Available online at www.sciencedirect.com
ScienceDirect
Mathematics and Computers in Simulation 204 (2023) 619–639
MATHEMATICS AND COMPUTERS IN SIMULATION
www.elsevier.com/locate/matcom

Original articles

Improved Harris hawks optimization for non-convex function optimization and design optimization problems

Helei Kang[a], Renyun Liu[a,*], Yifei Yao[b], Fanhua Yu[c]

[a] *Department of Mathematics, Changchun Normal University, Jilin 130032, China*
[b] *Department of Computer Science, Changchun Normal University, Jilin 130032, China*
[c] *Department of Computer Science, Beihua University, Jilin 132013, China*

Received 8 January 2022; received in revised form 27 July 2022; accepted 14 September 2022
Available online 21 September 2022

Abstract

Harris hawks optimization (HHO) is a nature-inspired algorithm. It has the advantages of very few parameters, a simple structure, fast convergence and strong local search capability. The main drawback of the Harris hawks optimization is that it can easily fall into a local optimum. To solve this problem, a novel mutant strategy based on Brownian motion is proposed to combine with the original HHO. This mutant strategy is driven by exploiting the randomness of Brownian motion and does not require location information between populations and user-set parameters. As a result, it can guide the algorithm to better jump out of the local optimum trap. To verify the performance of the proposed algorithm, numerical experiments are carried out to compare the proposed algorithm with heuristic optimization algorithms for 54 non-convex functions and two classic engineering design problems. The results show that our algorithm not only escapes the local optimum trap, but also has better robustness and convergence.

Keywords: Harris hawks optimization; Brownian motion; Intelligent algorithms; Mutant strategy; Engineering design

1. Introduction

Optimization is an essential branch of mathematics, and optimization algorithms are widely used in engineering, economics and many other branches of science. Optimization problems can be divided into convex optimization and non-convex optimization. There are already many effective optimization methods to deal with convex optimization problems [10]. However, non-convex optimization problems are inherently nonlinear and multimodal and widely seen in machine learning and engineering design problems [7,58]. They tend to have many local optimal solutions, which makes it very difficult to find the global optimum for a non-convex function [37,44]. Traditional optimization algorithms generally generate a series of point sequences to converge to an optimal solution based on the analytical nature of the problem. However, these optimization algorithms are dependent on the initial value and gradient information, which makes them easy to fall into the trap of local minima and thus fail to find the global optimum. Meta-heuristic algorithms are a class of methods that operate by repeated evaluation of objective functions. They

* Corresponding author.
E-mail address: liurenyun@ccsfu.edu.cn (R. Liu).

https://doi.org/10.1016/j.matcom.2022.09.010

do not require initial values and gradient information [18]. As a result, they have unique advantages in solving non-convex optimization problems. Since many non-convex optimization problems are discontinuous, non-differentiable and complex, meta-heuristic algorithms have a great advantage over traditional algorithms in solving non-convex optimization problems [17,47,50].

Many meta-heuristic algorithms have been developed, such as particle swarm optimization (PSO) [28], whale optimization algorithm (WOA) [34], bat-inspired algorithm (BAT) [54], firefly Algorithm (FA) [53], monarch butterfly optimization (MBO) [49], slime mould algorithm (SMA) [30], moth search algorithm (MSA) [48], hunger games search (HGS) [55], colony predation algorithm (CPA) [46], and Runge Kutta method (RUN) [5]. Particle swarm optimization is a population-based heuristic algorithm designed by mimicking the predatory behavior of birds. Whale optimization algorithm is a nature-inspired algorithm that mimics attack behavior of humpback whales with bubble nets. The idea of Bat-inspired algorithm originates from the simulation of advanced echolocation ability of bats, and a novel intelligent algorithm is obtained by modeling the echolocation behavior of bats and applying it to function optimization, feature extraction and other applications. Firefly algorithm is an optimization algorithm that simulates firefly luminous intensity, which uses particle swarm optimization to solve engineering design problems. The monarch butterfly optimization mainly simulates the migration and environmental adaptation behavior of monarch butterflies, and has received much attention in recent years because of its simple structure and superior experimental results. The slime mould algorithm is a metaheuristic algorithm inspired by the diffusion and foraging behavior of slime mould. The moth search algorithm is a bio-inspired metaheuristic algorithm which is built using the phototropism of moths and Lévy flight. The hunger games search algorithm is a novel intelligent optimization algorithm that is based on animal hunger-driven activities and behaviors. The colony predation algorithm is a stochastic optimizer which mimics the supportive behavior and selective hunting of social animals. The Runge Kutta method is inspired by the numerical method in mathematics, which avoids metaphor, converges faster and can avoid the defects of local optimum compared with the above intelligent algorithm.

