strategic probability p to capture food within their search range. See Eq (2.1) for the exact way the location is updated [14]:

$$X_{mean}(t) = \frac{1}{N}\sum_{i=1}^{N} X_i(t), X(t+1) = \begin{cases} X_{rand}(t) - r_1|X_{rand(t)} - 2r_2X(t)|, p \geq \frac{1}{2}, \\ (X_{prey}(t) - X_{mean}(t)) - r_3(lb + r_4(ub - lb)), p < \frac{1}{2}. \end{cases} \tag{2.1}$$

The meaning of these symbols is shown in Table 1.

Table 1. Table of symbol meanings.

Notation	Hidden meaning
$X_{mean}(t)$	Average position of Harris Hawks
$X_i(t)$	The search space location of the i-th hawk individual at the t-th iteration
N	Harris's hawk population
$X(t)$	Harris Hawk position at the t-th iteration
$X_{rand}(t)$	Randomized location of Harris Hawk at the t-th iteration
$X_{prey}(t)$	Prey position at the t-th iteration
$X(t+1)$	Harris Hawk position at the $(t+1)$-th iteration
$r_i, (i = 1, 2, 3, 4)$	[0, 1] range of random numbers
lb	Lower bound of feasible domain
ub	Upper bound of feasible region

2.1.3. Algorithm intermediate phase

The intermediate stage regulates its behavior through an energy dissipation mechanism, and the escape energy $E(t)$ exhibits an iterative linear decay, which is expressed in Eq (2.2):

$$E(t) = E_0 \times C = 2E_0(1 - \frac{t}{T}), \tag{2.2}$$

where the coefficient C exhibits iteration-dependent characteristics and E_0 is the initial state energy of the prey.

Below is a plot of the parameter C, the initial energy E_0, and the escape energy $E(t)$ as a function of the number of iterations (see Figures 4–6).

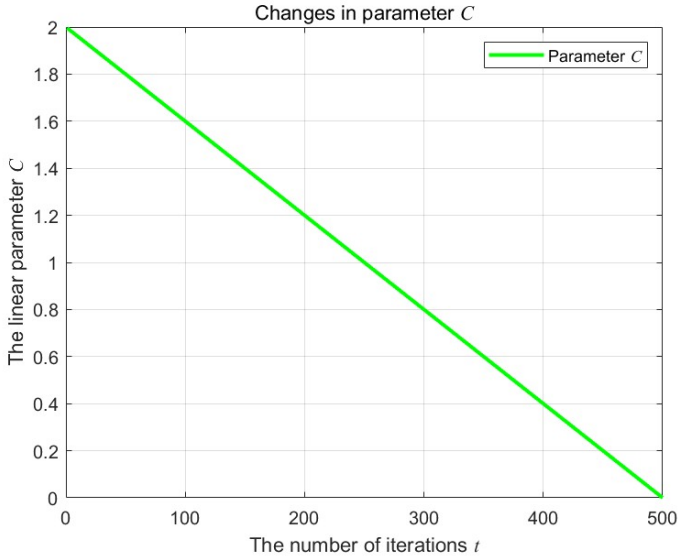

Figure 4. Linear parameter C.

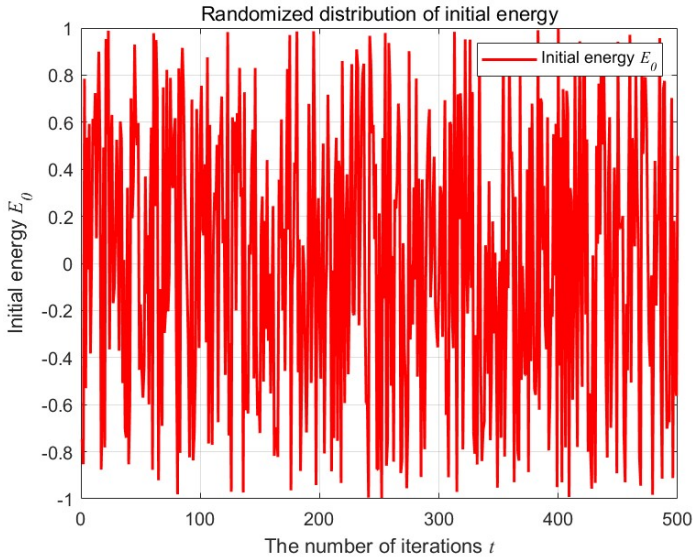

Figure 5. Initial energy E_0.

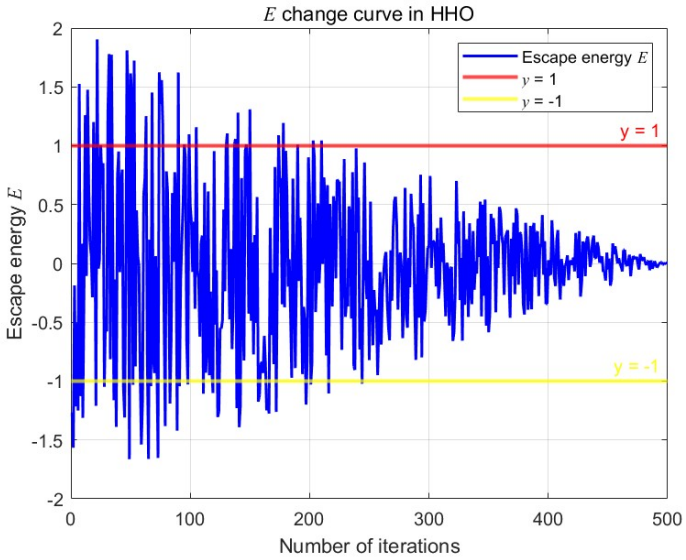

Figure 6. Escape energy $E(t)$.

If $|E| \geq 1$, the global exploration phase is triggered; otherwise $|E| < 1$ and the algorithm shifts to local optimization. As can be seen from Figure 6, the algorithm is prone to fall into local optimization, so it is necessary to upgrade the algorithm to overcome the shortcomings.

2.1.4. Algorithm development phase

During the development phase, Harris's Hawks adopt different hunting behaviors based on the escape behavior of their prey, and four different hunting strategies are given below:

Soft attack approach: if $q \geq \frac{1}{2}$ and $|E| \geq \frac{1}{2}$, the Harris Hawk will perform a soft attack on it, at which point the behavior can be abstracted as the following mathematical model (2.3) [14]:

$$X(t+1) = X_{prey}(t) - X(t) - E\left|JX_{prey}(t) - X(t)\right|. \tag{2.3}$$

Hard attack method: if $q \geq \frac{1}{2}$ and $|E| < \frac{1}{2}$, the Harris Hawk will hard attack it, at which point the behavior can be abstracted as the following mathematical model (2.4) [14]:

$$X(t+1) = X_{prey}(t) - E\left|X_{prey}(t) - X(t)\right|. \tag{2.4}$$

Progressive fast dive soft attack: if $q < \frac{1}{2}$ and $|E| \geq \frac{1}{2}$, the Harris Hawk employs a progressive fast dive soft attack strategy, and its specific formula refers to the following Eq (2.5):

$$\begin{aligned} &Y = X_{prey}(t) - E\left|JX_{prey}(t) - X(t)\right|, \\ &Z = Y + S \times Levy(D), \\ &X(t+1) = \begin{cases} Y, F(Y) < F(X(t)), \\ Z, F(Z) < F(X(t)), \end{cases} \end{aligned} \tag{2.5}$$

where Levy flight is a special stochastic wandering model. The introduction of Levy flight and the combination of a specific position update strategy can enhance the global convergence of the algorithm, which is formulated as follows:

$$Levy(D) = \frac{\mu\sigma}{100|v|^{\frac{1}{\beta}}}, \tag{2.6}$$

where μ and v are random numbers between $(0, 1)$ and β is a constant, generally taken as $\beta = 1.5$, and $\sigma = (\frac{\Gamma(1+\beta)\times sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2})\times\beta\times 2^{(\frac{\beta-1}{2})}})^{\frac{1}{\beta}}$.

Progressive fast dive hard attack: if $q < \frac{1}{2}$ and $|E| < \frac{1}{2}$, the Harris Hawk will use the progressive fast dive hard attack strategy to hunt prey, with a positional strategy detailed in Eq (2.7) below:

$$\begin{aligned} &Y = X_{prey}(t) - E\left|JX_{prey}(t) - X_m(t)\right|, \\ &Z = Y + S \times Levy(D), \\ &X(t+1) = \begin{cases} Y, F(Y) < F(X(t)), \\ Z, F(Z) < F(X(t)). \end{cases} \end{aligned} \tag{2.7}$$

The symbolic meanings of Eqs (2.3)–(2.7) are shown in Table 2.

Table 2. Table of symbol meanings.

