Harris hawks optimization based on global cross-variation and tent mapping

Lei Chen1 · Na Song2 · Yunpeng Ma1

Accepted: 3 October 2022 / Published online: 25 October 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
Harris hawks optimization (HHO) is a new meta-heuristic algorithm that builds a model by imitating the predation process of Harris hawks. In order to solve the problems of poor convergence speed caused by uniform choice position update formula in the exploration stage of basic HHO and falling into local optimization caused by insufficient population richness in the later stage of the algorithm, a Harris hawks optimization based on global cross-variation and tent mapping (CRTHHO) is proposed in this paper. Firstly, the tent mapping is introduced in the exploration stage to optimize random parameter q to speed up the convergence in the early stage. Secondly, the crossover mutation operator is introduced to cross and mutate the global optimal position in each iteration process. The greedy strategy is used to select, which prevents the algorithm from falling into local optimal because of skipping the optimal solution and improves the convergence accuracy of the algorithm. In order to investigate the performance of CRTHHO, experiments are carried out on ten benchmark functions and the CEC2017 test set. Experimental results show that the CRTHHO algorithm performs better than the HHO algorithm and is competitive with five advanced meta-heuristic algorithms.

Keywords Harris hawks optimization · Meta-heuristic algorithm · Crossover mutation · Tent mapping · Greedy selection
1 Introduction

Optimization refers to searching for the optimal solution among all possible solutions for a particular problem [1]. In scientific computing, medical testing, financial analysis, and other fields, optimization problems exist in large numbers and become more and more complex [2–5]. The traditional optimization methods include gradient descent, Newton’s method, quasi-Newton methods, conjugate gradient, and Lagrange multiplier Method, which is used to solve constraint optimization problems. Although the traditional optimization methods have the advantages of relatively perfect theory and small computation, they have obvious limitations when applied to complex and challenging optimization problems. Compared with conventional methods, the meta-heuristic algorithm has the advantages of simple operation, substantial flexibility, and no need for gradient information. It has more advantages in solving large-scale and complicated optimization problems [6, 7].

Meta-heuristic algorithms can be divided into the following two categories according to the different inspired mechanisms: Firstly, algorithms based on biological evolution and physical laws in nature, such as genetic algorithm (GA) [8], differential evolution algorithm (DE) [9], multi-verse optimization (MVO) [10] and atom search optimization (ASO) [11], are based on biological evolution mechanism. Secondly, algorithms based on social behaviors of social living organisms, such as particle swarm optimization (PSO) [12] and artificial swarm algorithm (ABC) [13], are also known as swarm intelligence optimization algorithms. In recent years, scholars have proposed many novel meta-heuristic algorithms, such as biogeography-based optimization (BBO) [14], dragonfly algorithm (DA) [15], sine–cosine algorithm (SCA) [16], moth flame optimization (MFO) [17], arithmetic optimization algorithm (AOA) [18], weighted mean of vectors (INFO) [19], coronavirus optimization algorithm (COVIDOA) [20], Runge–Kutta optimizer (RUN) [21], and artificial hummingbird algorithm (AHA) [22]. These novel optimization algorithms have been applied to many problems, such as fault diagnosis and feature selection. In 2019, Heidari et al. [23] proposed a new swarm intelligence optimization algorithm, Harris hawks optimization (HHO). The algorithm is based on the simulation of the hunting behavior of the Harris hawks, which is the most intelligent bird at present. The algorithm integrates the concepts of a population center, random population division, and escape energy and has a good solving ability for most continuous unconstrained optimization problems [24]. Moreover, the Harris hawks optimization has few parameters to adjust, and it is easy to operate and implement. So it has attracted extensive attention once proposed. As an excellent new swarm intelligence optimization algorithm, HHO has been used to solve optimization problems in many fields. For example, Ahmad Abbasi et al. [25] use HHO algorithm in minimizing the entropy generation of the microchannel and it has a superior performance. Mali Satya et al. [26] use Harris hawks optimization (HHO) to offer a maximum power point tracking technique (MPPT) for photovoltaic (PV) powered e-Vehicles. Mashaleh et al. [27] used HHO algorithm to improve machine learning algorithm for spam detection. In addition, HHO and its improved algorithm have also achieved good application results in image recognition [28].
machine optimization [29], neural network optimization [30] in the computer field, photovoltaic system modeling [31], constraint engineering optimization [32–34], rainfall modeling [35], and other problems in the engineering field.

However, the HHO algorithm still has some problems that the convergence rate is not fast enough in the exploration stage, and it is easy to fall into local optimum in the later stage. In view of these problems, scholars have proposed many improvements to the HHO algorithm. Song Shiming et al. [36] introduced Gaussian mutation strategy and dimension decision strategy in cuckoo search into the HHO algorithm, which is conducive to fully mining the solutions in the search area. Li Chenyang et al. [37] proposed an exploration strategy based on logarithmic spiral and opposition learning, and integrated it into HHO algorithm to improve the exploration ability of HHO. Shijie Zhao et al. [38] proposed an optimized HHO algorithm combining the regulation mechanism of periodic energy decline of prey and Newton’s local enhancement strategy, which enhanced the mining ability of the algorithm. Yuxin Guo et al. [39] introduced the elite reverse learning mechanism and golden sine algorithm to improve population diversity and reduce algorithm convergence time. Xiaolong Liu et al. [40] balanced the exploration and development of the algorithm through the multi-subgroup square neighborhood topology and fixed permutation probability. Oian Fan et al. [41] proposed a new quasi-reflection Harris hawks optimization (QRHHO), which combined the HHO algorithm with a quasi-reflection-based learning mechanism (QRBL) to improve the optimization accuracy of the HHO algorithm. Hussain et al. [42] introduced long-term memory into the HHO algorithm, referring to the past individual location information to increase the diversity of the population in the search process. Yiming Ma et al. [43] used maximum likelihood estimation to improve the algorithm’s fitness function and improve the solving accuracy.

Although these improved algorithms can improve the detection ability or avoid premature convergence to some extent, according to the No Free Lunch theorem, no algorithm can achieve the best effect on all optimization problems. Therefore, given the difficulties of slow convergence speed and local optimal stagnation in the early stage of the HHO algorithm, it is still necessary to continue studying and improving the strategy. In this work, two strategies are added into the Harris hawks optimization algorithm to better balance exploration and development. Tent mapping is considered to adjust the value of a random parameter, so that the ergodicity and regularity of chaotic mapping can be used to reduce unnecessary consumption in the exploration process of Harris hawks, and then make the convergence accelerated. The global cross-mutation strategy is added to make full use of the information of the eagle group, avoid skipping the optimal solution, and enhance the local exploration ability of the eagle group in the later stage. By modifying the search equation of HHO with the above two strategies, the search mechanism of HHO is improved, and the exploration and development of the algorithm are better balanced. This paper proposes a Harris hawks optimization based on global cross-variation and tent mapping (CRTHHO), and its performance is tested on ten benchmark functions and the CEC2017 test set. The algorithm’s improvement strategy and performance test will be discussed in detail in 3 and 4.
2 Harris hawks optimization (HHO)

Harris hawks optimization (HHO) is a novel meta-heuristic algorithm, proposed by Heidari in 2019, solving complicated combinatorial optimization problem. It is inspired by the hunting and attacking behavior of Harris hawks and its typical feature is swarm intelligence. This optimization algorithm consists of two stages: the exploration stage and the exploitation phase, and each phase mimics the different behaviors of hawks during predation. The parameter $E$ represents the escape energy of prey, which decreases over time, and its calculation formula is as follows:

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

where $E_0 = 2r_1 - 1$ represents the initial state of energy, $r_1$ is a random number between $(0, 1)$, and $E_0$ varies randomly between $(-1, 1)$ during each iteration. $T$ is the maximum iteration number, and $t$ is the current iteration number.

The Harris hawks choose different capture strategies depending on the escape energy $E$ of the prey. When $|E| > 1$, Harris hawks are in the exploration phase. Otherwise, they are in the exploitation phase.

2.1 Exploration phase

During the exploration phase, Harris hawks search for prey using two strategies to update their position based on the random location of rabbits and other flock members. Its mathematical model is as follows:

$$x(t+1) = \begin{cases} x_{\text{rand}}(t) - r_1 |x_{\text{rand}}(t) - 2r_2 x(t)|, & q \geq 0.5 \\ (x_{\text{prey}}(t) - x_m(t)) - r_3 [L_B + r_4 (U_B - L_B)], & q < 0.5 \end{cases}$$  \hspace{1cm} (2)

where $t$ represents the current iteration number, $T$ represents the maximum iteration number, $N$ represents the population number, $r_1, r_2, r_3, r_4$, and $q$ are random numbers between $(0, 1)$, which are updated in each iteration, $U_B$ and $L_B$ show the upper and lower bounds of the search space, $x(t+1)$ is the position vector of hawks in the next iteration, $x_{\text{prey}}(t)$ represents the current optimal position (the position of prey), $x(t)$ is the current position vector of hawks, $x_{\text{rand}}$ is a randomly selected hawk from the current population of hawks, and $x_m(t)$ represents the average position of the current population of hawks. The average position of hawks is calculated by the following formula:

$$x_m(t) = \frac{1}{N} \sum_{i=1}^{N} x_i(t)$$  \hspace{1cm} (3)

where $x_i(t)$ indicates the location of each hawk in iteration $t$. 

[Image: Springer logo]
2.2 Exploitation phase

In the game of chasing and escaping between hawks and their prey, the escape energy of prey decreases gradually. When $|E| < 1$, the Harris hawks enter the exploitation phase, and the captors further pursue the prey and try to capture it. HHO mimics this predatory behavior by using four strategies: soft besiege, hard besiege, soft besiege with progressive rapid dive, and hard besiege with progressive rapid dive. HHO confirms which strategy to use through the escape energy $E$ and the escape probability $r$.

When $|E| \geq 0.5, r \geq 0.5$, carry out the soft besiege. In this stage, the prey still has the energy to escape, and the eagles use a weak encircle strategy to deplete it and launch a surprise attack. This behavior is modeled by the following rules:

\[ x(t + 1) = \Delta x(t) - E|Jx_{\text{prey}}(t) - x(t)| \]  
\[ \Delta x(t) = x_{\text{prey}}(t) - x(t) \]  
\[ J = 2(1 - r_5) \]

where $\Delta x(t)$ represents the difference between the prey position and the current position of the Harris hawks in iteration $t$, $r_5$ is a random number between $0, 1$, and $J$ represents the random jump strength of the rabbit throughout the process of escaping. The $J$ value changes randomly in each iteration to simulate the nature of rabbit motions.

When $|E| < 0.5, r \geq 0.5$, the Harris hawks carry out the hard besiege. In this stage, the prey has no energy to escape, and the Harris hawks use a hard encircling to hunt the prey for a final assault. The current position is updated as follows:

\[ x(t + 1) = x_{\text{prey}}(t) - E|\Delta x(t)| \]

When $|E| \geq 0.5, r < 0.5$, the Harris hawks carry out the soft besiege with progressive rapid dive. In this stage, the prey has enough energy to escape from the eagles and the Harris hawks will gradually dive and soft surround the target. In order to simulate the escape mode and jumping action of the prey, the levy function to the position update process. This process is modeled as follows:

\[ Y = x_{\text{prey}}(t) - E|Jx_{\text{prey}}(t) - x(t)| \]
\[ Z = Y + S \times LF(D) \]
\[ x(t + 1) = \begin{cases} Y, & \text{if } F(Y) < F[X(t)] \\ Z, & \text{if } F(Z) < F[X(t)] \end{cases} \]

where $D$ is the problem dimension and $S$ is the $D$-dimensional random row vector.