Harris hawks optimization (HHO) is a recently proposed population-based intelligence algorithm inspired by simulating Harris hawks predation. Due to its simple structure, fast convergence, and freedom from user input parameters, HHO is widely popular among researchers. However, HHO can easily fall into a local optimum. Some improvements have been made by researchers to address this issue. Abdel-Basset et al. [1] proposed a novel hybrid Harris Hawks optimization algorithm which combines simulated annealing and Bitwise operations with HHO for feature selection. It was shown that the performance of the proposed algorithm is better than many other optimization algorithms. I. Gölcük et al. [20] proposed a multi-group HHO algorithm using dynamic optimization. Experimental results show that the proposed algorithm significantly outperforms other algorithms. ChenYangLi et al. [31] apply the Rosenbrock method to Harris Hawks optimization to obtain a new algorithm (RLHHO) and use it to solve the global optimization problems. Experimental results show that the modified algorithm has better convergence. Chencheng Fan et al. [16] proposed a neighborhood centroid opposite-based learning Harris Hawks optimization algorithm (NCOHHO) for training neural networks. It was shown that the performance of the proposed algorithm is significantly better than the other metaheuristic algorithms. S.J. Badashah et al. [9] present a kind of Fractional-Harris hawks optimization and apply it to optimize the adversarial network. The results show that the modified algorithm has a better effect. O. Akdag et al. [6] proposed to use the distribution function to correct the HHO algorithm. The performance of the corrected algorithm is better than the original algorithm through simulation experiments. E.H. Houssein et al. [23] combined the HHO algorithm with genetic operators. Experiments show that the proposed algorithm performs well in jumping out of the local optimum. Ya Su et al. [42] present a parallel Harris hawks optimization (HPHHO) for dealing with the reentry trajectory optimization problem. Simulation tests show that the algorithm has good robustness. Wunnava et al. [51] proposed an adaptive Harris hawks optimization technique (AHHO) for finding the optimal threshold values. The proposed algorithm is experimentally demonstrated to have better convergence speed and robustness. S. Jiao et al. [26] propose a new algorithm (CSHHO) in which Gaussian chaos mapping methods and optimal neighborhood perturbation mechanisms are introduced to improve accuracy and avoid premature maturation of the algorithm, while an adaptive weighting and variable spiral search strategy is designed to simulate the prey hunting behavior of Harris's hawk for faster convergence. Numerical experiments show that the improved algorithm outperforms other intelligent optimization algorithms. The CSHHO algorithm has been applied to the reactive power output of the analog synchronous regulator, and the effect is very noticeable. Too et al. [21] proposed a memory-based Harris hawk optimization method with learning agents (MEHHO). To improve algorithm exploitation and the ability to jump out of the local optimum, the MEHHO

approaches adopt an energetic learning strategy and memory saving and updating mechanism. MEHHO was used to solve the feature selection problem, and the experimental results show that MEHHO is an effective feature selection approach. Chen et al. [11] proposed a new algorithm called CMDHHO, which combines HHO with topological multi-population strategy, chaos strategy and DE strategy. To verify the performance of the improved algorithm, the proposed CMDHHO is compared with a range of algorithms, based on IEEE CEC2017 benchmark functions, and IEEE CEC2011 real-world problems. Experimental results show that the improved algorithm can effectively enhance the exploratory and exploitative of HHO, and the convergence speed is greatly improved. S. Song et al. [40] proposed to new persistent trigonometric differences and Adaptive mechanism and combined with the original HHO algorithm (ADHHO) for photovoltaic model parameter extraction. Experimental results show that the proposed algorithm has faster convergence speed and accuracy.

Mutant strategy can help an algorithm avoid falling into a local optimum. There are already many mutant strategies such as DE [41], Gaussian mutant [32] and uniform mutation [24]. These mutant strategies are effective in jumping out of a local optimum. However, these mutant strategies require user to input parameters, which negatively affects the efficiency of the mutant strategies. In addition, these mutant strategies rely on the interaction between populations, which may make these mutated strategies much less effective in the final stage of the algorithm. To overcome these problems, we propose a Brownian Motion-based mutant strategy. This strategy exploits the stochastic nature of Brownian motion, which in turn helps the algorithm to jump out of a local optimum. In this paper, in order to solve the local optimum trap problem when optimizing non-convex functions, HHO is combined with the Brownian motion-based mutant strategy to generate a new optimization algorithm named HHOBM algorithm, which can avoid falling into the local optimum trap and help to find the global optimum solution. 44 non-convex function [8], 10 modern CEC test functions [35] and two engineering design problems are used to verify the performance of the improved algorithm. In these experiments, we use the Wilcoxon signed-rank test to determine if the improvement of the proposed algorithm over other algorithms is significant [13]. We also use the Friedman average rank test to rank the algorithms [19]. Our experimental results indicate that HHOBM not only has better global search capability, but also has better convergence and stability than other optimization algorithms.

The rest of the paper is organized as follows. Section 2 describes the basic HHO algorithm in detail. Section 3 contains the proposed algorithm-Brownian Motion-based mutant strategy and HHOBM. Section 4 presents the experimental portion of this paper. Finally, conclusions and future work are given in Section 5.