Notation	Hidden meaning
J	Jumping strength of prey: [0,2]
D	Dimension
S	Randomized row vectors of dimension D

2.2. HHO algorithm steps

Step 1: Determine the population size N and the iteration count T, and then initialize the population;
Step 2: Compute fitness values for all individuals, identifying and storing the current optimum;
Step 3: Select different strategies to update the position of individuals according to p, referring to the exploration phase formula;
Step 4: Update the prey escape energy E, referring to the intermediate stage formula;
Step 5: Adjust individual positions using energy E and probability q via exploitation-phase equations;
Step 6: According to the algorithm stopping criterion, if it is satisfied, stop and output the optimal solution and fitness value; otherwise go to step 2.

2.3. HHO algorithm flow chart

The flowchart of the Harris Hawk algorithm is shown in Figure 7:

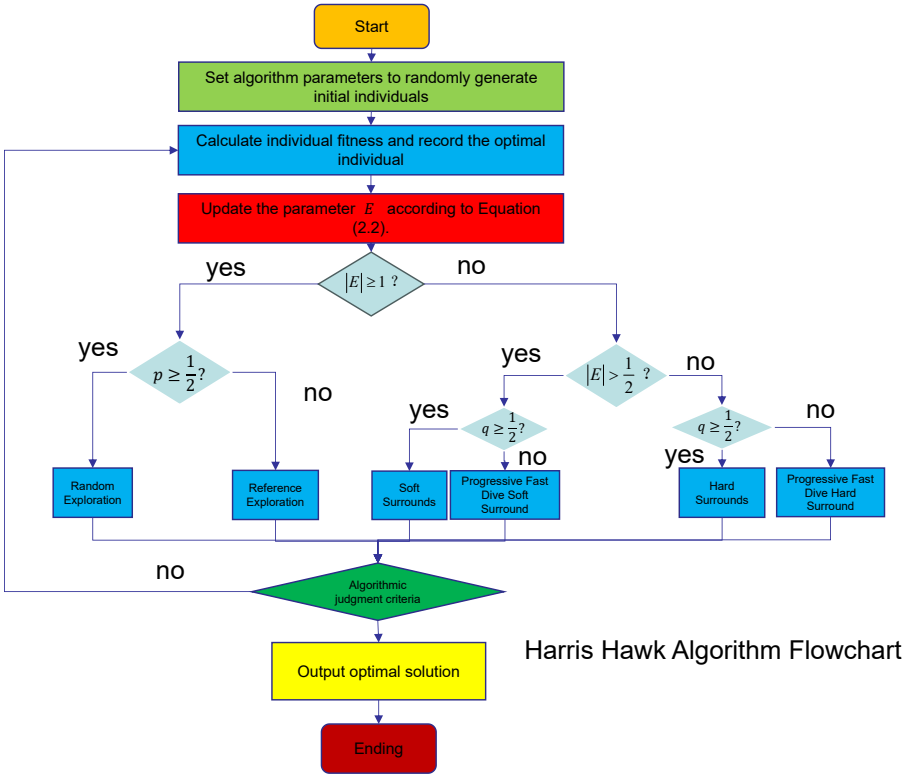

Figure 7. Harris Hawk algorithm flow chart.

2.4. Algorithm advantages and disadvantages analysis

2.4.1. Benefits analysis

(1) The HHO algorithm has a simple structure.

The HHO algorithm maps the optimization problem to a prey search process by simulating the predatory behavior of Harris Hawks. Its three phases (exploration, transition, and development) have a clear structure with few parameters and simple computation, which makes it easy to implement and

therefore simple to structure.

(2) The HHO algorithm converges quickly.

The algorithm's exploitation phase employs prey escape energy and probability to simulate hunting behavior, enabling rapid convergence toward optimal solutions. An adaptive mechanism dynamically adjusts the transition between global exploration and local exploitation based on escape energy, significantly accelerating convergence. Furthermore, Levy flight mechanisms enhance search randomness and directional guidance, further improving convergence speed.

2.4.2. Cons analysis

(1) Unreasonable in the exploration and development phase.

The escape energy E is designed in such a way that the algorithm is weakened in the later stage of exploration and is prone to fall into the local optimum. As the iteration proceeds, the change in the value of E causes the algorithm to be overly inclined to the exploitation stage, searching repeatedly in the local region and making it difficult to jump out of the local optimal solution.

(2) HHO overemphasizes elite individuals.

Each iteration forces the eagle population to move closer to the current optimum, which makes the algorithm prone to converge to a locally optimal solution and thus fall into local optima.

In the face of the aforementioned shortcomings of the HHO algorithm, further upgrading of the algorithm is necessary to upgrade its global exploration capability.

3. Multi-strategy upgrade of the Harris Hawk algorithm

3.1. Algorithm upgrade strategies

3.1.1. Modification of initial population generation

With a randomly generated initial population, the Harris Hawk algorithm has a relatively high probability of duplicate individuals and is prone to problems such as local optimization.

Hence, incorporating a novel approach for generating the initial population emerges as a necessity to surmount this limitation. For instance, the initial population crafted via the good point set method exhibits a uniform dispersion [19], which substantially diminishes the repetition rate among individuals and thereby amplifies the diversity of the Harris Hawk's initial population.

Definition 3.1. Let T_s be a unit cube in s-dimensional space and if there is

$$P_n(k) = \left\{(r_1^n k, r_2^n k, ..., r_s^n k)|r \in T_s, k = 1, 2, ..., n\right\}, \tag{3.1}$$

then $P_n(k)$ is said to be the set of good points and r is a good point. In order to visualize the initial population distribution, Figures 8 and 9 present a comparative visualization of two-dimensional initialization patterns, with the population size set to 400.

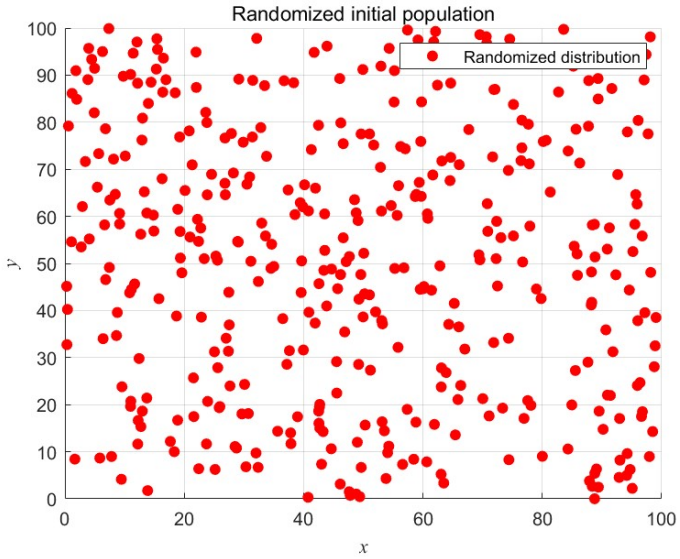

Figure 8. Randomized initial population.

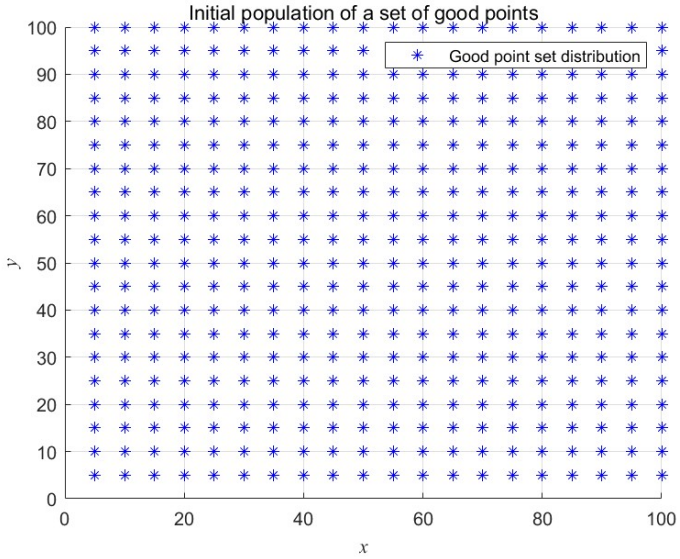

Figure 9. Initial population of a good point set.

From the above figure, it can be seen that the initial population produced by the good point set method outperforms the randomly generated initial population, thus optimizing the Harris Hawk algorithm to be more traversal.

3.1.2. Introducing a nonlinearly varying energy factor strategy

Intermediate phase:

Within the Harris Hawk optimization algorithm, the scope of the search is contingent upon the escape energy E, which is updated through Eq (2.2). The HHO algorithm employs a convergence

parameter C that undergoes linear reduction from 2 to 0 throughout iterations. However, this linear decay mechanism fails to reflect the nonlinear population dynamics observed during optimization. The fixed decrement rate of C restricts adaptive search behavior, creating an imbalance between exploration and exploitation phases. Consequently, this limitation frequently leads to premature convergence in the final optimization stages.