When $|E| < 0.5, r < 0.5$, carry out the hard besiege with progressive rapid dive. In this stage, the prey is exhausted and has low escape energy, so the Harris hawks surround the prey through the hard besiege with progressive rapid dive. They attempt to
reduce the distance between their average position and the target prey. The position update formula is as follows:

\[ x(t + 1) = \begin{cases} 
Y, & \text{if } F(Y) < F(X(t)) \\
Z, & \text{if } F(Z) < F(X(t)) 
\end{cases} \] (11)

where \(Y\) and \(Z\) are updated by formula (12) and (13):

\[ Y = x_{\text{prey}}(t) - E[Jx_{\text{prey}}(t) - x_m(t)] \] (12)

\[ Z = Y + S \times LF(D) \] (13)

The pseudocode of the basic HHO algorithm is shown in Algorithm 1.

---

**Algorithm 1** The Pseudo-code of HHO algorithm

**Input:** The population size \(N\) and maximum number of iterations \(T\)

**Output:** The location of rabbit and its fitness value

1. Initialize the random population \(x_i(i = 1, 2, \cdots, N)\)
2. \(E_0 = 2\text{rand}() - 1, J = 2(1 - \text{rand}())\)
3. while (stopping condition is not met) do
   4. Calculate the fitness values of hawks
   5. Set \(x_{\text{prey}}\) as the location of rabbit (best location)
   6. for each hawks \(x_i\) do
      7. Update the initial energy \(E\) and jump strength \(J\)
      8. Update the \(E\) using Eq.(1)
      9. if \(|E| \geq 1\) then
         10. Update the location vector using Eq.(2);
      11. else \(|E| < 1\) then
         12. if \(|E| \geq 0.5, r \geq 0.5\) then
            13. Update the location vector using Eq.(4);
         14. else if \(|E| < 0.5, r \geq 0.5\) then
            15. Update the location vector using Eq.(7);
         16. else if \(|E| \geq 0.5, r < 0.5\) then
            17. Update the location vector using Eq.(10);
         18. else if \(|E| < 0.5, r < 0.5\) then
            19. Update the location vector using Eq.(11);
      20. end if
   21. end if
   22. end for
   23. end while
3 Harris hawks optimization based on global cross-variation and tent mapping

In this section, the proposed algorithm CRTHHO has been described in detail. Tent mapping is introduced to solve the problem of slow convergence in the exploration phase. In order to solve the problem of low convergence accuracy in the later stage of the algorithm, a global crossover mutation is proposed. These two methods will be elaborated on below. The HHO algorithms that introduce tent mapping, global cross-variation alone, and combine the two strategies are named THHO, CRHHO, and CRTHHO.

3.1 Harris hawks optimization based on tent mapping (THHO)

Chaos mapping has the characteristics of randomness, ergodicity, and regularity and is often used to generate the initial population of the algorithm or as a disturbance in the algorithm process [12]. Because of the slow convergence speed of Harris hawks optimization in the early stage, the tent mapping is introduced into the exploration phase of the algorithm, which can significantly accelerate the optimization progress in the early stage of the search. The specific introduction method is as follows:

In the original Harris hawks optimization, the exploration stage of the algorithm is realized by Eq. (2), where $q$ is a random number between (0, 1), which is a simple uniform distribution. In the improved THHO algorithm, the tent map [44] is used to adjust the value of an arbitrary parameter $q$, which can better use the ergodicity and regularity of the chaotic map so that Harris hawks can reduce unnecessary consumption in the exploration process. Thus, it can accelerate the convergence speed. Therefore, the position update formula of the exploration process in THHO is changed to:

$$
x_i(t + 1) = \begin{cases} 
x_{rand} - r_1 |x_{rand} - 2r_2 x_i|, & q_i \geq 0.5 \\
x_{prey}(t) - x_m(t) - r_3 [lb_i + r_4 (ub_i - lb_i)], & q_i < 0.5 \end{cases}$$

(14)

where $q_i$ is constantly changing in the iteration process, and it is updated through tent mapping. The formula of $q_i$ is as follows [44]:

$$q_{i+1} = \begin{cases} q_i / 0.6, & \text{if } 0 < q_i \leq 0.6 \\
(1 - q_i) / 0.4, & \text{if } q_i > 0.6 \end{cases}$$

(15)

3.2 Harris hawks optimization based on global cross-variation (CRHHO)

When Harris hawks optimization enters the later stage, the search scope will be smaller and it is easier to fall into local optimum. Therefore, a global crossover mutation strategy was introduced. After completing a global iterative optimization,
the current position \((x_1(t), x_2(t), \ldots, x_N(t))\) cross-mutates the optimal position \(x_{\text{prey}}(t)\). This strategy can make full use of the information of the eagle group, avoid skipping the optimal solution, and enhance the local exploration ability of the eagle group in the later period. After crossover and mutation, the greedy selection strategy is used to update the optimal location, ensuring the exploration ability and avoiding diversity spillover. Using crossover mutation and greedy selection to improve the HHO algorithm can avoid local optimization in the later stage to enhance the optimization effect.

The crossover mutation strategy is realized by a crossover operator. Set the crossover operator to hybridize the parent solution to produce children that contain the characteristics of the parent solution. Now the crossover work used is defined as follows [45]:

\[
u_i(t) = \begin{cases} v_i^j(t), & \text{if } r_j < CR \\ x_{\text{prey}}^j(t), & \text{otherwise} \end{cases}
\] (16)

where \(CR\) is the crossover rate. \(v_i(t)\) is selected from \(x_1(t), x_2(t), \ldots, x_N(t)\) randomly, \(x_{\text{prey}}(t)\) is currently the best position for the eagles. \(r_j\) is the uniformly distributed random number between \((0, 1)\), \(j = 1, 2, \ldots, D\). In the present work, the crossover probability \(CR\) is taken as 0.3.

Greedy selection compares the fitness of the updated vector with the current solution vector to determine whether the updated vector will survive in the next generation. After crossover and mutation, whether to update the optimal position of prey is determined by the greedy selection, and the specific operation is as follows [45]:

\[
x_{\text{prey}}(t) = \begin{cases} u_i(t), & \text{if } F(u_{i,t}) \leq F(x_{\text{prey},t}) \\ x_{\text{prey}}(t), & \text{if } F(u_{i,t}) > F(x_{\text{prey},t}) \end{cases}
\] (17)

where \(F\) is the objective function, and the above selection process describes the selection of the minimization problem. Through cross-variation and greedy selection, it can effectively avoid falling into the local optimum and reduce the possibility of stagnation.

### 3.3 Harris hawks optimization combined the two strategies (CRTHHO)

The tent chaos diagram is introduced into the THHO algorithm, and the position updating formula of the original algorithm is changed in the exploration stage so that it has a faster convergence rate in the early stage and improves the optimization performance of the algorithm. CRHHO guides the current optimal individual to mutate through the crossover mutation operator and updates the optimal position of the current population through greedy selection to avoid falling into local optimal at the later stage of the algorithm and improve the convergence accuracy of the algorithm. A CRTHHO algorithm is proposed to combine the above two improved methods into the HHO algorithm. The CRTHHO contains the advantages of the above two improved strategies. It is expected that CRTHHO can not only converge at a faster speed in the early stage but also avoid local optimization to achieve better
accuracy in the later stage. The pseudocode of CRTHHO is shown in Algorithm 2, and the flowchart is shown in Fig. 1. The performance of the CRTHHO algorithm is described in 4.

Algorithm 2 Pseudocode of CRTHHO algorithm

Input: The population size $N$ and maximum number of iterations $T$
Output: The location of rabbit and its fitness value
1: Initialize the random population $x_i (i = 1, 2, \ldots, N)$
2: $E_0 = 2 \text{rand}() - 1, J = 2(1 - \text{rand}())$
3: while (stopping condition is not met) do
4:   Calculate the fitness values of hawks
5:   for each hawk($x_i$) do
6:     Update the initial energy $E$ and jump strength $J$
7:     Update the $E$ using Eq.(1)
8:     if $|E| \geq 1$ then
9:       Update the location vector using Eq.(14);
10:      else $|E| < 1$
11:      if $|E| \geq 0.5, r \geq 0.5$ then
12:          Update the location vector using Eq.(4);
13:      else if $|E| < 0.5, r \geq 0.5$ then
14:          Update the location vector using Eq.(7);
15:      else if $|E| \geq 0.5, r < 0.5$ then
16:          Update the location vector using Eq.(10);
17:      else if $|E| < 0.5, r < 0.5$ then
18:          Update the location vector using Eq.(11);
19:      end if
20:   end if
21: end if
22: Set crossover operator $CR = 0.3$
23: Update the prey location using Eq.(17)
24: end for
25: end while

4 Experimental simulation and result analysis

In order to verify the optimization performance of CRTHHO, two kinds of experiments are designed in this paper: (1) comparing the improved HHO and HHO; and (2) comparing CRTHHO with five mainstream meta-heuristic algorithms. The experiment selected ten benchmark test functions for testing, among which F1–F4 is a high-dimensional single-peak function, and F5–F10 is a high-dimensional multi-peak function. The function with only one extreme point in the feasible region is called the single-peak function, which can test the algorithm’s local development
ability and convergence speed. The function with multiple local optimal values in the feasible region is called the multi-peak function, which can better test the power of the optimization algorithm to jump out of the local optimal value. The detailed expressions, dimensions, feasible regions, and target values of the functions are shown in Table 1. In addition, the CRTHHO algorithm’s performance is also tested in CEC2017, which will be explained in detail in Sect. 4.2.

In order to ensure the fairness of experimental results, the population size of all algorithms was set as 30, and parameter settings of HHO and improved HHO are consistent with those in the literature [23]. At the same time, to reduce the experiment’s randomness, the experimental results are the average values after 30 independent runs. Because this experiment is limited by computer software, the minimum achievable accuracy is 2.2251E−308. Therefore, when the convergence accuracy is lower than 2.2251E−308, the result is 0. For ease of observation and comparison, the best results of each test function are shown in bold in the table. The ordinate of drawing is taken as $\log_{10}(fitness\_value)$.

4.1 Compare improved HHOs and basic HHO

The first part of the experiment is divided into two parts: 1) Compare the convergence accuracy and speed with fixed iteration times and 2) compare iteration times with fixed target precision. In this experiment, the population size and maximum iteration times of the improved algorithm THHO, CRHHO, CRTHHO, and the basic

![Fig. 1 Flowchart of CRTHHO](image)

---

Harris hawks optimization based on global cross-variation…
algorithm HHO are set to 30 and 500, respectively, and the dimension of the test function F1-F10 is 30. The final experimental result is the average of 30 independent running results.