2. Harris hawks optimization (HHO) algorithm

The Harris hawks optimization (HHO) algorithm was proposed by Heidari et al. [21] in 2019. HHO is based on predation process of Harris hawks. As a novel population-based and nature-inspired optimization method, HHO has two phases: the exploration phase and exploitation phase. The structure of the HHO algorithm is shown in Fig. 1. These two phases are controlled by energy E_{prey} for conversion. When $|E_{prey}| \geq 1$ it indicates that the HHO algorithm is in the exploration phase. On the other hand, when the value of escaping energy $|E_{prey}| < 1$, the algorithm is in the phase of exploitation. Here E_{prey} is calculated as follows:

$$E_{prey} = 2E_{init} \times (1 - \frac{t}{t_{max}}) \tag{1}$$

where t and t_{max} are the number of current iterations and total number of iterations, respectively, E_{init} is initial energy, and E_{init} is a random number between $(-1, 1)$.

2.1. Exploration phase

In exploration phase, there are two methods of updating the position of the Harris hawks: one describes Harris hawks perch based on the positions of other family members and the prey, and the other one describes the habitation of random trees by Harris hawks. For both methods, the selection is made by a randomly chosen number q from a uniform distribution in $(0, 1)$. So the position of the hawk is updated as follows [21]:

$$X(t+1) = \begin{cases} X_{\mathrm{r}}(t) - c_1 \left| X_{\mathrm{r}}(t) - 2 \times c_2 X(t) \right| & q \geq 0.5 \\ \left(X_{\text{rabbit}}\ (t) - X_{mean}(t)\right) - c_3 \left(LB + c_4(UB - LB)\right) & q < 0.5 \end{cases} \tag{2}$$

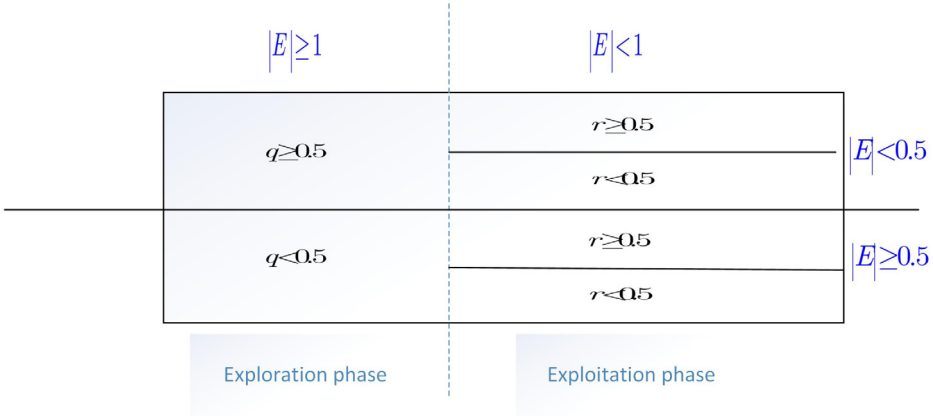

Fig. 1. The structure of HHO algorithm.

where $X(t+1)$ is the position of the hawks in the next iteration, $X(t)$, $X_{mean}(t)$, $X_r(t)$ and $X_{rabbit}(t)$ are hawks current position, the average location of the current population, the position of randomly selected hawk and the position of rabbit, respectively. Here t is the current iteration, c_1, c_2, c_3, c_4 are random numbers inside (0, 1), and LB and UB are the upper and lower boundaries of search space, respectively. The average position of the population is calculated by:

$$X_{mean}(t) = \frac{1}{N}\sum_{i=1}^{N} X_i(t) \tag{3}$$

where N is the number of population size, and $X_i(t)$ is the location of each population in current iteration.

2.2. Exploitation phase

In this phase, hawks have four different trapping strategies. The choice of these strategies relies on the escaping energy of prey, and the probability of prey in successfully escaping or not successfully escaping. Assume that r is a sample drawn from a uniformly distributed random variable from 0 to 1. If $r < 0.5$ then the escape was successful. Otherwise, the result is failure of escape. All strategies are described in the following.

2.2.1. Soft besiege

The soft besiege happens when $|E_{prey}| \geq 0.5$ and $r \geq 0.5$. In this situation, a rabbit has enough energy, and the rabbit has a high escaping probability. The position of the hawk is updated as follows [21]:

$$X(t+1) = X_{rabbit}(t) - X(t) - E_{prey} \mid JX_{\text{rabbit}}(t) - X(t) \mid \tag{4}$$

where X is the position of the hawk, E_{prey} is the energy of prey, t is the current iteration, and J is the jump strength, which is calculated as follows [21]:

$$J = 2(1 - c_5) \tag{5}$$

where c_5 is a random number in (0, 1) that is updated randomly in each iteration.