In order to overcome the shortcomings of the Harris Hawk algorithm, this paper gives an energy factor C_0 which presents a nonlinear decreasing change (see (3.2) below). In the early iteration period, C_0 decreases at a smaller rate to keep E_0 as a larger value, which is conducive to the algorithm searching in a wide range; in the later period, C_0 decreases faster, so that E_0 has a longer period of time to keep the value smaller.

$$\begin{aligned} C_0 &= 2(1 - (\frac{t}{T})^{\frac{1}{5}})^{\frac{1}{5}}, \\ E &= E_0 C_0. \end{aligned} \tag{3.2}$$

Thus solving the problem of insufficient global search ability of the above algorithm, Figures 10 and 11 show the iteration curve and the change of energy factor. In this way, it can better control the optimization of the search process with strong coordination between the global and local search ability.

Upon contrasting Figure 6 with Figure 11, it becomes evident that the adoption of this alternative energy updating strategy effectively prevents the algorithm from becoming trapped in local optima, thereby enhancing both the convergence precision and velocity of the algorithm.

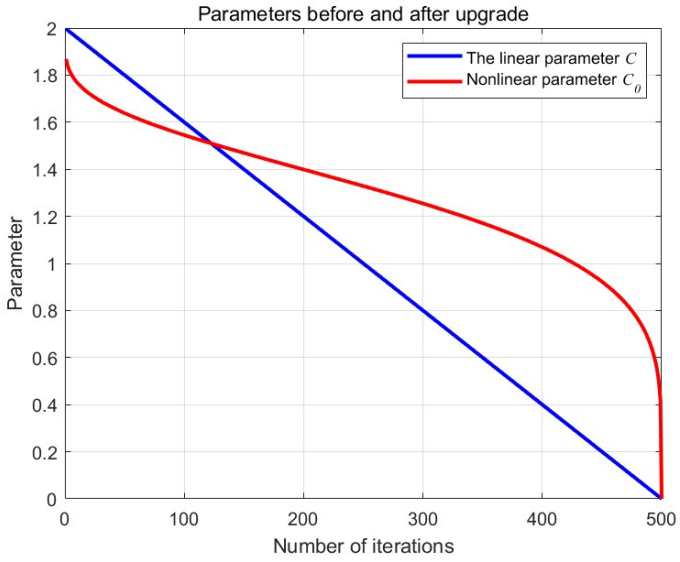

Figure 10. Parameters before and after upgrade.

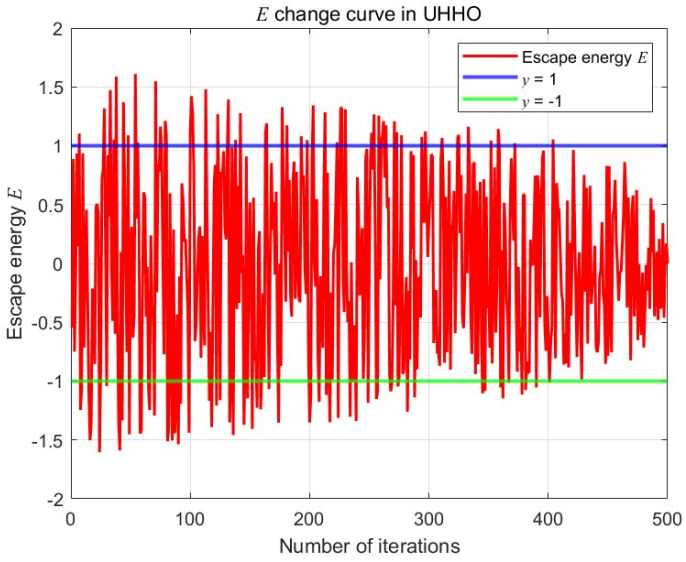

Figure 11. E change curve in UHHO.

3.1.3. Sine-cosine strategy

It can be seen from the process of the Harris Hawk development phase that the randomness of the prey position is not considered, which makes the Harris Hawk individuals easily miss the global optimal position and thus fall into local optimality. In order to solve the above problems, this paper adopts the sine-cosine strategy to update the individual positions in the soft attack and hard attack phases, which can improve the capture efficiency, effectively avoid the algorithm from falling into the local optimum, and improve its global search capability.

According to the sine-cosine strategy, Harris Hawk position Eqs (2.3) and (2.4) can be updated as:

$$\begin{aligned} X(t+1) &= X_{prey}(t) - sin(\alpha_E)(X(t) + E\left|JX_{prey}(t) - X(t)\right|), |E| \geq \frac{1}{2}, \\ X(t+1) &= X_{prey}(t) - cos(\alpha_E)E\left|X_{prey}(t) - X(t)\right|, |E| < \frac{1}{2}, \end{aligned} \tag{3.3}$$

where α_E denotes the angle of perception in $[0, 2\pi]$.

3.1.4. Disadvantaged individual L-C cascade chaos disturbance

Chaos is a phenomenon characterized by stochasticity and the ability to traverse all solutions in the problem space in a regular manner. For more details, refer to the literature [20]. Initial value sensitivity affects chaotic randomness and the Lyapunov exponent is a measure of initial value sensitivity [21]. Cascading of chaotic systems is a simple and effective way to improve the initial value sensitivity of the system.

Theorem 3.1. Let the cascade chaotic system formed by chaotic subsystems $f_1(x)$ and $f_2(x)$ be $f_2(f_1(x))$, and then the Lyapunov exponent of the chaotic system is equal to the sum of the Lyapunov exponents of the chaotic systems $f_1(x)$ and $f_2(x)$. The formula for the L-C cascade chaotic mapping is

as follows:

$$r_{n+1} = \left| \frac{[\mu r_n(1 - r_n)]^3}{a^2} - 3\mu r_n(1 - r_n) \right|, \tag{3.4}$$

where $a = 0.5$, $\mu \in [0, 2]$, and $r_n \in [0, 1]$.

In order to avoid the algorithm from falling into local optimum and to improve its convergence accuracy, this paper employs L-C cascade chaos to perturb the inferior individuals in its development stage so as to improve the search performance as shown in the following equations:

$$\begin{cases} Y = r_{n+1} \times X_{prey}(t) - E\left|Jr_{n+1} \times X_{prey}(t) - X(t)\right| \\ Z = Y + S \times LF(D) \end{cases}, |E| \geq \frac{1}{2},$$
$$\begin{cases} Y = r_{n+1} \times X_{prey}(t) - E\left|Jr_{n+1} \times X_{prey}(t) - X_m(t)\right| \\ Z = Y + S \times LF(D) \end{cases}, |E| < \frac{1}{2}, \tag{3.5}$$

and combined with the greedy retention strategy, the formula is as follows:

$$X(t + 1) = \begin{cases} Y, if F(Y) < F(X(t)), \\ Z, if F(Z) < F(X(t)), \end{cases} \tag{3.6}$$

where $F(x)$ is the fitness value of x.

3.2. Steps of the UHHO algorithm

In summary, the steps of the upgraded Harris Hawk algorithm proposed in this section are as follows:
Step 1: Generate the initial positions of N individuals in the population using the good point set strategy;
Step 2: Calculate the fitness value of each Harris Hawk position in the population by the fitness function and record the optimal individual;
Step 3: Perform the sine-cosine strategy;
Step 4: Select the Harris Hawk position update method based on the nonlinear escape energy of the prey;
Step 5: Execute L-C cascade chaos to perturb the inferior individuals;
Step 6: The iterative process terminates upon reaching the prescribed maximum generations, at which point subsequent operations are initiated; otherwise, return to Step 2 for further optimization;
Step 7: Output the global optimal position and fitness value.

The flowchart corresponding to the above steps of the UHHO algorithm is given below, see Figure 12:

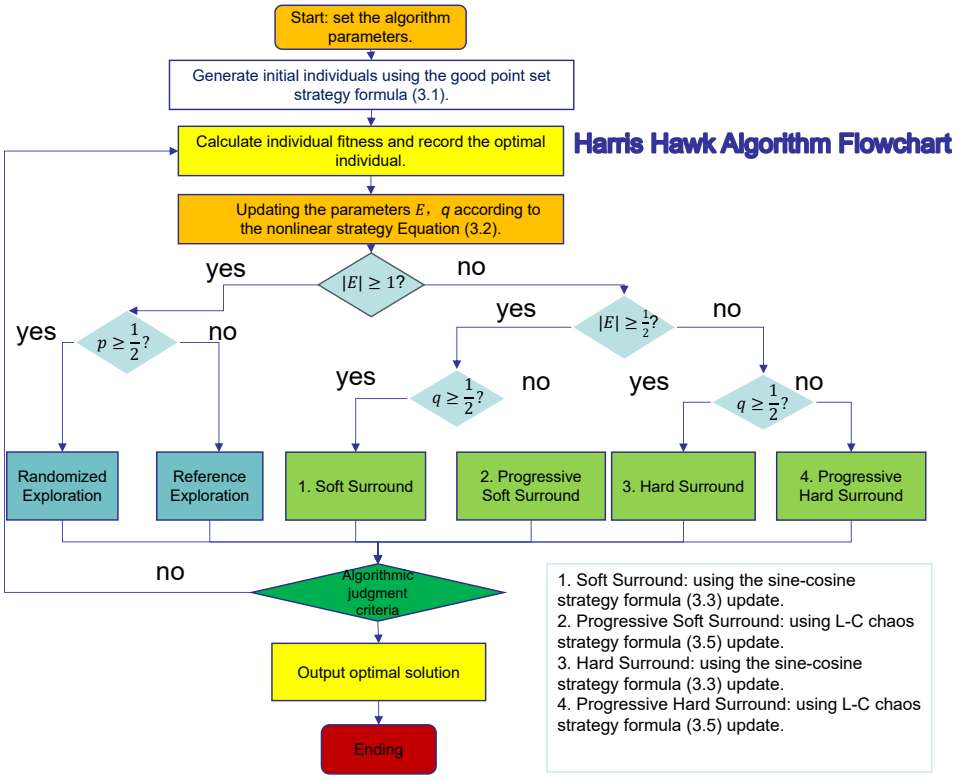

Figure 12. Flowchart of UHHO algorithm.