4.1.1 Compare the convergence accuracy and speed with fixed iteration times

When the number of iterations is fixed at 500, the test results of HHO, THHO, CRHHO, and CRTHHO on ten benchmark test functions are shown in Table 2. Figure 2 shows the convergence curves of these four algorithms on ten benchmark functions.

In Table 2, the first row of each function records the average of the 30 independent runs and the second standard deviation of behavior. As can be seen from the data in Table 2, in the above 10 test functions, the three improved algorithms, THHO, CRHHO, and CRTHHO, are superior to the original HHO algorithm, and except for F3, the CRTHHO algorithm combining the two improved strategies is superior to the other two algorithms introducing only one approach. It shows that both systems improve the performance of HHO, and the algorithm combining the two methods can better improve the performance of the original HHO algorithm. The convergence precision of the four algorithms on F5, F7, and F8 is very close to the theoretical optimal value, and they have the same convergence precision on F5-F8.

| Number | Function | Dim | Range | Target |
|--------|----------|-----|-------|--------|
| F1     | \( f(x) = \sum_{i=1}^{n} x_i^2 \) | 30/50/100 | [-100, 100] | 0 |
| F2     | \( f(x) = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i| \) | 30/50/100 | [-10, 10] | 0 |
| F3     | \( f(x) = \sum_{i=1}^{n} \left( \frac{i}{\prod_{j=1}^{i} x_j} \right) \) | 30/50/100 | [-100, 100] | 0 |
| F4     | \( f(x) = \max \left\{ |x_i|, 1 \leq i \leq n \right\} \) | 30/50/100 | [-100, 100] | 0 |
| F5     | \( f(x) = \sum_{i=1}^{n} (x_i^2 - 10(\cos 2\pi x_i) + 10) \) | 30/50/100 | [-5.12, 5.12] | 0 |
| F6     | \( f(x) = 20 + e - 20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2}) \) | 30/50/100 | [-32, 32] | 0 |
| F7     | \( f(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos \left( \frac{x_i}{\sqrt{\frac{n}{2}}} \right) + 1 \) | 30/50/100 | [-600, 600] | 0 |
| F8     | \( f(x) = \sum_{i=1}^{n} \left( x_i^2 - 10(\cos 2\pi y_i) + 10 \right) \) | 30/50/100 | [-5.12, 5.12] | 0 |
| F9     | \( f(x) = \left( \frac{1}{n} \sum_{i=1}^{n} x_i^2 \right)^{0.25} \left( \sin^3 \left( 50 \left( \frac{1}{n} \sum_{i=1}^{n} x_i^2 \right)^{0.1} \right) + 1 \right) \) | 30/50/100 | [-100, 100] | 0 |
| F10    | \( f(x) = x_1^2 + 10^6 \sum_{i=2}^{n} x_i^2 \) | 30/50/100 | [-100, 100] | 0 |
Therefore, we can further analyze the performance difference of the four algorithms through the convergence curve.

As shown in Fig. 2, the convergence curves of F1, F2, F4, F9, and F10 clearly show that compared with HHO, THHO has significantly improved the convergence speed in the early stage and the convergence accuracy in the late stage. Compared with HHO, CRHHO has no apparent advantage in the early stage but can jump out of local optimum in the late stage and improve the convergence accuracy. Compared with HHO, CRTHHO has both the advantages of the former two, accelerating the convergence speed in the early stage and improving the convergence accuracy in the later stage. Similarly, compared with F5-F8, it can be seen that although the absolute convergence precision is the same, the convergence speed of CRTHHO is significantly better than that of HHO.

4.1.2 Compare iteration times with fixed target precision

In order to further study the effectiveness of the improved strategy, this section compares the number of iterations required by HHO, THHO, CRHHO, and CRTHHO with the same target accuracy (the experimental results are rounded). Table 3 shows the target accuracy of each function in the experiment and the number of iterations required by each algorithm to achieve the accuracy.

By analyzing the data in Table 3, it can be seen that for the above 10 test functions, within the maximum number of iterations, 6 of the basic HHO algorithm, 3 of the THHO algorithm, and 2 of the CRHHO algorithm cannot achieve the target accuracy. In contrast, the CRTHHO combined with the two improved strategies can achieve the target accuracy. Among the above four algorithms, CRTHHO has the highest success rate of reaching the target accuracy within 500 times. In addition, a horizontal comparison of the results of each test function shows that CRTHHO can achieve the same accuracy as HHO, THHO, and CRHHO with fewer iterations. In summary, CRTHHO significantly improves the performance of the HHO algorithm.

4.2 Compare CRTHHO with 5 advanced algorithms

In order to increase the difficulty of the experiment, the second kind of experiment compares the optimization performance of CRTHHO and five mainstream meta-heuristic algorithms on different dimensions of 10 benchmark functions and CEC2017 test functions. The five algorithms selected in the experiment are as follows: particle swarm optimization (PSO, 1995) [12], biogeography-based optimization (BBO, 2009) [14], dragonfly algorithm (DA, 2016) [15], sine–cosine algorithm (SCA, 2016) [16], and the arithmetic optimization algorithm (AOA, 2021) [18]. PSO is a classic swarm intelligence optimization algorithm. BBO, DA, and SCA are excellent optimization algorithms which are proposed and widely used in recent years. AOA is a novel algorithm proposed in 2021. And the performance of them is outstanding on benchmark functions and real engineering problems. Therefore, the performance of proposed CRTHHO algorithm can be proved by comparing
Fig. 2 Convergence curves of improved HHO and HHO fixed iterations
with these five algorithms. The initial parameters of these algorithms are shown in Table 4.

4.2.1 Compare CRTHHO with 5 advanced algorithms on basic test functions

In this section, five novel optimization algorithms are selected to perform performance tests with CRTHHO to verify that the performance of the proposed CRTHHO
### Table 3  Comparison of iteration times under fixed target accuracy

| Function | Limitation | Item | HHO | THHO | CRHHO | CRTHHO |
|----------|------------|------|-----|------|-------|--------|
| F1       | 1.E−150    | Mean | 500 | 376  | 383   | 340    |
|          |            | Max  | 500 | 500  | 407   | 351    |
|          |            | Min  | 500 | 348  | 375   | 331    |
|          |            | Std  | 0   | 0.9  | 1     | 1      |
| F2       | 1.E−100    | Mean | 500 | 500  | 451   | 379    |
|          |            | Max  | 500 | 500  | 500   | 388    |
|          |            | Min  | 500 | 500  | 406   | 370    |
|          |            | Std  | 0   | 0    | 0.6   | 1      |
| F3       | 1.E−100    | Mean | 500 | 317  | 500   | 317    |
|          |            | Max  | 500 | 338  | 500   | 343    |
|          |            | Min  | 500 | 290  | 500   | 290    |
|          |            | Std  | 0   | 1    | 0     | 1      |
| F4       | 1.E−100    | Mean | 500 | 500  | 500   | 456    |
|          |            | Max  | 500 | 500  | 500   | 500    |
|          |            | Min  | 500 | 500  | 500   | 389    |
|          |            | Std  | 0   | 0    | 0     | 0.53   |
| F5       | 1.E−100    | Mean | 187 | 51   | 184   | 41     |
|          |            | Max  | 224 | 64   | 232   | 53     |
|          |            | Min  | 120 | 37   | 135   | 29     |
|          |            | Std  | 1   | 1    | 1     | 1      |
| F6       | 1.E−15     | Mean | 256 | 96   | 259   | 85     |
|          |            | Max  | 270 | 114  | 273   | 107    |
|          |            | Min  | 221 | 78   | 238   | 67     |
|          |            | Std  | 1   | 1    | 1     | 1      |
| F7       | 1.E−100    | Mean | 217 | 60   | 216   | 59     |
|          |            | Max  | 236 | 72   | 239   | 95     |
|          |            | Min  | 163 | 43   | 187   | 45     |
|          |            | Std  | 1   | 1    | 1     | 1      |
| F8       | 1.E−100    | Mean | 198 | 54   | 208   | 47     |
|          |            | Max  | 231 | 68   | 230   | 60     |
|          |            | Min  | 81  | 43   | 163   | 34     |
|          |            | Std  | 1   | 1    | 1     | 1      |
| F9       | 1.E−50     | Mean | 500 | 500  | 467   | 385    |
|          |            | Max  | 500 | 500  | 500   | 403    |
|          |            | Min  | 500 | 500  | 408   | 367    |
|          |            | Std  | 0   | 0    | 0.4   | 1      |
| F10      | 1.E−150    | Mean | 500 | 413  | 388   | 348    |
|          |            | Max  | 500 | 500  | 424   | 357    |
|          |            | Min  | 500 | 353  | 380   | 333    |
|          |            | Std  | 0   | 0.67 | 1     | 1      |
algorithm is excellent among meta-heuristic algorithms. The test functions are ten basic test functions, as shown in Table 1. At the same time, in order to test its performance in higher dimensions, the experiment in this section is divided into three groups for comparative experiments in dimensions 30, 50, and 100, respectively. The experimental results are shown in Table 5, and the convergence curves are shown in Figs. 3 and 4.

Table 5 shows the convergence accuracy of 6 optimization algorithms on three different dimensions of 10 benchmark functions. Figure 3 shows the convergence curves of the six optimization algorithms on the benchmark function when the dimension is 30. In comparison, the convergence curves of the six optimization algorithms when the size is 100 are shown in Fig. 4. Combined with the data in Table 5 and the convergence curves in Figs. 3 and 4, the following analysis is carried out.

By comparing the optimization effects of the algorithm on different functions in the same dimension, it can be seen that: At 30 and 50 sizes, CRTHHO’s convergence accuracy is optimal for F1 and F3-F10, and the optimization result of CRTHHO is close to the theoretical optimal value at F5, F7, and F8. In contrast, the convergence accuracy of CRTHHO is not as good as that of AOA at F2. However, Fig. 3 shows that the convergence speed of the CRTHHO algorithm is higher than that of the AOA algorithm. When the dimension is 100, CRTHHO’s convergence accuracy is superior to the other five algorithms for F1-F10, and Fig. 4 shows that the convergence speed of CRTHHO is also faster than other algorithms.

By comparing the optimization effects of the algorithms in different dimensions of the same function, it can be concluded that: When the dimension of the function increases, the convergence accuracy of the same optimization algorithm will decrease, but the CRTHHO convergence accuracy changes little or even improves. This can be obtained by comparing the accuracy changes of AOA and CRTHHO as the dimension increases. Compared with AOA, the accuracy of CRTHHO decreased in F1, F3, F4, F9, and F10, but the variation range was small. In F2, with the increase of dimension, the accuracy of AOA decreased significantly, from close to the theoretical optimal value to 3.3E−44, while the accuracy of CRTHHO remained stable at about 1E−110 and even increased slightly. On F6, AOA’s accuracy dropped by 12 orders of magnitude as the dimension went up to 100, but CRTHHO’s accuracy remained 8.9E−16.