2.2.2. Hard besiege

The hard besiege happens when $|E_{prey}| < 0.5$ and $r \geq 0.5$. In this situation, the rabbit is very exhausted with very has low escaping energy. Nevertheless, the rabbit can still successfully evade pursuit. The position of the hawk is updated as follows [21]:

$$X(t+1) = X_{rabbit}(t) - E_{prey} \mid X_{rabbit}(t) - X(t) \mid . \tag{6}$$

2.2.3. Soft besiege with progressive rapid dives

The soft besiege with progressive rapid dives happens when $|E_{prey}| \geq 0.5$ and $r < 0.5$. In this situation, the rabbit has enough energy to escape from an attack, and the hawks still construct a soft besiege. The position of the hawks is updated as follows [21]:

$$Y = X_{rabbit}(t) - E_{prey} \mid JX_{rabbit}(t) - X(t) \mid \tag{7}$$

Then, hawks will compare the possible result of this action with previous dives. If the results are not good, they will start to perform irregular, abrupt, and rapid dives. The position of the hawks is updated as follows [21]:

$$Z = Y + S \times Levy(dim) \tag{8}$$

where dim is the dimension of the optimization problem, S is a random vector of size $1 \times dim$, and $Levy$ is the levy flight function that can be calculated as follows [21]:

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

where u and v are random values inside (0, 1), β is a default constant set to be 1.5, and σ is calculated as follows:

$$\sigma = \left(\frac{\Gamma(1+\beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right)^{\frac{1}{\beta}}. \tag{10}$$

In this situation, the position of the hawk is updated as follows [21]:

$$X(t+1) = \begin{cases} Y & \text{if } F(Y) < F(X(t)), \\ Z & \text{if } F(Z) < F(X(t)). \end{cases} \tag{11}$$

where Y and Z are calculated by Eqs. (7) and (8), and F is the fitness function.

2.2.4. Hard besiege with progressive rapid dives

The hard besiege with progressive rapid dives happens when $|E_{prey}| < 0.5$ and $r < 0.5$. In this situation, the rabbit has too low energy to escape from attack, and the hawks perform a hard besiege at the same time. In this situation, two new solutions are calculated as follows [21]:

$$Y = X_{rabbit}(t) - E_{prey} \mid JX_{rabbit}(t) - X_{mean}(t) \mid \tag{12}$$

$$Z = Y + S \times Levy(dim) \tag{13}$$

where $X_{mean}(t)$ is the average location of the current population of hawks. In this situation, the position of the hawk is updated as follows [21]:

$$X(t+1) = \begin{cases} Y & \text{if } F(Y) < F(X(t)), \\ Z & \text{if } F(Z) < F(X(t)). \end{cases} \tag{14}$$

3. The proposed method

3.1. Brownian motion-based mutant strategy

Brownian motion is the never-ending irregular motion made by particles suspended in a liquid or gas. The motion was found by Robert Brown in 1828 [27]. Brownian motion is a type of stochastic process. It can be proved that Brownian motion is a Markov process, harness process and Ito process [25]. The one-dimensional Brownian motion (see Fig. 2) trend in the time domain looks very similar to that of the stock price curve, which has led to the interest in using Brownian motion to describe the stock price trend. Stochastic differential equations (SDEs) driven by Brownian motion have also had great success in modeling and predicting financial directions and price

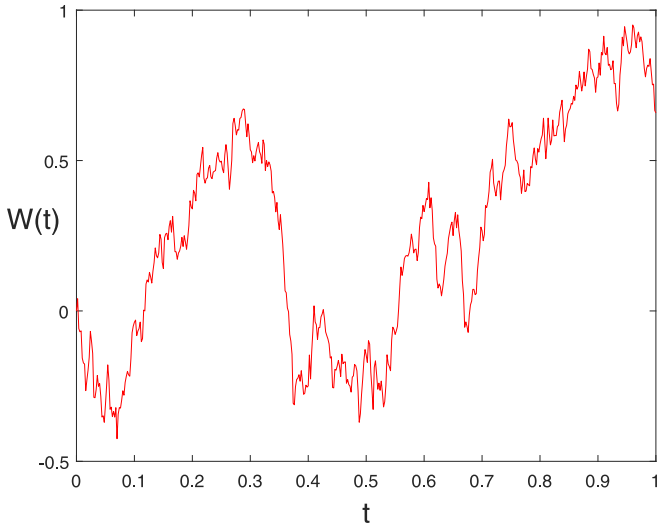

Fig. 2. Brownian motion in time domain.

estimations [29]. Mathematical descriptions of Brownian motion have developed more slowly. A rigorous definition and description of Brownian motion was proposed by Norbert Wiener in 1938. A scalar standard Brownian motion is a random variable $W(t)$ that depends continuously on $t \in [0, T]$ and satisfies the following three conditions [22]:

- $W(0) = 0$ or $W(0) = 1$.
- For $x, y \in [0, T], 0 \leq x < y \leq T$, the increment $W(y) - W(x) \sim \sqrt{y - x}N(0, 1)$, where $N(0, 1)$ is a standard normally distributed random variable.
- For $x, y, a, b \in [0, T], 0 \leq x < y < a < b \leq T$, the increment $W(y) - W(x)$ and $W(a) - W(b)$ are independent.

In general, it is useful to discretize the Brownian motion, define $\delta_t = T/N$ for some positive integer N and W_i is updated by the following equation [22]:

$$W_{i+1} = W_i + dW_i \tag{15}$$

where W_i is $W(t_i)$ with $t_i = i\delta_t$. dW_i is an independent random variable of the form $\sqrt{\delta_t}N(0, 1)$, Fig. 4 shows the path of a single particle in Brownian motion.