4. Algorithm performance evaluation

4.1. Experimental conditions

The hardware configuration of the experimental environment is a Windows 11 operating system, and the software configuration is MATLAB R2023*a*. To maintain experimental consistency, identical initialization parameters were applied across all algorithms: population size ($N = 400$), maximum iterations ($T = 500$), and 20 independent execution trials.

4.2. Test functions

This section is dedicated to evaluating the performance of the multi-policy upgraded Harris Hawk optimization algorithm. Six benchmark test functions have been chosen to assess the upgraded Harris Hawk optimization algorithm, with the detailed information of each benchmark test function presented in Tables 3 and 4.

Table 3. Function expression (math.).

F	Function expression (math.)
f_1	$\frac{1}{2} + \frac{sin^2(x_1^2+x_2^2)-\frac{1}{2}}{(1+\frac{1}{1000}(x_1^2+x_2^2))^2}$
f_2	$\sum_{i=1}^{n} x_i^2$
f_3	$(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})cosx_1 + 10$
f_4	$[1 + (x_1 + x_2 + 1)^2(19 + (x_1 + x_2)(3x_1 + x_2 - 14))]\times$ $[30 + (2x_1 - 3x_2^2)(18 + (2x_1 - 3x_2)(6x_1 - 9x_2 - 16))]$
f_5	$\sum_{i=1}^{n}(x_i + 0.5)^2$
f_6	$\sum_{i=1}^{n}[x_i^2 - 10cos(2\pi x_i) + 10]$

Table 4. Test function parameters.

Serial No.	F	Dimension	Range of values	Minimum value
1	$f_1(x)$	2	[-100,100]	0
2	$f_2(x)$	30	[-100,100]	0
3	$f_3(x)$	2	[-5,5]	0.398
4	$f_4(x)$	2	[-2,2]	3
5	$f_5(x)$	30	[-100,100]	0
6	$f_6(x)$	30	[-5.12,5.12]	0

Below are three-dimensional spatial images of the six benchmark functions, see Figures 13–18:

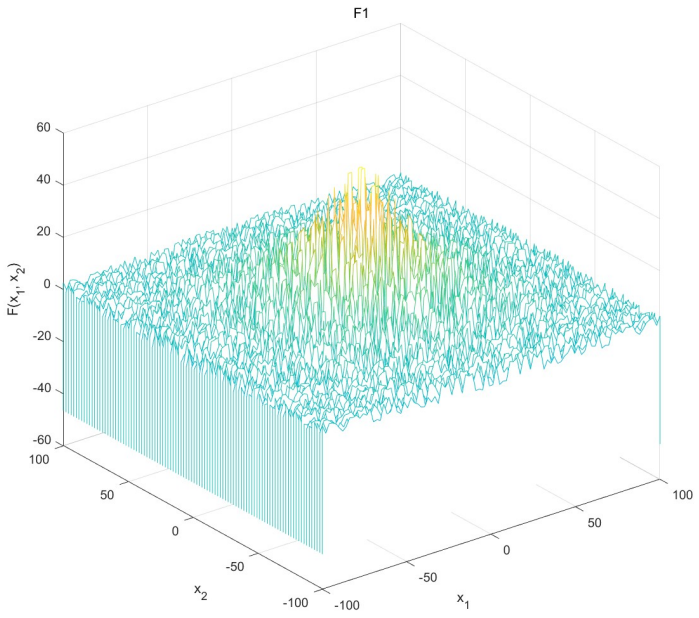

Figure 13. f_1 three-dimensional diagram.

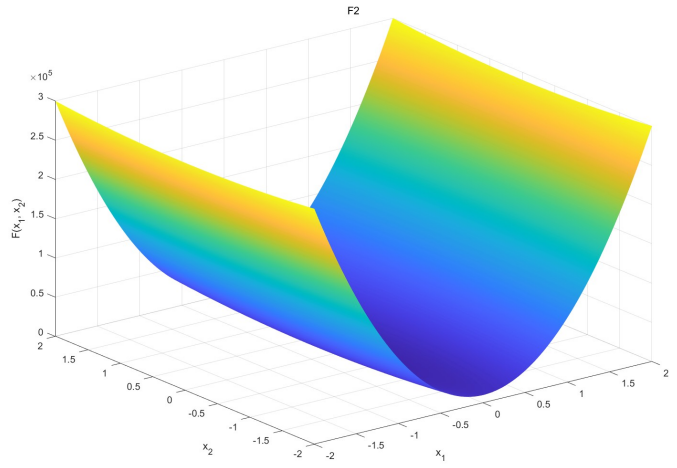

Figure 14. f_2 three-dimensional diagram.

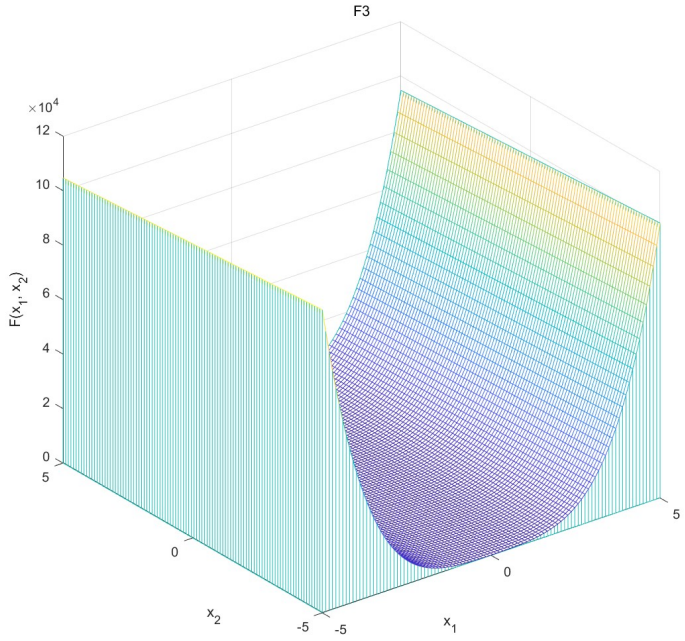

Figure 15. f_3 three-dimensional diagram.

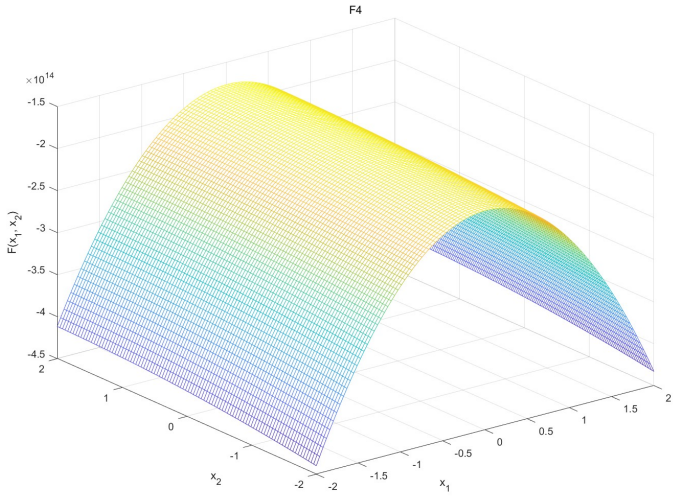

Figure 16. f_4 three-dimensional diagram.

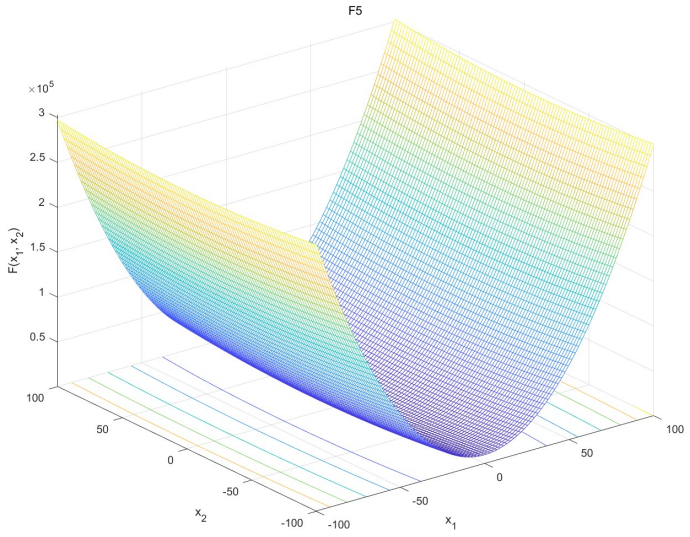

Figure 17. f_5 three-dimensional diagram.