Table 4 Parameter Settings of algorithms

| Algorithm | Pop size | Max generation | Parameters |
|-----------|----------|----------------|------------|
| DA        | 30       | 500            | $w \in [0.4, 0.9]; s = 0.1; a = 0.1; c = 0.7; f = 1; e = 1$ |
| SCA       | 30       | 500            | $a = 2$    |
| PSO       | 30       | 500            | $w \in [0.3, 0.8]; c_1 = c_2 = 2$ |
| BBO       | 30       | 500            | $MOP_{Max} = 1; MOP_{Min} = 0.2; C_{Iter} = 1; \text{Alpha} = 5; \text{Mu} = 0.499$ |
| AOA       | 30       | 500            | $KeepRate = 0.2; alpha = 0.9; pMutation = 0.1$ |
| CRTHHO    | 30       | 500            | $CR = 0.3$ |
| Function | Dim | Item   | DA       | SCA     | PSO     | BBO     | AOA     | CRTHHO |
|----------|-----|--------|----------|---------|---------|---------|---------|--------|
|          |     | Mean   | 2.2E+03  | 7.4E+00 | 2.5E+04 | 2.5E−02 | 2.0E−25 | 1.8E−20 |
| F1       | 30  | Std    | 1.2E+03  | 9.2E+00 | 6.1E+03 | 5.5E−03 | 1.1E−24 | 0.0E+00 |
|          | 50  | Mean   | 8.3E+03  | 6.6E+02 | 4.6E+04 | 2.8E−01 | 3.1E−04 | 3.5E−197|
|          |     | Std    | 4.8E+03  | 6.5E+02 | 1.6E+04 | 3.8E−02 | 1.0E−03 | 0.0E+00 |
|          | 100 | Mean   | 2.3E+04  | 1.1E+04 | 1.1E+05 | 2.3E+00 | 2.5E−02 | 2.0E−209|
|          |     | Std    | 1.4E+04  | 6.9E+03 | 2.9E+04 | 2.5E−01 | 9.4E−03 | 0.0E+00 |
|          | 30  | Mean   | 1.5E+01  | 2.2E−02 | 8.9E+10 | 5.1E−01 | 0.0E+00 | 2.4E−109|
|          |     | Std    | 5.9E+00  | 4.3E−02 | 2.3E+11 | 6.5E−02 | 0.0E+00 | 1.3E−108|
|          | 50  | Mean   | 8.3E+03  | 6.6E+02 | 4.6E+04 | 2.8E−01 | 3.1E−04 | 3.5E−197|
|          |     | Std    | 4.8E+03  | 6.5E+02 | 1.6E+04 | 3.8E−02 | 1.0E−03 | 0.0E+00 |
|          | 100 | Mean   | 2.3E+04  | 1.1E+04 | 1.1E+05 | 2.3E+00 | 2.5E−02 | 2.0E−209|
|          |     | Std    | 1.4E+04  | 6.9E+03 | 2.9E+04 | 2.5E−01 | 9.4E−03 | 0.0E+00 |
| F2       | 30  | Mean   | 1.5E+01  | 2.2E−02 | 8.9E+10 | 5.1E−01 | 0.0E+00 | 2.4E−109|
|          |     | Std    | 5.9E+00  | 4.3E−02 | 2.3E+11 | 6.5E−02 | 0.0E+00 | 1.3E−108|
| F3       | 30  | Mean   | 1.3E+04  | 1.0E+04 | 5.8E+04 | 4.8E+00 | 3.6E−03 | 2.4E−121|
| F4       | 30  | Mean   | 3.2E+01  | 3.7E+01 | 5.89E+01 | 1.6E−01 | 2.8E−02 | 1.1E−94 |
|          |     | Std    | 9.6E+00  | 1.2E+01 | 8.3E+00 | 2.0E−02 | 1.9E−02 | 4.6E−94 |
|          | 50  | Mean   | 4.2E+01  | 6.7E+01 | 6.9E+01 | 4.6E−01 | 4.8E−02 | 3.7E−93 |
|          |     | Std    | 7.2E+00  | 7.1E+01 | 1.0E+01 | 1.0E−01 | 6.4E−03 | 2.0E−92 |
|          | 100 | Mean   | 5.9E+01  | 8.9E+01 | 7.7E+01 | 2.2E+00 | 9.5E−02 | 1.7E−94 |
|          |     | Std    | 5.3E+00  | 3.0E+00 | 9.2E+00 | 2.1E−01 | 1.4E−02 | 8.9E−94 |
| F5       | 30  | Mean   | 1.7E+02  | 3.8E+01 | 4.1E+02 | 4.8E+01 | 0.0E+00 | 0.0E+00 |
|          |     | Std    | 3.8E+01  | 3.5E+01 | 3.4E+01 | 1.1E+01 | 0.0E+00 | 0.0E+00 |
|          | 50  | Mean   | 3.4E+02  | 1.1E+02 | 7.4E+02 | 1.6E+02 | 0.0E+00 | 0.0E+00 |
|          |     | Std    | 6.7E+01  | 5.2E+01 | 5.9E+01 | 2.3E+01 | 0.0E+00 | 0.0E+00 |
|          | 100 | Mean   | 8.0E+02  | 2.5E+02 | 1.5E+03 | 5.6E+02 | 0.0E+00 | 0.0E+00 |
|          |     | Std    | 1.4E+02  | 1.3E+02 | 1.3E+02 | 5.6E+01 | 0.0E+00 | 0.0E+00 |
| F6       | 30  | Mean   | 1.0E+01  | 1.3E+01 | 1.9E+01 | 1.4E−01 | 8.9E−16 | 8.9E−16 |
|          |     | Std    | 1.8E+00  | 8.9E+00 | 7.9E−01 | 2.3E−02 | 0.0E+00 | 0.0E+00 |
|          | 50  | Mean   | 1.3E+01  | 1.6E+01 | 1.9E+01 | 5.1E−01 | 8.9E−16 | 8.9E−16 |
|          |     | Std    | 2.2E+00  | 6.7E+00 | 6.2E−01 | 6.2E−02 | 0.0E+00 | 0.0E+00 |
|          | 100 | Mean   | 1.4E+01  | 1.8E+01 | 2.0E+01 | 1.3E+00 | 6.8E−04 | 8.9E−16 |
|          |     | Std    | 2.5E+00  | 5.1E+00 | 4.2E−01 | 9.6E−02 | 1.2E−03 | 0.0E+00 |
| F7       | 30  | Mean   | 1.8E+01  | 8.9E−01 | 1.1E+02 | 5.4E−03 | 1.7E−01 | 0.0E+00 |
|          |     | Std    | 9.9E+00  | 3.3E−01 | 4.1E+01 | 5.1E−03 | 1.4E−01 | 0.0E+00 |
|          | 50  | Mean   | 8.0E+01  | 7.4E+00 | 2.2E+02 | 1.1E−02 | 1.1E+00 | 0.0E+00 |
|          |     | Std    | 3.5E+01  | 8.0E+00 | 7.2E+01 | 4.8E−03 | 4.4E−01 | 0.0E+00 |
|          | 100 | Mean   | 2.0E+02  | 9.4E+01 | 4.1E+02 | 4.3E−02 | 6.9E+02 | 0.0E+00 |
|          |     | Std    | 1.1E+02  | 6.4E+01 | 6.8E+01 | 7.8E−03 | 2.1E+02 | 0.0E+00 |

Table 5: Comparison of the convergence accuracy with other algorithms on the benchmark function
In conclusion, compared with the above five advanced algorithms, CRTHHO has apparent advantages in convergence accuracy and convergence speed and has a stable solution accuracy for high-dimensional problems.

### 4.2.2 Compare CRTHHO and 5 advanced algorithms on the CEC2017 test set

CEC2017 [46] is a set of challenging test functions, the details of which are shown in Table 6. The test functions in CEC2017 are characterized by rotation and translation. These functions are complicated, and it is difficult to find the optimal solution. Therefore, the number of iterations in this section is set to 1000. Considering the time cost and computer performance constraints, the function dimension is selected as ten dimensions. We compared CRTHHO with five advanced algorithms on the CEC2017 test set in this experiment. The experimental results are shown in Table 7. Figure 5 shows the convergence curves of the six algorithms on 30 test functions. In addition, the difference between the proposed algorithm and other algorithms is analyzed by the rank-sum test.

The rank-sum test is a nonparametric statistical method used to compare whether the two are significantly different. In this experiment, the rank-sum test is performed on the other five algorithms, respectively, with CRTHHO, and the significance level

| Function | Dim | Item | DA   | SCA   | PSO   | BBO   | AOA   | CRTHHO |
|----------|-----|------|------|-------|-------|-------|-------|--------|
| F8       | 30  | Mean | 1.6E+02 | 8.8E+01 | 3.9E+02 | 2.7E+01 | 0.0E+00 | 0.0E+00 |
| Std      |     | 4.3E+01 | 3.8E+01 | 3.4E+01 | 6.4E+00 | 0.0E+00 | 0.0E+00 |
| 50       | Mean | 3.0E+02 | 1.7E+02 | 7.1E+02 | 1.3E+02 | 0.0E+00 | 0.0E+00 |
| Std      |     | 6.5E+01 | 6.0E+01 | 5.2E+01 | 3.1E+01 | 0.0E+00 | 0.0E+00 |
| 100      | Mean | 7.6E+02 | 4.3E+02 | 1.5E+03 | 4.9E+02 | 0.0E+00 | 0.0E+00 |
| Std      |     | 1.3E+02 | 1.7E+02 | 1.2E+02 | 5.2E+01 | 0.0E+00 | 0.0E+00 |
| F9       | 30  | Mean | 9.7E+00 | 5.1E+00 | 1.4E+01 | 4.0E+00 | 1.9E−04 | 3.7E−55 |
| Std      |     | 1.5E+00 | 1.4E+00 | 1.3E+00 | 2.9E−01 | 1.1E−03 | 1.5E−54 |
| 50       | Mean | 1.2E+01 | 9.8E+00 | 1.7E+01 | 4.9E+00 | 3.0E−01 | 5.9E−55 |
| Std      |     | 1.4E+00 | 1.6E+00 | 1.3E+00 | 3.0E−01 | 1.3E−01 | 3.2E−54 |
| 100      | Mean | 1.4E+01 | 1.5E+01 | 2.0E+01 | 6.5E+00 | 6.4E−01 | 7.2E−55 |
| Std      |     | 1.8E+00 | 1.1E+00 | 1.6E+00 | 2.4E−01 | 8.2E−02 | 2.7E−54 |
| F10      | 30  | Mean | 1.8E+09 | 1.9E+07 | 2.2E+10 | 2.0E+04 | 8.7E−19 | 2.2E−216 |
| Std      |     | 1.1E+09 | 3.3E+07 | 8.2E+09 | 4.3E+03 | 4.6E−18 | 0.0E+00 |
| 50       | Mean | 7.3E+09 | 7.2E+08 | 4.4E+10 | 2.6E+05 | 6.1E+02 | 2.4E−212 |
| Std      |     | 3.9E+09 | 8.0E+08 | 1.5E+10 | 3.7E+04 | 1.2E+03 | 0.0E+00 |
| 100      | Mean | 1.8E+10 | 1.2E+10 | 1.1E+11 | 2.4E+06 | 2.7E+04 | 5.7E−206 |
| Std      |     | 1.0E+10 | 7.8E+09 | 3.2E+10 | 2.3E+05 | 9.3E+03 | 0.0E+00 |
Fig. 3 Convergence curves of 6 algorithms on 10 benchmark functions (30 dims)
The results and ranking of each algorithm are shown in Table 8. $p$ value is the result of rank-sum test. When $p$ value is less than 0.05, the result is significant, indicating that the algorithm is significantly different from the CRTHHO algorithm.