Similar to the above formulation, we propose a new mutant strategy as follows. For each particle X, a mutant particle X_{mutant} is generated through the following formula:

$$X_{mutant} = X + (-1)^r * dW_t \tag{16}$$

where r is an integer drawn from a uniformly distributed random variable from UB to LB and dW_t is independent random variable of the form $\sqrt{(Ub - Lb)/N}N(0, 1)$. Note that Lb and Ub are the upper and lower boundaries of search space, respectively, and N is the number of population size. Also note that the effect of $(-1)^r$ is to make the increments more random and the fluctuations more frequent (see Fig. 3). The position of the particles is updated as follows:

$$X(t + 1) = \begin{cases} X_{mutant}, iff\ (X_{mutant}) < f\ (X(t)), \\ X(t), \text{ else }. \end{cases} \tag{17}$$

Compared with other mutant strategies, Brownian motion-based mutant strategy does not input many parameters or increase the computation of the algorithm. Compared with DE, the Brownian motion-based mutant strategy is not affected by parameters. Also, it avoids user input parameters, thus simplifying the structure of the algorithm. In addition, Brownian motion based mutant strategy does not rely on the interaction between populations, which makes the new populations generated by the mutation more widely distributed, thus achieving a wider local search

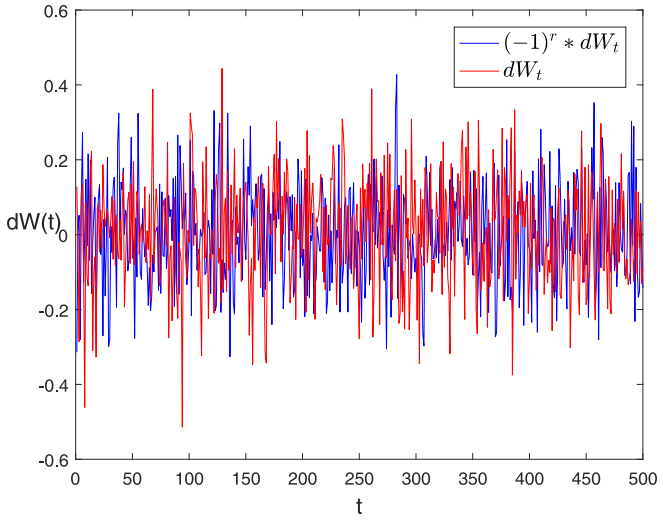

Fig. 3. Compare dW_t with $(-1)^r * dW_t$.

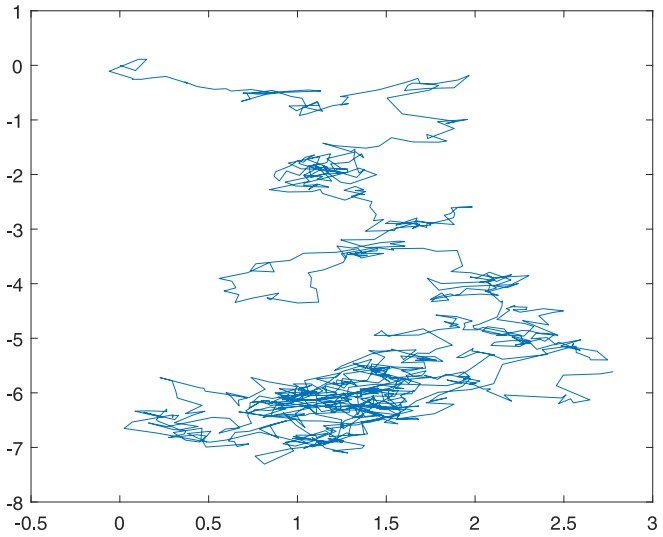

Fig. 4. The path of a single particle in Brownian motion.

area and more easily guiding the algorithm to jump out of the local optimum. Fig. 5 indicates that the population falls into a local optimum after a certain iteration, and four mutant strategies are used to simulate the iteration to jump out of local optimum. In the DE mutant strategy, the effect of mutant factor of 0.8 is much better than that of 0.4, and the effect of Brownian motion based mutant performs better than that of DE mutant. For DE mutant, considering the interaction between populations, the effect of any is not noticeable. On the other hand, the Brownian motion based mutant does not need to consider the influence of parameters and is not affected by the interaction between populations, thus the ability to jump out of the local optimum is higher. The pseudo-code of the Brownian Motion-based mutant strategy can be represented by Algorithm 1.