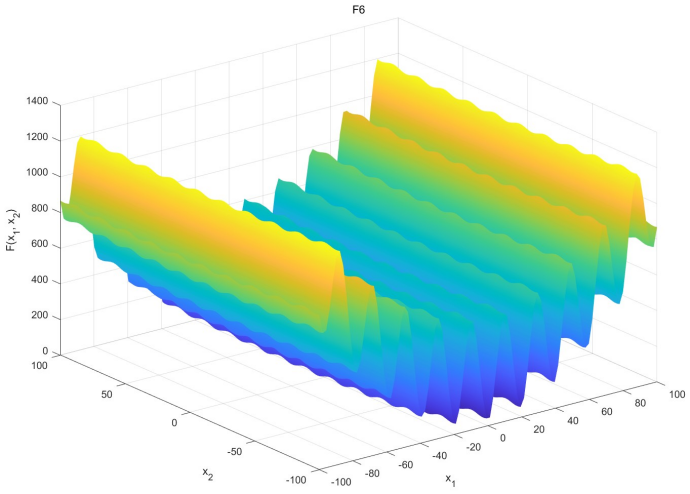

Figure 18. f_6 three-dimensional diagram.

4.3. Algorithm optimization accuracy analysis

In this section, the multi-policy upgraded Harris Hawk optimization algorithm (UHHO) is experimented with seven other algorithms in six benchmark functions with the parameter settings shown in Table 5. The experimental results are shown in Table 6.

Table 5. Algorithm parameter setting table.

Arithmetic	Parameterization
UHHO	Energy factor E_0: $[-1, 1]$
DBO	Producer scale: $[0.1, 0.3]$; Roll angle θ: $[0, \frac{\pi}{2}]$
GWO	Convergence factor a: $[0, 2]$
SMA	Inertia weights W: $[0.5, 1]$
FA	Attractivity coefficient γ: $[0.1, 1]$
BWO	Predation coefficient a: $[0.1, 1]$
WOA	Spiral coefficient $b = 1$
HHO	Energy factor E_0: $[-1, 1]$

Table 6. Statistical table of test function results.

F	Norm	UHHO	DBO	GWO	SMA	FA	BWO	WOA	HHO
	Best	0.00e+00	8.82e−08	0.00e+00	0.00e+00	3.16e−03	2.22e−16	0.00e+00	0.00e+00
f_1	Mean	5.57e−15	1.03e−03	0.00e+00	0.00e+00	4.43e−02	1.09e−03	3.13e−04	0.00e+00
	Std	2.45e−14	1.88e−03	0.00e+00	0.00e+00	4.18e−02	1.53e−03	9.62e−04	0.00e+00
	Best	0.00e+00	2.11e+03	9.09e−217	8.53e−09	4.45e+04	3.18e−05	1.00e−190	0.00e+00
f_2	Mean	7.57e−144	6.79e+03	6.53e−213	6.40e−08	5.51e+04	5.68e−05	1.70e−186	0.00e+00
	Std	5.90e−144	1.47e+03	0.00e+00	1.91e−08	4.63e+03	9.15e−05	0.00e+00	0.00e+00
	Best	3.98e−01	3.98e−01	3.98e−01	3.98e−01	3.98e−01	3.98e−01	3.98e−01	3.98e−01
f_3	Mean	3.98e−01	3.98e−01	3.98e−01	3.98e−01	3.98e−01	3.98e−01	3.98e−01	3.98e−01
	Std	1.99e−06	4.25e−09	1.91e−08	1.42e−07	3.00e−10	6.28e−14	3.23e−08	2.02e−15
	Best	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00
f_4	Mean	1.11e+01	3.00e+00	3.00e+00	3.00e+00	3.00e+00	4.35e+00	3.00e+00	3.00e+00
	Std	2.49e+01	6.35e−08	9.51e−07	6.11e−07	1.81e−08	6.04e+00	6.60e−08	7.14e−07
	Best	8.73e−06	2.58e+03	2.12e−06	1.58e−05	4.69e+04	1.37e−05	3.23e−05	1.62e−05
f_5	Mean	1.33e−03	1.29e+04	4.99e−02	3.44e−05	5.54e+04	9.03e−05	2.42e−04	1.51e−04
	Std	1.90e−03	5.19e+03	1.31e−01	1.35e−05	4.69e+03	6.16e−05	2.69e−04	1.86e−04
	Best	0.00e+00	2.90e+01	0.00e+00	4.40e+01	2.75e+02	9.65e+01	0.00e+00	0.00e+00
f_6	Mean	0.00e+00	5.36e+01	1.57e−01	8.42e+01	2.87e+02	1.71e+02	0.00e+00	0.00e+00
	Std	0.00e+00	1.79e+01	7.03e−01	2.27e+01	6.21e+00	3.77e+01	0.00e+00	0.00e+00

From the point of view of optimal value, mean value, and standard deviation, it can be seen from the data in Table 6 that the UHHO algorithm obtains the theoretical optimal value for solving the functions f_1, f_2, f_3, f_4, and f_6. Although for function f_5, the theoretical optimal value is not obtained, it is the best quality and closest to the theoretical value compared to the other seven algorithms. In terms of mean and standard deviation, the UHHO shows strong stability compared to the results of the other algorithms solving the test functions separately.

It shows that the upgraded HHO algorithm overcomes the problem of poor optimization accuracy of the standard HHO algorithm, greatly improves the optimization performance of the basic algorithm, and has an obvious competitive advantage compared with other algorithms. Therefore, the good point set strategy, nonlinear energy factor strategy, sine-cosine strategy, and L-C chaos strategy introduced by the HHO algorithm in this paper effectively enhance the local search and global search ability, which greatly improves the performance of the optimization search and makes the optimization search effect of the UHHO algorithm better than other algorithms.

4.4. Algorithm convergence analysis

The experimental results below demonstrate the convergence plots of UHHO and the other seven algorithms on the six benchmark functions.

The above figure shows the optimization performance comparison between the UHHO algorithm and the other seven algorithms on different functions, with the horizontal coordinate being the number of iterations and the vertical coordinate being the optimized function values.

(1) Convergence speed: As can be seen in Figures 19–24, the UHHO curve decreases faster in the early stage, indicating that in the optimization of these functions, the UHHO can quickly approach the better solution, compared to the other algorithms, which shows the ability to find the better solution faster in the early stage of the iteration;

(2) Convergence accuracy: As can be seen from Figures 19 and 24, the UHHO algorithm has

relatively low values of the functions in the later stages of the iteration, which indicates that the UHHO algorithm is able to converge to a better solution in these functions, and has an advantage in accuracy compared to other algorithms.

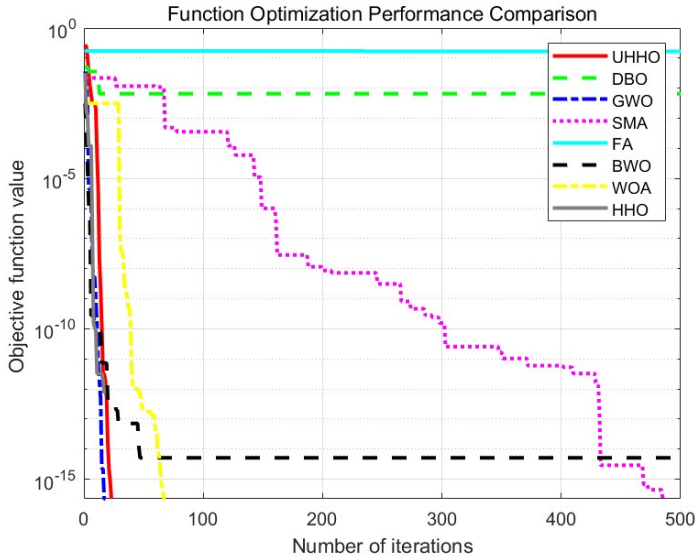

Figure 19. Convergence curves for UHHO and other algorithms for solving f_1.