Table 7 shows the optimization results of CRTHHO and the other five optimization algorithms on CEC2017, where mean and STD represent the mean and standard deviation of the results of 30 independent runs, respectively. According to the data in Table 7, it can be seen that CRTHHO has the optimal convergence accuracy on the 17 functions F1, F3–F6, F8, F10–13, F15, F17, F20, F22–23, F25, and F30, and is suboptimal on the five functions F16, F18–19, and F23–24. Therefore, CRTHHO can achieve superior results on more than seventy percent of the tasks in the CEC2017 test set. Figure 5 shows the convergence curves of CRTHHO and the other five advanced algorithms on CEC2017. Combined with Fig. 5, it can be seen that CRTHHO has a fast convergence speed in the early stage and is faster than other algorithms on more than half of the functions. The rank-sum test results and ranking of each algorithm are shown in Table 8, where $p$ value is used to determine whether the algorithm is significantly different from CRTHHO, rank is a ranking of the six algorithms according to convergence accuracy, $+$ means that the algorithm is significantly different from and better than CRTHHO, $=$ means that the algorithm is not significantly different from CRTHHO, and $-$ indicates that the algorithm is significantly different from and inferior to CRTHHO. Combining Table 8, we can see that CRTHHO ranks first in 17 functions. In addition, Table 8 gives the comprehensive ranking of the six algorithms on the 30 functions, and CRTHHO also ranks first. In conclusion, CRTHHO outperforms other algorithms in half or more functions of the CEC test set, which fully proves the effectiveness of the proposed algorithm in complex optimization problems.
Fig. 4 Convergence curves of 6 algorithms on 10 benchmark functions (100 dims)
4.3 Comparison of CRTHHO with other improved HHOs

To further compare the performance of CRTHHO, it is compared with other improved HHO algorithms on CEC2017. LMHHO [41] is an improved algorithm proposed by Hussain K et al in 2019, while GCHHO [35] is proposed by Song et al in 2021. The test results of CRTHHO, LMHHO, GCHHO, and HHO on CEC2017 are shown in Table 9, and the optimal results have been indicated in bold. By comparison the test results, it can be seen that CRTHHO has 20 optimal results and 5 suboptimal results on 29 test functions except F2. It can be concluded that the proposed algorithm is competitive with other improvements for HHO.

4.4 Real-world optimization problems

In order to study the performance of CRTHHO in real-world engineering applications, we tested CRTHHO with PSO, SCA, BBO, DA, and AOA on four engineering design problems. Due to the low dimensionality of these four engineering problems, compression spring design, pressure vessel design, speed reducer design, and string design, we set the maximum number of iterations and the population to 50 and 30, respectively. Moreover, the final statistical result of each algorithm is the average of 30 independent runs.

4.4.1 Compression spring design

The goal of compression spring design (CSD) [47] is to minimize its mass while satisfying certain constraints, including four inequality constraints: minimum deflection, shear stress, oscillation frequency, and outer diameter limit. The average coil diameter, the diameter of the spring wire, and the effective coil number of the spring are three design variables. The problem can be described as:
| Class                | Number | Functions                                      | Target |
|---------------------|--------|-----------------------------------------------|--------|
| Simple              | 1      | Shifted and Rotated Bent Cigar Function        | 100    |
| Unimodal            | 2      | Shifted and Rotated Sum of Different Power Function* | 200    |
| Functions           | 3      | Shifted and Rotated Zakharov Function         | 300    |
|                     | 4      | Shifted and Rotated Rosenbrock’s Function     | 400    |
|                     | 5      | Shifted and Rotated Rastrigin’s Function      | 500    |
| Simple              | 6      | Shifted and Rotated Expanded Scaffer’s F6 Function | 600    |
| Multimodal          | 7      | Shifted and Rotated Lunacek Bi_Rastrigin Function | 700    |
| Functions           | 8      | Shifted and Rotated Non-Continuous Rastrigin’s Function | 800    |
|                     | 9      | Shifted and Rotated Levy Function             | 900    |
|                     | 10     | Shifted and Rotated Schwefel’s Function       | 1000   |
|                     | 11     | Hybrid Function 1 (N=3)                      | 1100   |
|                     | 12     | Hybrid Function 2 (N=3)                      | 1200   |
|                     | 13     | Hybrid Function 3 (N=3)                      | 1300   |
|                     | 14     | Hybrid Function 4 (N=4)                      | 1400   |
| Hybrid              | 15     | Hybrid Function 5 (N=4)                      | 1500   |
| Functions           | 16     | Hybrid Function 6 (N=4)                      | 1600   |
|                     | 17     | Hybrid Function 7 (N=5)                      | 1700   |
|                     | 18     | Hybrid Function 8 (N=5)                      | 1800   |
|                     | 19     | Hybrid Function 9 (N=5)                      | 1900   |
|                     | 20     | Hybrid Function 10 (N=6)                     | 2000   |
|                     | 21     | Composition Function 1 (N=3)                 | 2100   |
|                     | 22     | Composition Function 1 (N=3)                 | 2200   |
|                     | 23     | Composition Function 1 (N=4)                 | 2300   |
|                     | 24     | Composition Function 1 (N=4)                 | 2400   |
|                     | 25     | Composition Function 1 (N=5)                 | 2500   |
| Composition         | 26     | Composition Function 1 (N=5)                 | 2600   |
| Functions           | 27     | Composition Function 1 (N=6)                 | 2700   |
|                     | 28     | Composition Function 1 (N=6)                 | 2800   |
|                     | 29     | Composition Function 1 (N=3)                 | 2900   |
|                     | 30     | Composition Function 1 (N=3)                 | 3000   |
| Function | Item | DA | SCA | PSO | BBO | AOA | CRTHHO |
|----------|------|----|-----|-----|-----|-----|--------|
| F1       | Mean | 6.87E+07 | 7.80E+08 | 6.64E+09 | 1.99E+10 | 7.79E+09 | 4.88E+04 |
|          | Std  | 1.57E+08 | 2.97E+08 | 3.05E+09 | 9.63E−06 | 2.81E+09 | 4.61E+04 |
| F2       | Mean | -    | -    | -   | -   | -   | -      |
|          | Std  | -    | -    | -   | -   | -   | -      |
| F3       | Mean | 6.33E+03 | 1.88E+03 | 5.31E+04 | 1.21E+04 | 9.92E+03 | 3.03E+02 |
|          | Std  | 7.16E+03 | 979.94 | 2.46E+04 | 0.00 | 2601.65 | 2.26 |
| F4       | Mean | 4.42E+02 | 4.54E+02 | 9.38E+02 | 3.51E+03 | 8.02E+02 | 4.20E+02 |
|          | Std  | 56.62 | 21.61 | 342.66 | 0.00 | 256.31 | 27.48 |
| F5       | Mean | 5.64E+02 | 5.51E+02 | 6.05E+02 | 6.25E+02 | 5.62E+02 | 5.41E+02 |
|          | Std  | 18.89 | 7.12 | 19.85 | 7.76 | 20.83 | 10.76 |
| F6       | Mean | 6.34E+02 | 6.20E+02 | 6.61E+02 | 6.55E+02 | 6.41E+02 | 6.18E+02 |
|          | Std  | 16.01 | 4.24 | 15.87 | 2.60 | 7.71 | 10.63 |
| F7       | Mean | 7.48E+02 | 7.80E+02 | 9.25E+02 | 7.85E+02 | 7.98E+02 | 7.81E+02 |
|          | Std  | 18.68 | 11.16 | 61.54 | 10.29 | 12.84 | 15.37 |
| F8       | Mean | 8.45E+02 | 8.44E+02 | 8.97E+02 | 8.35E+02 | 8.31E+02 | 8.29E+02 |
|          | Std  | 18.95 | 7.26 | 23.87 | 3.79 | 6.54 | 8.39 |
| F9       | Mean | 1.23E+03 | 1.05E+03 | 2.75E+03 | 1.62E+03 | 1.34E+03 | 1.26E+03 |
|          | Std  | 394.17 | 97.60 | 1.25E+03 | 99.12 | 197.64 | 283.13 |
| F10      | Mean | 2.20E+03 | 2.33E+03 | 3.39E+03 | 2.90E+03 | 2.13E+03 | 1.83E+03 |
|          | Std  | 363.24 | 261.14 | 260.38 | 152.29 | 313.88 | 322.35 |
| F11      | Mean | 1.31E+03 | 1.22E+03 | 5.34E+03 | 1.41E+07 | 1.93E+03 | 1.13E+03 |
|          | Std  | 144.81 | 52.64 | 4056.49 | 9.36 | 1141.77 | 17.70 |
| F12      | Mean | 6.43E+06 | 1.74E+07 | 3.25E+08 | 2.69E+09 | 2.23E+07 | 1.26E+06 |
|          | Std  | 5.80E+06 | 1.57E+07 | 2.56E+08 | 92.27 | 4.95E+07 | 1.49E+06 |
| F13      | Mean | 2.03E+04 | 5.00E+04 | 7.52E+06 | 1.57E+09 | 1.25E+04 | 1.02E+04 |
|          | Std  | 1.59E+04 | 4.00E+04 | 1.07E+07 | 17.81 | 9.03E+03 | 8.76E+03 |
| F14      | Mean | 2.04E+03 | 1.70E+03 | 7.50E+04 | 1.20E+09 | 9.21E+03 | 2.22E+03 |
|          | Std  | 795.20 | 176.86 | 1.14E+05 | 2.63 | 8.49E+03 | 1.02E+03 |
| F15      | Mean | 1.86E+04 | 2.78E+03 | 4.05E+05 | 4.72E+07 | 1.62E+04 | 2.44E+03 |
|          | Std  | 1.48E+04 | 1.25E+04 | 4.54E+05 | 4.71 | 5.90E+03 | 1.82E+03 |
| F16      | Mean | 1.94E+03 | 1.76E+03 | 2.28E+03 | 2.39E+03 | 2.04E+03 | 1.91E+03 |
|          | Std  | 170.15 | 64.58 | 186.44 | 86.58 | 177.10 | 151.42 |
| F17      | Mean | 1.85E+03 | 1.78E+03 | 2.08E+03 | 1.83E+03 | 1.89E+03 | 1.76E+03 |
|          | Std  | 61.07 | 10.05 | 118.69 | 45.62 | 105.08 | 33.85 |
| F18      | Mean | 2.53E+04 | 1.94E+05 | 1.82E+07 | 8.68E+09 | 1.57E+04 | 1.74E+04 |
|          | Std  | 1.61E+04 | 1.91E+05 | 2.21E+07 | 8.50 | 9.89E+03 | 1.26E+04 |
| F19      | Mean | 2.16E+04 | 7.49E+03 | 1.40E+06 | 7.85E+09 | 3.97E+04 | 1.25E+04 |
|          | Std  | 4.29E+04 | 7.10E+03 | 2.29E+06 | 4.33 | 3.07E+04 | 1.06E+04 |
| F20      | Mean | 2.18E+03 | 2.11E+03 | 2.35E+03 | 2.42E+03 | 2.17E+03 | 2.08E+03 |
|          | Std  | 76.49 | 31.89 | 112.68 | 33.14 | 73.67 | 57.18 |
| F21      | Mean | 2.32E+03 | 2.28E+03 | 2.38E+03 | 2.57E+03 | 2.33E+03 | 2.32E+03 |
|          | Std  | 66.86 | 64.71 | 38.56 | 5.24 | 32.65 | 48.32 |
| F22      | Mean | 2.44E+03 | 2.38E+03 | 2.87E+03 | 4.15E+03 | 2.94E+03 | 2.35E+03 |
|          | Std  | 346.81 | 29.30 | 312.08 | 50.21 | 312.84 | 201.23 |
The comparison results between CRTHHO and the five algorithms is recorded in Table 10. As we can see from Table 10, the performance of CRTHHO is better than other five algorithms. Therefore, CRTHHO can effectively solve the problem of compression spring design.