3.2. Harris hawks optimization based on Brownian motion mutant strategy (HHOBM)

HHO is a simple and effective algorithm that is popular among researchers because of its effectiveness in solving engineering problems and other problems, but there is still room for improvement when dealing with

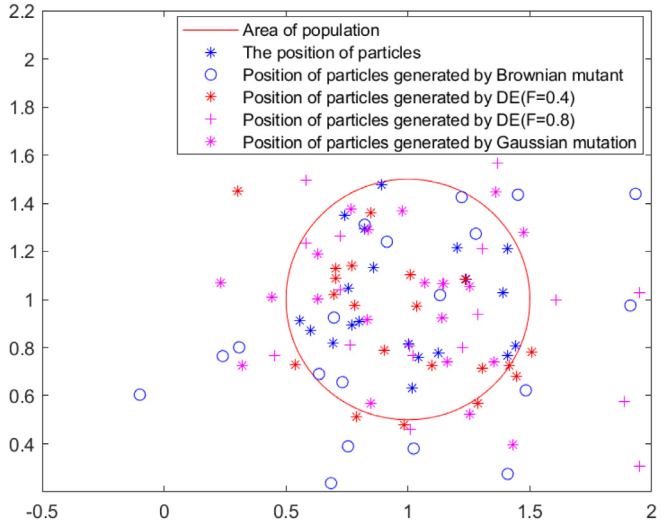

Fig. 5. Comparison of populations produced by various mutation strategies.

Algorithm 1 Brownian Motion-based mutant strategy

1: Generating mutant particles X_{mutant} using Eq.(16)
2: if($f(X_{mutant}) < f(X(t))$) then
3: Use X_{mutant} instead of $X(t+1)$
4: Else Use $X(t)$ instead of $X(t+1)$

multimodal problems. The main problem of HHO is that it may not be able to maintain a balance between exploration and exploitation and it can fall into a local optimum trap. On the other hand, non-convex optimization usually admits a multimodal structure, and traditional HHO may trap in poor local optima. Therefore, the Brownian Motion-based mutant strategy was considered in combination with the HHO algorithm (HHOBM) to solve non-convex optimization problems. Taking advantage of the strong optimality-seeking ability of the HHO algorithm and the based on Brownian motion mutant strategy to help the algorithm fill out the local optimum, the proposed HHOBM algorithm is promising in finding the global optimal solution of a non-convex optimization problem. The pseudo-code of the HHOBM is given by Algorithm 2.

3.3. Computational complexity

The computational complexity of HHOBM can be calculated based on its main processes: population initialization, fitness calculation, the updated mechanism, mutation strategy. The complexity of HHOBM for a D-dimensional optimization problem with N hawks and a number of iterations T is calculated as follows. The computational complexity of the population initialization is $O(N)$. The computational complexity of the updating mechanism and fitness calculation is $O(T \times N) + O(T \times D \times N)$. The computational complexity of mutation strategy is $O(N \times T)$. Therefore, the total computational complexity of HHOBM is $O(N \times (2T + TD + 1))$.

4. Experimental results

4.1. Benchmark functions and experimental design

To verify the performance of HHOBM, the experiments are designed based on 44 non-convex benchmark functions [8] and 10 modern CEC test functions [35]. These functions are classified into three groups: (i) non-convex fixed dimension benchmark functions ($f_1 - f_{27}$); (ii) non-convex variable dimension benchmark functions ($f_{28} - f_{44}$);

Algorithm 2 HHOBM

Input: The population size$N$ and maximum number of iterations T
Output: The location of rabbit and its fitness value

Initialize the position of hawks $X_i(i = 1, 2, \cdots, N)$
While(the end condition s not met) do
  FOR $i = 1 : N$
  Calculate the fitness values of hawks
  Update the location of rabbit X_{rabbit}
  END FOR
  For (each hawk) do
    Update the initial energyE_0 and jump strength$J$ $\%E_0 = randn(), J = 2(1 - rand())$
    Update the E using Eq.(1)
    if($|E| \geq 1$) then
      Update the location of hawks using Eq.(2)
    if($|E| < 1$) then
      if ($r \geq 0.5 and |E| \geq 0.5$) then
        Update the location of hawks using Eq.(4)
      Else if ($r \geq 0.5 and |E| < 0.5$) then
        Update the location of hawks using Eq.(6)
      if ($r < 0.5 and |E| < 0.5$) then
        Update the location of hawks using Eq.(14)
      Else if ($r < 0.5 and |E| \geq 0.5$) then
        Update the location of hawks using Eq.(11)
  Update the location of hawks using Algorithm.1
  END FOR
  END While
Return X_{rabbit}

(iii) modern CEC benchmark test functions ($CEC01 - CEC10$). The names of these functions and their associated analytic properties are displayed in Tables A.12, A.13, and A.14 in Appendix.

Each experiments include three parts. Firstly, the scalability is tested on HHOBM. Secondly, the performance comparisons are made with well-known algorithms such as the PSO [28], DE [36], FA [53], BAT [54], WOA [34] and some of the state-of-the-art methods. Finally, HHOBM is applied to solve two classic engineering design problems. The parameters of all algorithms are displayed in Table 2. All experimental results are obtained by running 30 times independently to calculate the mean value. The population size of all algorithms is chosen to be 30, the maximum number of iterations is designed to be 100, and all experiments are run with matlab 2018b under Windows 10 64-bit professional and 4 GB RAM.

4.2. The scalability test on HHOBM

In this subsection, the HHOBM and the original HHO are used for scalability experiments based on non-convex variable dimension benchmark functions ($f_{28} - f_{44}$) with the dimensions of 30, 50 and 100. The experimental results of scalability tests are recorded in Table 1. In Table 1, the optimal solution obtained by two algorithms of the same dimension for the same problem is highlighted. The last column counts the number of problems that are optimal/equal/poor compared to the two algorithms under each dimension.