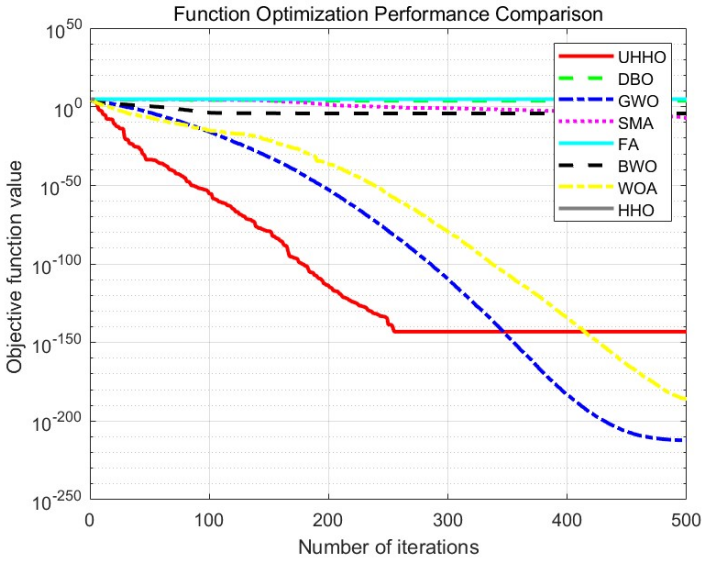

Figure 20. Convergence curves for UHHO and other algorithms for solving f_2.

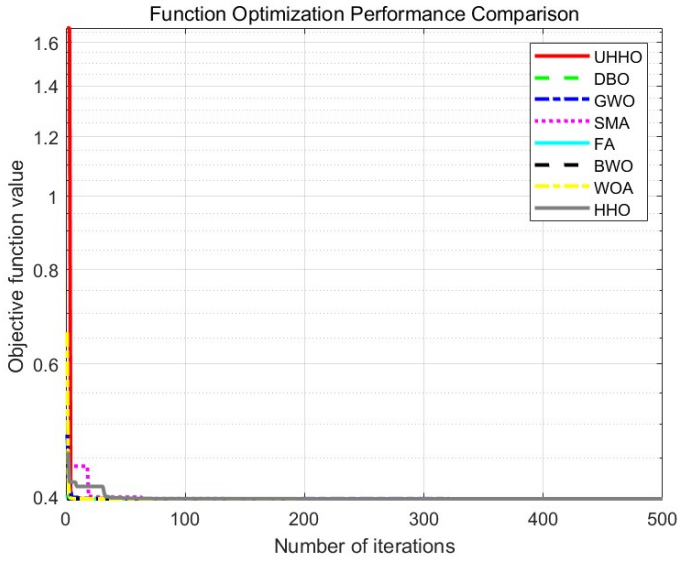

Figure 21. Convergence curves for UHHO and other algorithms for solving f_3.

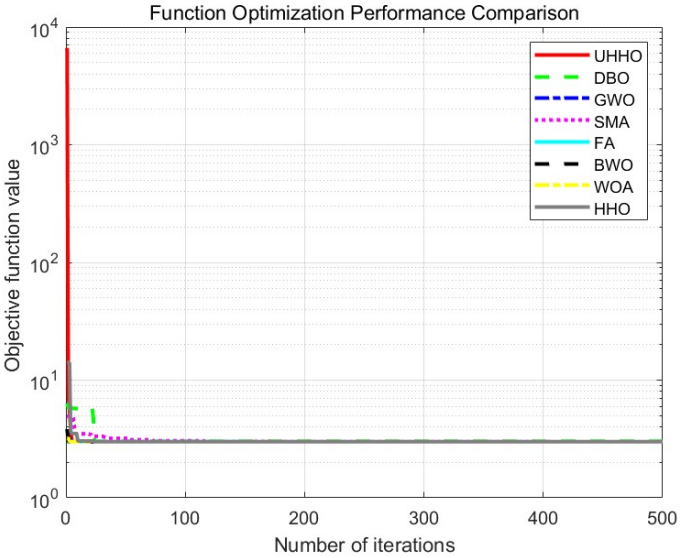

Figure 22. Convergence curves for UHHO and other algorithms for solving f_4.

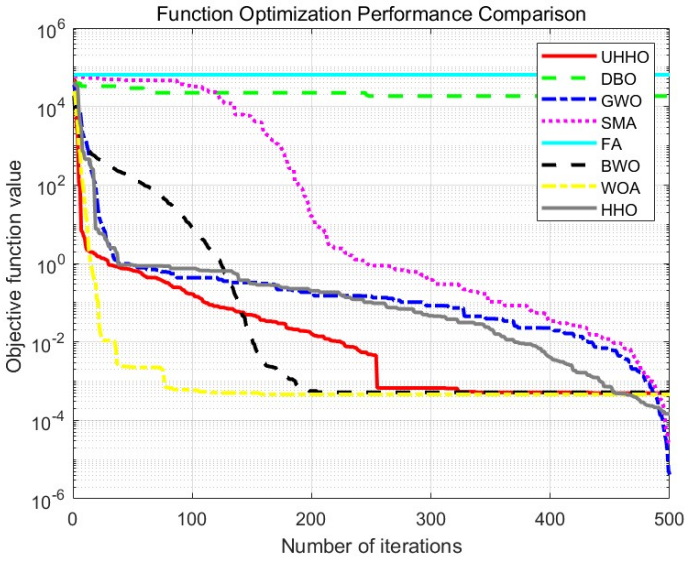

Figure 23. Convergence curves for UHHO and other algorithms for solving f_5.

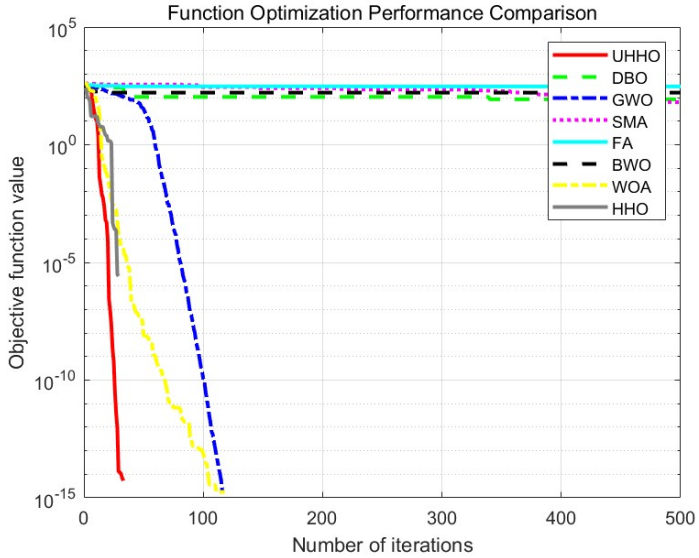

Figure 24. Convergence curves for UHHO and other algorithms for solving f_6.

Overall, compared with other comparative algorithms, the UHHO algorithm has certain advantages in terms of convergence speed and convergence accuracy, and performs more stably in the search process with better overall optimization performance. This fully demonstrates that the multi-strategy approach proposed in this paper has a significant effect on improving the convergence performance of the HHO algorithm.

5. Numerical experiments

In order to verify that the multi-strategy UHHO algorithm is more effective than the standard HHO algorithm in solving nonlinear inequality-constrained optimization problems, and to further show that the UHHO algorithm can avoid the HHO algorithm from falling into local optimums to a certain extent, in this section, the following three numerical examples of nonlinear inequality constrained optimization problems are considered and solved using the UHHO algorithm and the HHO algorithm, respectively. During the experiments, the population size is taken to be $N = 400$, the maximum number of iterations $T = 500$, each algorithm is run independently 20 times, and the optimal results are taken for comparison.

5.1. Numerical experiment 1 [22]

$$\begin{aligned} &min\ f(x) = 1.57x_1\sqrt{x_2^2 + 5776}, \\ &s.t.\ \frac{15\sqrt{5776 + x_2^2}}{0.785x_1x_2} \le 7.03,\ 0.001 \le x_1 \le 7, \\ &\frac{15\sqrt{5776 + x_2^2}}{0.785x_1x_2} \le \frac{(1295 + 20730x_1^2)}{(46208 + 8x_2^2)},\ 0.001 \le x_2 \le 187.98. \end{aligned} \tag{5.1}$$

The problem (5.1) is next solved using the multi-strategy UHHO algorithm and the standard HHO algorithm, respectively, and the results of the numerical experiments are shown in Table 7 below.

From the experimental data in Table 7, it can be seen that the UHHO algorithm gives results closer to the theoretical optimum than the HHO, indicating that the UHHO algorithm is more feasible in solving the nonlinear optimization problem (5.1). Figures 25 and 26 below show the graph of the optimal solution of the optimization problem solved by the UHHO algorithm and the convergence curves, respectively.

Combined with Figures 25 and 26, the graphs show that the convergence curve of the UHHO algorithm decreases rapidly and stabilizes during the iteration process, indicating that it converges faster and stabilizes on a better solution.

Table 7. Results of solving for UHHO and HHO separately.