4.4.2 Pressure vessel design

The goal of pressure vessel design [48] is to minimize the total cost while meeting the production needs. The four design variables are the thickness of the shell $T_s(= x_1)$, the thickness of the head $T_h(= x_2)$, the inner radius $R(= x_3)$, and the length of the

$$
\text{min } f(x) = (x_3 + 2)x_1x_3^2,
$$

Subject to:

$$
g_1(x) = 1 - \frac{x_1^2}{71785} \leq 0,
$$

$$
g_2(x) = \frac{4x_1^2 - x_1x_3}{17560x_3} + \frac{1}{5106x_3} - 1 \leq 0,
$$

$$
g_3(x) = 1 - \frac{144x_3}{x_1^2} \leq 0,
$$

$$
g_4(x) = \frac{x_3 + 4}{15} - 1 \leq 0,
$$

Variance range:

0.05 $\leq x_1 \leq 2$,

0.25 $\leq x_2 \leq 1.3$,

2 $\leq x_3 \leq 15$.

The comparison results between CRTHHO and the five algorithms is recorded in Table 10. As we can see from Table 10, the performance of CRTHHO is better than other five algorithms. Therefore, CRTHHO can effectively solve the problem of compression spring design.

### Table 7 (continued)

| Function | Item | DA | SCA | PSO | BBO | AOA | CRTHHO |
|----------|------|----|-----|-----|-----|-----|--------|
| F23      | Mean | 2.69E+03 | 2.66E+03 | 2.71E+03 | 3.20E+03 | 2.74E+03 | 2.66E+03 |
|          | Std  | 31.89  | 11.26 | 22.46 | 45.93 | 50.17 | 20.33 |
| F24      | Mean | 2.78E+03 | 2.77E+03 | 2.85E+03 | 3.17E+03 | 2.83E+03 | 2.78E+03 |
|          | Std  | 98.86  | 42.92 | 46.35 | 0.45 | 63.44 | 89.47 |
| F25      | Mean | 2.96E+03 | 2.96E+03 | 3.31E+03 | 4.01E+03 | 3.27E+03 | 2.93E+03 |
|          | Std  | 47.14  | 14.97 | 219.66 | 0.34 | 155.72 | 63.56 |
| F26      | Mean | 3.27E+03 | 3.11E+03 | 4.05E+03 | 4.93E+03 | 3.94E+03 | 3.28E+03 |
|          | Std  | 485.55 | 83.85 | 472.92 | 99.32 | 298.29 | 513.54 |
| F27      | Mean | 3.14E+03 | 3.10E+03 | 3.19E+03 | 3.79E+03 | 3.24E+03 | 3.15E+03 |
|          | Std  | 37.95  | 2.16  | 24.66 | 60.83 | 65.35 | 40.61 |
| F28      | Mean | 3.38E+03 | 3.29E+03 | 3.30E+03 | 4.04E+03 | 3.74E+03 | 3.35E+03 |
|          | Std  | 133.16 | 65.97 | 1.62  | 0.00 | 181.33 | 133.35 |
| F29      | Mean | 3.30E+03 | 3.24E+03 | 3.59E+03 | 6.44E+03 | 3.40E+03 | 3.32E+03 |
|          | Std  | 74.98  | 29.79 | 161.67 | 114.04 | 119.41 | 114.21 |
| F30      | Mean | 1.28E+06 | 1.28E+06 | 2.74E+06 | 1.96E+08 | 2.06E+07 | 6.77E+05 |
|          | Std  | 9.80E+05 | 7.04E+05 | 4.20E+06 | 0.00 | 2.15E+07 | 1.03E+06 |

© Springer
| Function | Item | DA       | SCA       | PSO       | BBO       | AOA       | CRTHHO   |
|----------|------|----------|-----------|-----------|-----------|-----------|----------|
| F1       | p    | 3.02E−11 | 3.02E−11  | 3.02E−11  | 1.41E−11  | 3.02E−11  | –        |
| F3       | p    | 3.02E−11 | 3.02E−11  | 3.02E−11  | 3.02E−11  | 3.02E−11  | –        |
| F4       | p    | 2.01E−04 | 1.64E−05  | 5.49E−11  | 2.90E−11  | 3.02E−11  | –        |
| F5       | p    | 1.61E−06 | 1.49E−04  | 3.02E−11  | 3.02E−11  | 1.87E−05  | –        |
| F6       | p    | 6.20E−04 | 1.50E−02  | 8.99E−11  | 3.02E−11  | 8.48E−09  | –        |
| F7       | p    | 6.53E−08 | 9.71E−01  | 3.02E−11  | 2.40E−01  | 4.08E−05  | –        |
| F8       | p    | 4.22E−04 | 2.02E−08  | 3.02E−11  | 1.11E−03  | 2.77E−01  | –        |
| F9       | p    | 1.67E−01 | 1.68E−03  | 5.00E−09  | 7.74E−06  | 7.98E−02  | –        |
| F10      | p    | 4.46E−04 | 5.19E−07  | 3.02E−11  | 3.02E−11  | 1.37E−03  | –        |
| F11      | p    | 1.78E−10 | 1.21E−10  | 3.02E−11  | 2.96E−11  | 4.50E−11  | –        |
| F12      | p    | 3.83E−05 | 2.61E−10  | 3.02E−11  | 2.63E−11  | 3.27E−02  | –        |
| F13      | p    | 3.18E−03 | 3.52E−07  | 3.02E−11  | 3.02E−11  | 1.62E−01  | –        |
| F14      | p    | 7.96E−01 | 1.91E−01  | 1.46E−10  | 3.02E−11  | 1.25E−04  | –        |
| F15      | p    | 1.07E−09 | 2.53E−04  | 3.02E−11  | 3.02E−11  | 2.15E−10  | –        |
| F16      | p    | 8.19E−01 | 4.35E−05  | 3.50E−09  | 3.02E−11  | 3.34E−03  | –        |
| F17      | p    | 4.18E−09 | 1.49E−06  | 3.02E−11  | 7.12E−09  | 1.31E−08  | –        |
| F18      | p    | 5.01E−02 | 2.87E−10  | 3.02E−11  | 3.02E−11  | 8.30E−01  | –        |
| F19      | p    | 2.28E−01 | 2.84E−01  | 1.09E−10  | 3.02E−11  | 8.66E−05  | –        |
| F20      | p    | 1.09E−05 | 1.56E−02  | 6.07E−11  | 3.02E−11  | 3.32E−06  | –        |
| F21      | p    | 1.67E−01 | 5.01E−01  | 1.19E−06  | 3.02E−11  | 7.96E−01  | –        |
| F22      | p    | 1.47E−07 | 7.38E−10  | 4.20E−10  | 3.02E−11  | 4.62E−10  | –        |
cylindrical part of the container \( L(= x_4) \). In addition, \( x_1 \) and \( x_2 \) are integer multiples of 0.625, \( R \) and \( L \) are continuous variables. The problem can be described as:

\[
\min f(x) = 0.6224x_1x_2x_3 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3,
\]

\[\begin{align*}
Subject \ to: \\
g_1(x) &= -x_1 + 0.0193x_3 \leq 0, \\
g_2(x) &= -x_2 + 0.00954x_3 \leq 0, \\
g_3(x) &= -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0, \\
g_4(x) &= x_4 - 240 \leq 0, 
\end{align*}\]

\[\begin{align*}
\text{Variable Range}: \\
x_1, x_2 \in \{1 \times 0.0625, 2 \times 0.0625, 3 \times 0.0625, \ldots, 1600 \times 0.0625,\}, \\
10 \leq x_3, \\
x_4 \leq 200.
\end{align*}\]

The comparison results between CRTHHO and the five algorithms is recorded in Table 11. As we can see from Table 11, the performance of CRTHHO is better than other five algorithms. Therefore, CRTHHO can effectively solve the problem of pressure vessel design.
4.4.3 Speed reducer design

In the mechanical system, the reducer is one of the important parts of the gearbox. In the speed reducer design problem (SRD) [49], the weight of the reducer is minimized under 11 constraints. The seven variables affecting the weight of the reducer are: face width \( b(=x_1) \), module of teeth \( m(=x_2) \), the number of teeth in pinion \( z(=x_3) \), the length of the first shaft between bearings \( l_1(=x_4) \), the length of the second shaft between bearings \( l_2(=x_5) \), the diameter of the first shaft \( d_1(=x_6) \), the diameter of the second shaft \( d_2(=x_7) \). The problem can be described as:

\[
\begin{align*}
\min f(X) &= 0.7854x_1x_2^2 (3.3333x_3^2 + 14.9334x_3 - 43.0934) \\
&\quad -1.508x_1(x_2^2 + x_7^2) + 7.4777(x_3^2 + x_2^2) + 0.7854(x_4x_6^2 + x_5x_7^2), \\
\text{Subject to :} \\
g_1(X) &= \frac{27}{x_1x_2x_3} - 1 \leq 0, \\
g_2(X) &= \frac{397.5}{x_1x_2x_3} - 1 \leq 0, \\
g_3(X) &= \frac{1.93x_1x_3}{x_4x_2x_3} - 1 \leq 0, \\
g_4(X) &= \frac{1.93x_1}{x_2x_3} - 1 \leq 0, \\
g_5(X) &= \sqrt{\frac{(745x_1x_2x_3)^2 + 16.9 \times 10^6}{110x_6^3}} - 1 \leq 0, \\
g_6(X) &= \sqrt{\frac{(745x_1x_2x_3)^2 + 157.5 \times 10^6}{110x_6^3}} - 1 \leq 0, \\
g_7(X) &= \frac{85x_1^2}{x_3} - 1 \leq 0, \\
g_8(X) &= \frac{5x_1}{x_2} - 1 \leq 0, \\
g_9(X) &= \frac{1.5x_1 + 1.9}{12x_5} - 1 \leq 0, \\
g_{10}(X) &= \frac{x_4}{x_5} - 1 \leq 0, \\
g_{11}(X) &= \frac{11x_4 + 1.9}{x_5} - 1 \leq 0, \\
\text{Variance Range :} \\
2.6 \leq x_1 \leq 3.6, \\
0.7 \leq x_2 \leq 0.8, \\
x_3 \in \{17, 18, 19, \ldots, 28\}, \\
7.3 \leq x_4, \\
x_5 \leq 8.3, \\
2.9 \leq x_6 \leq 3.9, \\
5 \leq x_7 \leq 5.5.
\end{align*}
\]