The results in Table 1 show that in 30 dimensions HHOBM produces the same results as HHO for f_{29}, f_{30}, f_{34}, f_{38} and f_{39}. For f_{33} and f_{36}, HHOBM produces worse results, and HHOBM produces better results for all the other problems. In 50 dimensions, except for f_{36}, HHOBM has the same or better results as HHO for all other problems. In 100 dimensions, HHOBM has better or same results on all problems but for f_{31}, f_{32} and f_{33}. From this, it can be seen that HHOBM can jump out of the local optimum more easily and has better scalability performance.

Table 1
Experimental results of scalability tests in different dimensions.

Problem	HHOBM			HHO		
	30	50	100	30	50	100
f_{28}	**6.06E−03**	**1.20E−02**	**1.07E−02**	9.00E−03	1.38E−01	1.91E−02
f_{29}	**0**	**0**	**0**	**0**	**0**	**0**
f_{30}	**0.9**	**0.9**	**0.9**	**0.9**	**0.9**	**0.9**
f_{31}	**456.24868**	**4178.4273**	56 168.0437	487.760315	4250.47579	**55 564.618**
f_{32}	**4.81E−49**	**7.35E−53**	2.96E−06	1.31E−06	3.59E−06	**1.20E−48**
f_{33}	1.92E−12	**8.27E−17**	7.95E−12	**1.27E−22**	9.26E−16	**8.38E−23**
f_{34}	**−8.88E−16**	**−8.88E−16**	**−8.88E−16**	**−8.88E−16**	**−8.88E−16**	**−8.88E−16**
f_{35}	**1.0000117**	**1.000036**	**1.0001145**	1.00103107	1.00248279	1.002568103
f_{36}	7.96E−50	1.72E−46	**8.32E−49**	**2.21E−51**	**4.45E−50**	1.15E−47
f_{37}	**−1174.985**	**−1958.307**	**−3916.615**	−1174.9774	−1958.2753	−3916.60116
f_{38}	**0**	**0**	**0**	**0**	**0**	**0**
f_{39}	**−1**	**−1**	**−1**	**−1**	**−1**	**−1**
f_{40}	**3.51E−12**	**1.21E−20**	**4.66E−42**	3.53E−12	**1.21E−20**	4.69E−42
f_{41}	**1.41E−06**	**3.68E−06**	**8.05E−06**	8.15E−05	1.21E−04	1.35E−04
f_{42}	**3.58E−07**	**1.79E−07**	**7.92E−08**	7.61E−06	4.76E−06	2.16E−06
f_{43}	**−13.39934**	**−19.17868**	**−31.80769**	−10.937886	−16.570264	−24.9759933
f_{44}	**7.23E−05**	**2.14E−04**	**1.25E−04**	9.28E−05	1.30E−04	1.99E−04
+/=/−	10/5/2	10/6/1	9/5/3	2/5/10	1/6/10	3/5/9

Table 2
Parameter setting of various optimization algorithm.

Algorithm	Parameter	Value
DE	Scaling factor	0.5
	Crossover probability	0.5
PSO	Inertia factor	0.3
	c1	1
	c2	1
BA	Qmin frequency minimum	0
	Qmax frequency maximum	2
	A Loudness	0.5
	r Pulse rate	0.5
FPA	Probability switch p	0.8
WOA	a	[0, 2]
	b	1
	l	[−1, 1]
HHO	∼	∼

4.3. Comparison with well-known algorithms

In this subsection, HHOBM is compared with some classical algorithms such as PSO [28], HHO [21], WOA [34], BAT [54], FPA [53]. The experiments were conducted using the 54 non-convex functions described above, and each algorithm was run 30 times independently. In order to better analyze the performance among the algorithms, Wilcoxon signed-rank test is used to assess whether there is a significant difference among the algorithms and the Friedman test is used to rank the performance of the algorithm for each participating experiment [13,19]. All the experimental results are shown at Tables 3, 4 and 5. The p-values of the Wilcoxon signed rank test are presented in Tables 6, 7 and 8. Figs. 6–8 show the result of average raking value for non-convex fixed and variable dimension benchmark functions, modern CEC benchmark test functions.

The results indicate that the HHOBM algorithm is significantly better than the other algorithms in terms of performance as can be seen in Tables 6, 7 and 8. Note that the benchmark functions chosen for the experiments

Table 3
Results of a comparison with the algorithms based on non-convex fixed dimension benchmark functions.