Serial No.	Arithmetic	x_1	x_2	Minimum value
1	UHHO	4.7975	52.2557	694.6923
2	HHO	4.9392	50.0909	705.8421

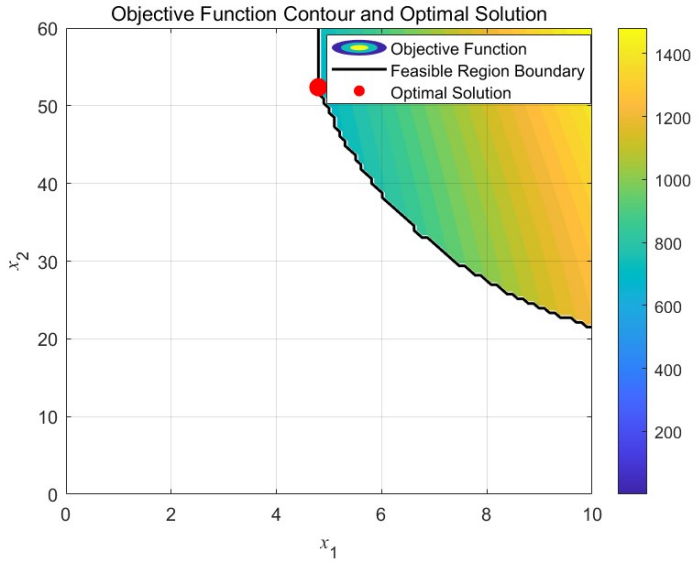

Figure 25. Relationship diagram between the optimal solution and optimization problem.

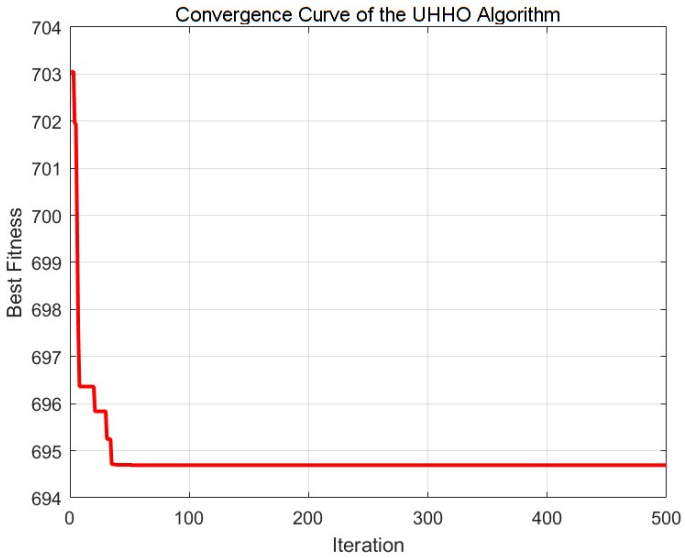

Figure 26. Convergence plot of the solution results.

5.2. Numerical experiment 2 [23]

$$\begin{aligned} &min\ f(x) = 100(2\sqrt{2}x_1 + x_2), \\ &s.t.\ 2\frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} - 2 \le 0, 0 \le x_1 \le 1, \\ &\quad 2\frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2} - 2 \le 0, 0 \le x_2 \le 1, \\ &\quad 2\frac{1}{\sqrt{2}x_2 + x_1} - 2 \le 0. \end{aligned} \tag{5.2}$$

Subsequently, this paper solves the problem (5.2) using the multi-strategy upgraded Harris Hawk algorithm and the standard Harris Hawk algorithm, respectively, and the results of the numerical experiments are shown in Table 8 below.

Table 8. Results of solving for UHHO and HHO separately.

Serial No.	Arithmetic	x_1	x_2	Minimum value
1	UHHO	0.7735	0.4530	264.0746
2	HHO	0.8245	0.3154	264.7334

From the experimental data in Table 8, it can be seen that the UHHO algorithm gives results closer to the theoretical optimum than the HHO, indicating that the UHHO algorithm is more feasible in solving the nonlinear optimization problem (5.2). Figures 27 and 28 below show the graph of the optimal solution of the optimization problem solved by the UHHO algorithm as well as the convergence curve, respectively.

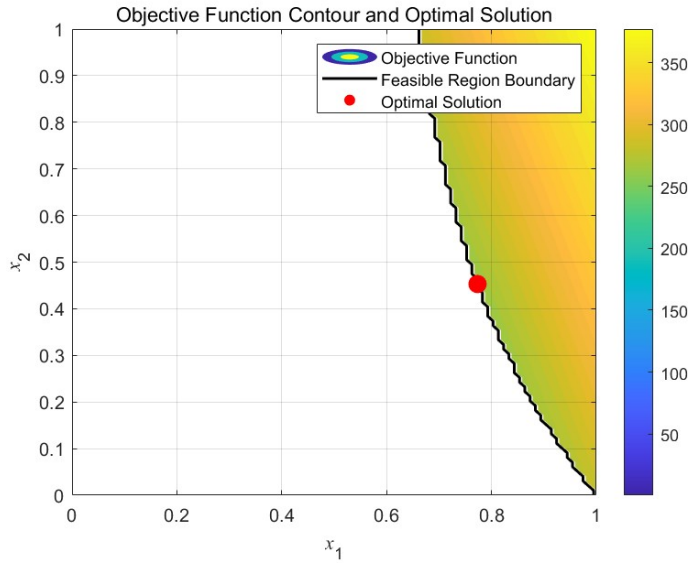

Figure 27. Relationship diagram between the optimal solution and optimization problem.

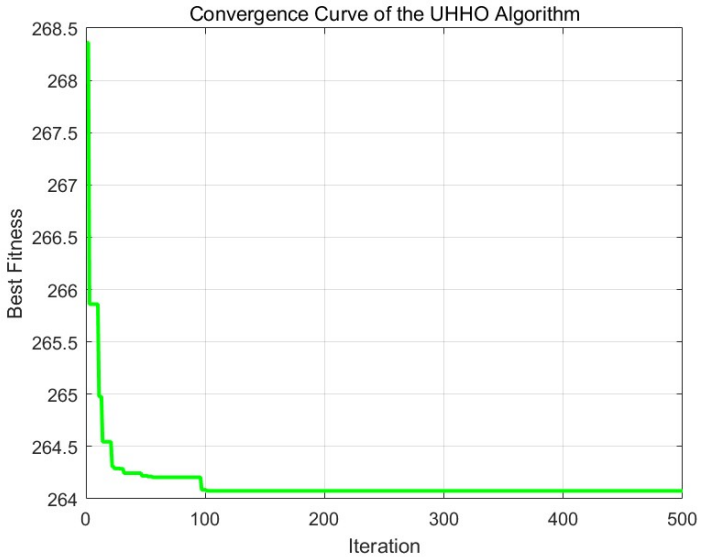

Figure 28. Convergence plot of the solution results.

From the figure, it is obvious that the convergence curve of the UHHO algorithm decreases faster during the iteration process and the fitness value is close to the theoretical optimal value, which indicates that the algorithm converges faster.

5.3. Numerical experiment 3 [24]

$$\begin{aligned} &min\ f(x) = 13x_1^2x_2, \\ &s.t.\ 1 - \frac{11x_2^3}{71785x_1^4} \le 0, 0.05 \le x_1 \le 2, \\ &\quad \frac{4x_2^2 - x_1x_2}{12566(x_1^3x_2 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \le 0, \\ &\quad 1 - \frac{140.45x_1}{11x_2^2} \le 0, 0.25 \le x_2 \le 1.3, \\ &\quad \frac{2(x_1 + x_2)}{3} - 1 \le 0. \end{aligned} \tag{5.3}$$

In this paper, the problem is next solved independently using UHHO and HHO, and the results of the simulation experiment data are shown in Table 9 below.

Table 9. Results of solving for UHHO and HHO separately.

Serial No.	Arithmetic	x_1	x_2	Minimum value
1	UHHO	0.0519	0.3618	0.0127
2	HHO	0.0520	0.3646	0.0128

As can be seen from Table 9, UHHO produces solutions closer to the theoretical optimum than HHO, and the multi-strategy upgraded HHO solution is slightly better than the HHO algorithm.

Figures 29 and 30 show the graph of the optimal solution to the optimization problem (5.3) as well as the convergence profile, respectively.

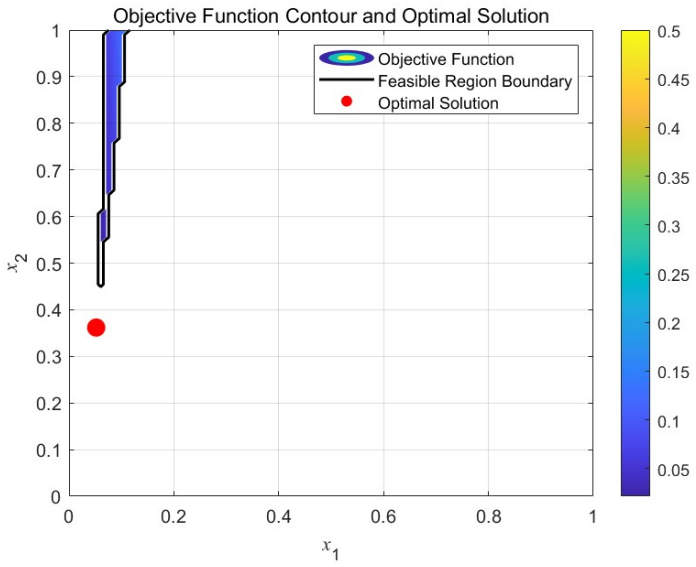

Figure 29. Relationship diagram between the optimal solution and optimization problem.