The experimental results of CRTHHO and the five algorithms on reducer design are shown in Table 12. By comparing the optimization algorithms, it can be seen that CRTHHO can find the lowest evaluation function value. Therefore, it has better optimization performance than other algorithms in reducer design.
Fig. 5 Convergence curves of CRTHHO and five optimization algorithms on CEC2017
Fig. 5 (continued)
Fig. 5 (continued)
The goal of the tubular column design problem (TCD) [48] is to obtain a uniform string using minimal cost so that the string can withstand compressive loads. This problem has two design variables, the mean diameter of the column \( d = x_1 \) and the thickness of tube \( t = x_2 \). The optimization model of this problem is:
Table 9 Comparison of CRTHHO and other improved HHOs

| Function Item | LMHHO [42] | GCHHO [35] | CRTHHO |
|---------------|------------|------------|--------|
| F1 Mean       | 9.98E+08   | 3.31E+03   | 4.88E+04 |
| F1 Std        | 4.07E+08   | 4.01E+03   | 4.61E+04 |
| F3 Mean       | 4.76E+03   | 6.11E+02   | 3.03E+02 |
| F3 Std        | 8.03E+02   | 2.11E+02   | 2.26E+00 |
| F4 Mean       | 5.26E+02   | 4.86E+02   | 4.20E+02 |
| F4 Std        | 4.14E+01   | 2.73E+01   | 2.75E+01 |
| F5 Mean       | 9.02E+02   | 7.16E+02   | 5.41E+02 |
| F5 Std        | 4.14E+00   | 1.08E+01   | 1.08E+01 |
| F6 Mean       | 6.15E+02   | 6.53E+02   | 6.18E+02 |
| F6 Std        | 2.81E+00   | 1.06E+01   | 1.06E+01 |
| F7 Mean       | 1.11E+03   | 1.08E+03   | 7.81E+02 |
| F7 Std        | 6.66E+01   | 7.38E+01   | 1.54E+01 |
| F8 Mean       | 1.07E+03   | 9.48E+02   | 8.29E+02 |
| F8 Std        | 6.18E+01   | 8.39E+00   | 8.39E+00 |
| F9 Mean       | 6.41E+02   | 4.79E+03   | 1.26E+03 |
| F9 Std        | 1.78E+02   | 2.83E+02   | 2.83E+02 |
| F10 Mean      | 1.37E+04   | 5.35E+03   | 1.83E+03 |
| F10 Std       | 5.20E+02   | 3.22E+02   | 3.22E+02 |
| F11 Mean      | 1.43E+04   | 1.24E+03   | 1.13E+03 |
| F11 Std       | 2.95E+03   | 1.77E+01   | 1.77E+01 |
| F12 Mean      | 3.48E+06   | 1.19E+06   | 1.26E+06 |
| F12 Std       | 3.59E+06   | 1.49E+06   | 1.49E+06 |
| F13 Mean      | 1.14E+04   | 1.29E+04   | 1.02E+04 |
| F13 Std       | 1.10E+04   | 8.76E+03   | 8.76E+03 |
| F14 Mean      | 1.94E+04   | 3.18E+04   | 2.22E+03 |
| F14 Std       | 2.09E+04   | 1.02E+03   | 1.02E+03 |
| F15 Mean      | 5.98E+04   | 5.13E+03   | 2.44E+03 |
| F15 Std       | 9.87E+04   | 1.82E+03   | 1.82E+03 |
| F16 Mean      | 2.31E+04   | 2.75E+03   | 1.91E+03 |
| F16 Std       | 1.94E+03   | 1.51E+02   | 1.51E+02 |
| F17 Mean      | 3.15E+03   | 2.27E+03   | 1.76E+03 |
| F17 Std       | 8.05E+02   | 3.39E+01   | 3.39E+01 |
| F18 Mean      | 6.00E+05   | 2.56E+05   | 1.74E+04 |
| F18 Std       | 5.96E+05   | 1.26E+04   | 1.26E+04 |
| F19 Mean      | 1.22E+05   | 8.37E+03   | 1.25E+04 |
| F19 Std       | 7.85E+04   | 1.06E+04   | 1.06E+04 |
| F20 Mean      | 1.01E+03   | 2.51E+03   | 2.08E+03 |
| F20 Std       | 3.13E+01   | 5.72E+01   | 5.72E+01 |
| F21 Mean      | 2.13E+03   | 2.48E+03   | 2.32E+03 |
| F21 Std       | 2.57E+01   | 4.83E+01   | 4.83E+01 |
| F22 Mean      | 2.07E+04   | 4.37E+03   | 2.35E+03 |
| F22 Std       | 1.08E+03   | 2.01E+02   | 2.01E+02 |
Table 9 (continued)

| Function Item | LMHHO [42] | GCHHO [35] | CRTHHO |
|---------------|------------|------------|--------|
| F23 Mean      | 3.39E+03   | 2.93E+03   | 2.66E+03 |
| F23 Std       | 2.94E+02   | 7.30E+01   | 2.03E+01 |
| F24 Mean      | 5.24E+03   | 3.07E+03   | 2.78E+03 |
| F24 Std       | 3.56E+02   | 5.24E+01   | 8.95E+01 |
| F25 Mean      | 4.29E+03   | 2.90E+03   | 2.93E+03 |
| F25 Std       | 1.95E+02   | 1.92E+01   | 6.36E+01 |
| F26 Mean      | 8.39E+03   | 5.53E+03   | 3.28E+03 |
| F26 Std       | 6.94E+02   | 1.48E+03   | 5.13E+02 |
| F27 Mean      | 5.47E+03   | 3.27E+03   | 3.15E+03 |
| F27 Std       | 5.92E+02   | 3.15E+01   | 4.05E+01 |
| F28 Mean      | 3.21E+03   | 3.21E+03   | 3.35E+03 |
| F28 Std       | 5.28E+02   | 3.00E+01   | 1.33E+02 |
| F29 Mean      | 5.72E+03   | 4.12E+03   | 3.32E+03 |
| F29 Std       | 9.86E+02   | 2.72E+02   | 1.14E+02 |
| F30 Mean      | 5.89E+05   | 1.11E+04   | 6.77E+05 |
| F30 Std       | 7.28E+04   | 4.41E+01   | 1.03E+06 |

Table 10 Comparison results of CRTHHO and other 5 algorithms on CSD problem

| Algorithm | Best   | Worst  | Mean   | Std    |
|-----------|--------|--------|--------|--------|
| PSO       | 4.10E+02 | 4.10E+02 | 4.10E+02 | 1.36E−01 |
| BBO       | 3.67E+00 | 9.60E+01 | 2.01E+01 | 2.94E+01 |
| SCA       | 3.68E+00 | 4.70E+00 | 3.93E+00 | 2.08E−01 |
| DA        | 5.79E+02 | 6.82E+02 | 6.00E+02 | 3.37E+01 |
| AOA       | 5.92E+00 | 1.19E+01 | 7.32E+00 | 1.44E+00 |
| CRTHHO    | 3.66E+00 | 3.74E+00 | 3.70E+00 | 3.15E−02 |

Table 11 Comparison results of CRTHHO and other 5 algorithms on PVD problem

| Algorithm | Best | Worst | Mean | Std   |
|-----------|------|-------|------|-------|
| PSO       | 2.04E+05 | 2.08E+05 | 2.05E+05 | 1.04E+03 |
| BBO       | 9.64E+03 | 3.52E+05 | 9.22E+04 | 7.73E+04 |
| SCA       | 2.34E+03 | 6.29E+03 | 5.28E+03 | 1.54E+03 |
| DA        | 6.67E+05 | 1.90E+06 | 8.53E+05 | 3.61E+05 |
| AOA       | 4.21E+03 | 1.49E+05 | 2.78E+04 | 3.89E+04 |
| CRTHHO    | 2.03E+03 | 3.67E+03 | 2.77E+03 | 6.57E+02 |
The CRTHHO algorithm and five other algorithms are used to solve this problem, and the test results are shown in Table 13. By comparison, it can be seen that the best value, worst value, and average value of CRTHHO algorithm are the lowest. Therefore, compared with the other five algorithms, CRTHHO algorithm has better optimization effect.

\[
\min f(X) = 9.8x_1x_2 + 2x_1,
\]

subject to:

\[
g_1(X) = \frac{p}{x_1x_2 \sigma_f} - 1 \leq 0,
\]

\[
g_2(X) = \frac{8pL^2}{\pi^2 E_1 x_2 \left( x_1^2 + x_2^2 \right)} - 1 \leq 0,
\]

\[
g_3(X) = \frac{2.0}{x_1} - 1 \leq 0,
\]

\[
g_4(X) = \frac{x_1}{14} - 1 \leq 0,
\]

\[
g_5(X) = \frac{0.2}{x_2} - 1 \leq 0,
\]

\[
g_6(X) = \frac{x_2}{8} - 1 \leq 0,
\]

\text{Variance Range:}

\[
2 \leq x_1 \leq 14,
\]

\[
0.2 \leq x_2 \leq 0.8.
\]
5 Conclusion and prospect

This paper presents a Harris hawks optimization based on global cross-variation and tent mapping. Firstly, the tent map is introduced into the exploration stage of the algorithm, and the parameter is adjusted by using its ergodicity and regularity, which makes up for the slow convergence speed of the algorithm in the early stage and improves the convergence speed of the HHO algorithm. Secondly, the crossover mutation strategy is introduced to mutate the current global optimal solution and select the optimal individual by the greedy selection, which can avoid missing the potential optimal solution in a certain probability and improve the convergence accuracy. The improved algorithm proposed in this paper combines the above two strategies. The performance of CRTHHO is tested by three different experiments in Sect. 4. Comparing the test results of three different dimensions in the 10 benchmark functions, it can be found that the CRTHHO algorithm has the best test results for the 9 functions, and with the increase of dimension, the advantage of CRTHHO algorithm become more outstanding. Comparing the test results of 30 test functions in CEC2017, it can be concluded that the results of CRTHHO are best in 17 functions and suboptimal in 5 functions, while it wins the first place in overall score. Finally, the performance of CRTHHO is further tested by four engineering problems, and the optimal test results are obtained. The above three experiments show that CRTHHO algorithm has better optimization performance than the basic HHO algorithm and five novel meta-heuristic algorithms. The CRTHHO algorithm still has a lot of development space in the future, and there are many aspects to study. For example, whether CRTHHO can introduce new strategies to further improve the algorithm’s performance, whether the proposed improved strategy can be applied to the multi-objective optimization algorithm, and whether the introduction has a good effect. It is also worth thinking about what practical problems this proposed algorithm can study.