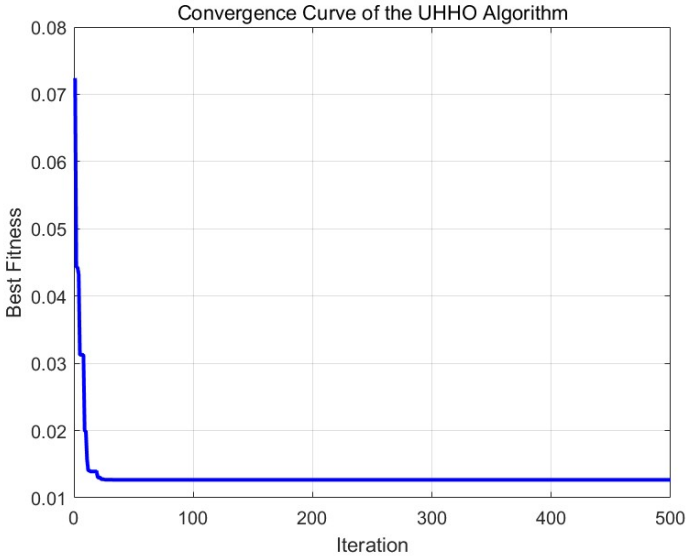

Figure 30. Convergence plot of the solution results.

Combining Figures 29 and 30, it can be seen that the convergence curve decreases faster, which indicates that the algorithm converges faster.

The three numerical experiments demonstrate that UHHO outperforms HHO in convergence.

Specifically, UHHO achieves smaller objective function values, indicating higher solution accuracy. Moreover, its convergence curve shows faster and more stable convergence, reaching a satisfactory solution with fewer iterations. These results confirm UHHO's effectiveness in solving nonlinear optimization problems.

6. Conclusions

When addressing optimization problems, the Harris Hawk optimization algorithm tends to encounter issues such as becoming trapped in local optima, so this paper combines the principle of the Harris Hawk algorithm to upgrade it, and puts forward the upgraded Harris Hawk optimization algorithm, which is mainly designed to make up for the shortcomings of the local optimization and other defects. The upgraded Harris Hawk algorithm is used to test the six benchmark functions, and synthesis data confirms that the upgraded algorithm improves the solution accuracy. In addition, the algorithm can be seen from the convergence curves to have better convergence, and finally the three numerical experiments given show that the upgraded algorithm has a very good effect in solving the nonlinear optimization problems.

Author contributions

Juhe Sun: Methodology, Conceptualization, Writing—review; Guolin Huang: Conceptualization, Writing—original draft, editing; Li Wang: Methodology, Conceptualization; Chuanjun Yin: Conceptualization; Ning Ma: Conceptualization, Funding acquisition.

Use of Generative-AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (Project No. 52475482).

Conflict of interest

The authors declare that they have no conflicts of interest.

References

1. H. Faris, A. M. Al-Zoubi, A. A. Heidari, I. Aljarah, M. Mafarja, M. A. Hassonah, et al., An intelligent system for spam detection and identification of the most relevant features based on evolutionary Random Weight Networks, *Inform. Fusion*, **48** (2019), 67–83. https://doi.org/10.1016/j.inffus.2018.08.002
2. R. Abbassi, A. Abbassi, A. A. Heidari, S. Mirjalili , An efficient salp swarm-inspired algorithm for parameters identification of photovoltaic cell models, *Energ. Convers. Manage*, **179** (2019), 362–372. https://doi.org/10.1016/j.enconman.2018.10.069
3. G. H. Wu, Across neighborhood search for numerical optimization, *Inform. Sci.*, **329** (2016), 597–618. https://doi.org/10.1016/j.ins.2015.09.051
4. J. Nocedal, S. J. Wright, *Numerical optimization*, New York: Springer, 2006. https://doi.org/10.1007/978-0-387-40065-5
5. V. S. Mikhalevich, I. V. Sergienko, N. Z. Shor, Investigation of optimization methods and their applications, *Cybern. Syst. Anal.*, **17** (1981), 522–548. https://doi.org/10.1007/BF01082482
6. X. S. Xiang, Y. Tian, X. Y. Zhang, J. H. Xiao, Y. C. Jin, A pairwise proximity learning-based ant colony algorithm for dynamic vehicle routing problems, *IEEE T. Intell. Transp.*, **23** (2022), 5275–5286. https://doi.org/10.1109/TITS.2021.3052834
7. M. B. Ghanamijaber, A hybrid fuzzy-PID controller based on gray wolf optimization algorithm in power system, *Evol. Syst.*, **10** (2019), 273–284. https://doi.org/10.1007/s12530-018-9228-x
8. A. Raza, M. Zhong, Hybrid lane-based short-term urban traffic speed forecasting: A genetic approach, In: *2017 4th International Conference on Transportation Information and Safety (ICTIS)*, 2017, 271–279. https://doi.org/10.1109/ICTIS.2017.8047776
9. M. Dorigo, G. D. Caro, Ant colony optimization: a new meta-heuristic, In: *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, 1999. https://doi.org/10.1109/CEC.1999.782657
10. T. M. Shami, A. A. El-Saleh, M. Alswaitti, M. Alswaitti, Q. Al-Tashi, M. A. Summakieh, et al., Particle swarm optimization: A comprehensive survey, *IEEE Access*, **10** (2022), 10031–10061. https://doi.org/10.1109/ACCESS.2022.3142859
11. P. J. Denning, Genetic algorithms, *Am. Sci.*, **80** (1992), 12–14.
12. X. S. Yang, Firefly algorithms for multimodal optimization, In: *Stochastic algorithms: Foundations and applications*, 2009. https://doi.org/10.1007/978-3-642-04944-6_14
13. Z. D. Wang, L. L. Huang, S. X. Yang, D. H. Li, D. J. He, S. Chan, A quasi-oppositional learning of updating quantum state and Q-learning based on the dung beetle algorithm for global optimization, *Alex. Eng. J.*, **81** (2023), 469–488. https://doi.org/10.1016/j.aej.2023.09.042
14. A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. L. Chen, Harris Hawks optimization: Algorithm and applications, *Future Gener. Comp. Sy.*, **97** (2019), 849–872. https://doi.org/10.1016/j.future.2019.02.028

15. M. Issa, A. Samn, Passive vehicle suspension system optimization using Harris Hawk Optimization algorithm, *Math. Comput. Simul.*, **191** (2022), 328–345. https://doi.org/10.1016/j.matcom.2021.08.016

16. K. Balakrishnan, R. Dhanalakshmi, U. M. Khaire, A novel control factor and Brownian motion-based improved Harris Hawks optimization for feature selection, *J. Ambient Intell. Human. Comput.*, **14** (2023), 8631–8653. https://doi.org/10.1007/s12652-021-03621-y

17. M. H. Qais, H. M. Hasanien, S. Alghuwainem, Paraments extraction of three-diode photovoltaic model using computation and Harris Hawks optimization, *Energy*, **195** (2020), 117040. https://doi.org/10.1016/j.energy.2020.117040

18. S. Barshandeh, F. Piri, S. R. Sangani, HMPA: An innovative hybrid multi-population algorithm based on artificial ecosystem-based and Harris Hawks optimization algorithms for engineering problems, *Eng. Comput.*, **38** (2022), 1581–1625. https://doi.org/10.1007/s00366-020-01120-w

19. Y. X. Chen, X. M. Liang, Y. F. Huang, Improved quantum particle swarm optimization based on good-point set, *J. Cent. South Univ. (Sci. Technol.)*, **44** (2013), 1409–1414.

20. L. Z. Duan, S. Q. Yang, D. B. Zhang, The optimization of feature selection based on chaos clustering strategy and niche particle swarm optimization, *Math. Probl. Eng.*, **2020** (2020), 3138659. https://doi.org/10.1155/2020/3138659

21. G. Y. Wang, F. Yuan, Cascade chaos and its dynamic characteristics, *Acta Phys. Sin.*, **62** (2013), 020506. https://doi.org/10.7498/aps.62.020506

22. J. H Huang, Application of optimization design methods to mechanics of materials, *J. Jimei Univ. (Nat. Sci. Ed.)*, **02** (1999), 33–37. https://doi.org/10.19715/j.jmuzr.1999.02.007

23. T. Ray, P. Saini, Engineering design optimization using a swarm with an intelligent information sharing among individuals, *Eng. Optim.*, **33** (2001), 735–748. https://doi.org/10.1080/03052150108940941

24. A. Sadollah, A. Bahreininejad, H. Eskandar, M. Hamdi, Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems, *Appl. Soft Comput.*, **13** (2013), 2592–2612. https://doi.org/10.1016/j.asoc.2012.11.026

AIMS Press