Acknowledgements The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

Author contributions CL was responsible for methodology, writing and reviewing, and supervision. SN was involved in data collation and tabulation, writing, reviewing, and editing, revising the manuscript and software. MY took part in writing, reviewing, and editing; and plotting the figures.

Funding This study was supported by the National Natural Science Foundation of China (No. 61535008), the Natural Science Foundation of Tianjin (No. 20JCQNJC00430), the National Natural Science Foundation of China (No. 62203332) and the Science and Technology Research Team in Higher Education Institutions of Hebei Province (No. ZD2018045).

Data availability Not applicable.

Declarations

Conflict of interest We declare that the authors have no competing interests as defined by Springer, or other interests that might be perceived to influence the results and/or discussion reported in this paper.

Consent for publication Not applicable.
Ethics approval and Consent to participate  Informed consent was obtained from all individual participants included in the study.

Human and animal rights  Not applicable.

References

1. Shabania A, Asgariana B, Salidob M, Gherebaghi SA (2020) Search and rescue optimization algorithm: a new optimization method for solving constrained engineering optimization problems. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2020.113698
2. Chen YL, He FZ, Li HR, Zhang DJ, Wu YQ (2020) A full migration bbo algorithm with enhanced population quality bounds for multimodal biomedical image registration. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2020.106335
3. Fateh H, Baharamara S, Safari A (2020) Modeling operation problem of active distribution networks with retailers and microgrids: a multi-objective bi-level approach. Appl Soft Comput 94:106484. https://doi.org/10.1016/j.asoc.2020.106484
4. Sundaramurthy S, Jayavel P (2020) A hybrid grey wolf optimization and particle swarm optimization with c4.5 approach for prediction of rheumatoid arthritis. Appl Soft Comput 94:106500. https://doi.org/10.1016/j.asoc.2020.106500
5. Almohammadi KM, Harby K (2020) Operational conditions optimization of a proposed solar-powered adsorption cooling system: experimental, modeling, and optimization algorithm techniques. Energy 206:118007. https://doi.org/10.1016/j.energy.2020.118007
6. Moazzeni AR, Khamehchi E (2020) Rain optimization algorithm (roa): a new metaheuristic method for drilling optimization solutions. J Petrol Sci Eng 195:107512. https://doi.org/10.1016/j.petrol.2020.107512
7. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw. https://doi.org/10.1016/j.advengsoft.2013.12.007
8. Holland JH (1992) Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. https://doi.org/10.7551/mitpress/1090.001.0001
9. Storn R, Price K (1997) Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 11(4):341–359. https://doi.org/10.1023/A:1008202821328
10. Mirjalili S, Mirjalili SM, Hatamlou A (2015) Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Comput Appl 27(2):495–513. https://doi.org/10.1007/s00521-015-1870-7
11. Zhao WG, Wang LY, Zhang ZX (2018) A novel atom search optimization for dispersion coefficient estimation in groundwater. Future Gener Comput Syst 91(FEB):601–610
12. Eberhart R, Kennedy J (1995) A new optimizer using particle swarm optimization. In: Mhs95 Sixth International Symposium on Micro Machine & Human Science, pp 39–43. https://doi.org/10.1109/MHS.1995.494215
13. Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. J Global Optim 39(3):459–471. https://doi.org/10.1007/s10898-007-9149-x
14. Simon D (2009) Biogeography-based optimization. IEEE Trans Evol Comput 12(6):702–713. https://doi.org/10.1109/TEVC.2008.919004
15. Mirjalili S (2016) Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Comput Appl 27(4):1053–1073. https://doi.org/10.1007/s00521-015-1920-1
16. Mirjalili S (2016) Sca: a sine cosine algorithm for solving optimization problems. Knowl Based Syst. https://doi.org/10.1016/j.knosys.2015.12.022
17. Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl-Based Syst 89:228–249. https://doi.org/10.1016/j.knosys.2015.07.006
18. Abualigah L, Diabat A, Mirjalili S, Elaziz MA, Gandomi AH (2021) The arithmetic optimization algorithm. Comput Methods Appl Mech Eng 376:113609. https://doi.org/10.1016/j.cma.2020.113609

19. Ahmadianfar I, Heidari AA, Noshadian S, Chen H, Gandomi AH (2022) Info: An efficient optimization algorithm based on weighted mean of vectors. Expert Syst Appl 195:116516

20. Martinez-Alvarez F, Cortés GA, José TF, Gutierrez-Aviles D, Melgar-García L, Pérez-Chacón R, Rubio-Escudero C, Rubio-Escudero C, Troncoso A (2020) Coronavirus optimization algorithm: a biospired metaheuristic based on the covid-19 propagation model. Big Data 8:308–322

21. Abd El-Sattar H, Kamel S, Hassan MH, Jurado F (2022) Optimal sizing of an off-grid hybrid photovoltaic/biomass gasifier/battery system using a quantum model of runge kutta algorithm. Energy Convers Manag 258:115539

22. Weiguo Z, Liying W, Seyedali M (2022) Artificial hummingbird algorithm: a new bio-inspired optimizer with its engineering applications. Comput Methods Appl Mech Eng 388:114194

23. Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen HL (2019) Harris hawks optimization: algorithm and applications. Futur Gener Comput Syst 97:849–872. https://doi.org/10.1016/j.future.2019.02.028

24. Pasandideh SHR, Niaki STA, Gharaei A (2015) Optimization of a multiproduct economic production quantity problem with stochastic constraints using sequential quadratic programming. Knowl Based Syst 84(8):98–107. https://doi.org/10.1016/j.knosys.2015.04.001

25. Abbasi A, Firoozu B, Sendur P (2021) On the application of harris hawks optimization (HHO) algorithm to the design of microchannel heat sinks. Eng Comput 37:1409–1428. https://doi.org/10.1007/s00366-019-00892-0

26. Konijeti MSNK, Bharathi ML (2022) Extraction of maximum power from solar with BLDC motor driven electric vehicles based HHO algorithm. Adv Eng Softw 170:103137. https://doi.org/10.1016/j.advengsoft.2022.103137

27. Mashaleh AS, Ibrahim NFB, Al-Betar MA, Mustafa HMI, Yaseen QM (2022) Detecting spam email with machine learning optimized with harris hawks optimizer (hho) algorithm. Procedia Comput Sci 201:659–664. https://doi.org/10.1016/j.procs.2022.03.087

28. Wunnava A, Naik MK, Panda R, Jena B, Abraham A (2020) A differential evolutionary adaptive harris hawks optimization for two dimensional practical masi entropy-based multilevel image thresholding. J King Saud Univ Comput Inf Sci. https://doi.org/10.1016/j.jksuci.2020.05.001

29. Houssein EH, Hosney ME, Oliv D, Mohamed WM, Hassaballah M (2020) A novel hybrid harris hawks optimization and support vector machines for drug design and discovery. Comput Chem Eng 133(Feb.2):106656–110666516. https://doi.org/10.1016/j.compchemeng.2019.106656

30. Essa FA, Elaziz MA, Elsheikh AH (2020) An enhanced productivity prediction model of active solar still using artificial neural network and harris hawks optimizer. Appl Thermal Eng. https://doi.org/10.1016/j.applthermaleng.2020.115020

31. Chen HL, Jiao S, Wang MJ, Heidari AA, Zhao XH (2019) Parameters identification of photovoltaic cells and modules using diversification-enriched harris hawks optimization with chaotic drifts. J Clean Prod. https://doi.org/10.1016/j.jclepro.2019.118778

32. KumarKamboj V, Nandi A, Bhadoria A, Sehgal S (2020) An intensify harris hawks optimizer for numerical and engineering optimization problems - sciencedirect. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2019.106018

33. Qu CW, He W, Peng XN, Peng XN (2020) Harris hawks optimization with information exchange. Appl Math Model. https://doi.org/10.1016/j.apm.2020.03.024

34. Khalifeh S, Akbarifard S, Khalifeh V, Zallaghi E (2020) Optimization of water distribution of network systems using the harris hawks optimization algorithm (case study: Homashahr city). MethodsX. https://doi.org/10.1016/j.mex.2020.100948

35. Tikhamarine Y, Souag-Gamane D, Ahmed AN, Sammen SS, Kisi O, Huang YF, El-Shafie A (2020) Rainfall-runoff modelling using improved machine learning methods: Harris hawks optimizer vs particle swarm optimization. J Hydrol 589:125133. https://doi.org/10.1016/j.jhydrol.2020.125133

36. Song S, Wang P, Heidari AA, Wang M, Zhao X, Chen H, He W, Xu S (2021) Dimension decided harris hawks optimization with gaussian mutation: balance analysis and diversity patterns. Knowl Based Syst 215(Mar.5):106425–110642526

37. Li C, Li J, Chen H, Jin M, Ren H (2021) Enhanced harris hawks optimization with multi-strategy for global optimization tasks. Expert Syst Appl 185:115499. https://doi.org/10.1016/j.eswa.2021.115499
38. Zhao SJ, Gao LF, Yu DM, Tu J (2021) Improved harris hawks optimization coupling energy cycle decline mechanism and newton local enhancement strategy. Control Decis 36(3):629–636. https://doi.org/10.13195/j.kzyjc.2019.0810
39. Guo YX, Liu S, Gao WX, Zhang L (2022) Elite opposition-based learning golden-sine harris hawks optimization. Comput Eng Appl 58(10):153–161
40. Liu XL, Liang TY (2022) Harris hawk optimization algorithm based on square neighborhood and random array. Control Decis. https://doi.org/10.13195/j.kzyjc.2021.0478
41. Fan Q, Chen ZJ, Xia ZH (2020) A novel quasi-reflected harris hawks optimization algorithm for global optimization problems. Soft Comput 24:14825–14843. https://doi.org/10.1007/s00500-020-04834-7
42. Hussain K, Zhu W, Salleh MNM (2019) Long-term memory harris’ hawk optimization for high dimensional and optimal power flow problems. IEEE Access. https://doi.org/10.1109/ACCESS.2019.2946664
43. Ma YM, Shi ZD, Zhao K, Gong CL, Shan LH (2020) Tdoa localization based on improved harris hawk optimization algorithm. Comput Eng Appl 46(12):179–184. https://doi.org/10.19678/j.issn.1000-3428.0056965
44. Zawbaa HM, Emary E, Grosan C (2016) Feature selection via chaotic antlion optimization. PLoS ONE 11(3):0150652
45. Gupta S, Deep K (2019) Improved sine cosine algorithm with crossover scheme for global optimization. Knowle Based Syst 165:374–406. https://doi.org/10.1016/j.knosys.2018.12.008
46. Wu G, Mallipeddi R, Suganthan PN (2016) Problem definitions and evaluation criteria for the cec 2017 competition on constrained real-parameter optimization. Technical Report, Nanyang Technological University, Singapore, September 2017
47. Arora J (2012) Introduction to optimum design. Elsevier, Amsterdam Netherlands
48. Bayzidi H, Talatahari S, Saraee M, Lamarche C-P (2021) Social network search for solving engineering optimization problems. Comput Intell Neurosci 2021:8548639. https://doi.org/10.1155/2021/8548639
49. Sattar D, Salim R (2020) A smart metaheuristic algorithm for solving engineering problems. Engineering with Computers 37:2389–2417

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.