	HHOBM	HHO	WOA	PSO	FPA	BAT
f_1	8.498E−28	2.298E−26	4.292E−23	**2.550E−45**	1.236E+01	7.115E+00
f_2	**−1.956E+02**	−1.956E+02	−1.956E+02	**−1.956E+02**	−1.718E+02	−1.908E+02
f_3	**−2.022E+00**	**−2.022E+00**	**−2.022E+00**	**−2.022E+00**	−1.633E+00	−1.518E+00
f_4	**−1.068E+02**	−1.058E+02	−1.038E+02	−1.029E+02	−5.055E+01	−9.566E+01
f_5	**−1.032E+00**	**−1.032E+00**	**−1.032E+00**	**−1.032E+00**	1.889E−01	−4.732E−01
f_6	3.979E−01	3.980E−01	3.988E−01	**3.979E−01**	3.056E+00	5.176E−01
f_7	**3.000E+00**	**3.000E+00**	4.365E+00	1.110E+01	1.131E+02	1.151E+02
f_8	−3.853E+00	−3.837E+00	−3.751E+00	**−3.860E+00**	−3.323E+00	−2.308E+00
f_9	−3.056E+00	−2.929E+00	−3.062E+00	**−3.144E+00**	−1.611E+00	−6.118E−01
f_{10}	**−2.063E+00**	**−2.063E+00**	**−2.063E+00**	**−2.063E+00**	−1.978E+00	−2.028E+00
f_{11}	**1.000E+00**	**1.000E+00**	**1.000E+00**	4.724E+03	9.774E+03	1.000E+00
f_{12}	**1.803E+02**	**1.803E+02**	**1.803E+02**	**1.803E+02**	**1.803E+02**	3.254E−01
f_{13}	**−2.416E+01**	**−2.415E+01**	**−2.415E+01**	−1.546E+01	−1.834E+01	**−2.416E+01**
f_{14}	−4.281E+01	−4.277E+01	−4.254E+01	**−4.294E+01**	−3.162E+01	−4.114E+01
f_{15}	**4.848E−05**	**4.848E−05**	**4.848E−05**	**4.848E−05**	5.118E−05	4.959E−05
f_{16}	**−2.002E−01**	−2.843E−03	−3.412E−04	−8.880E−03	−1.082E−04	−3.000E−04
f_{17}	2.714E−02	3.755E−02	2.943E−01	**2.402E−02**	1.034E+00	3.483E−01
f_{18}	**−9.999E−01**	−9.998E−01	−8.994E−01	−4.922E−01	−5.380E−60	−5.500E−01
f_{19}	**6.447E−02**	**6.447E−02**	**6.447E−02**	**6.447E−02**	1.031E−01	1.022E−01
f_{20}	**1.408E−01**	3.552E−01	2.788E+00	7.610E−01	6.115E+02	1.321E+00
f_{21}	2.231E−05	2.216E−03	5.173E−03	**2.367E−31**	6.409E+00	1.928E−06
f_{22}	**−1.921E+01**	**−1.921E+01**	**−1.921E+01**	−1.661E+01	−1.577E+01	−1.891E+01
f_{23}	**−9.635E−01**	**−9.635E−01**	**−9.635E−01**	−9.541E−01	−9.314E−01	**−9.635E−01**
f_{24}	**−1.087E+01**	−1.086E+01	−1.085E+01	−1.086E+01	−9.789E+00	−1.064E+01
f_{25}	**−1.867E+02**	−1.867E+02	−1.866E+02	−1.867E+02	−7.727E+01	−1.228E+02
f_{26}	**−9.924E+00**	−5.546E+00	−5.425E+00	−6.756E+00	−9.510E−01	−4.926E+00
f_{27}	3.813E−28	2.770E−27	1.195E−01	**5.108E−47**	5.775E−01	1.195E−01
+/=/−	∼	0/13/14	2/11/14	8/7/9	0/1/26	0/2/25
AVR	2.11	2.83	3.30	2.63	5.59	4.61
Rank	1	3	4	2	6	5

are all non-convex functions, which better test the algorithm's ability to jump out of the local optimum. In Tables 3, 4 and 5, "+/=/−" statistics indicate whether HHOBM is better than, the same as, inferior to other algorithms, respectively. Out of 54 test problems, the HHOBM algorithm outperformed other algorithms in up to 24 problems, had the same result in 11 problems, and had worse results than other algorithms in 19 problems. In the test of non-convex fixed dimension benchmark functions, HHOBM has better or equal results compared to other algorithms except f_1, f_6, f_8, f_9, f_{14}, f_{17}, f_{21} and f_{27}. From Table 4, it can be found that except for f_{31}, f_{34}, f_{36}, and f_{43}, HHOBM has better or equal results compared to other algorithms for non-convex variable dimension benchmark functions. In the experiment of modern CEC benchmark test functions, it can be seen that except for $CEC02$, $CEC04$, $CEC05$, $CEC09$ and $CEC10$, HHOBM has better or same results compared to other algorithms. The Friedman test ranking of the algorithms is counted in the "AVR" row and the "Rank" row, it can be seen that the HHOBM algorithm ranks first in the three groups of experiments with 2.11, 1.35, and 1.65. This also demonstrates that HHOBM performs much better than other algorithms and has a strong ability to jump out of local optimum.

4.4. Comparison with some of the state-of-the-art methods

In this subsection, for further validating the performance of HHOBM, HHOBM is compared with some of the state-of-the-art algorithms published in the literature, including multi-search Arithmetic Optimization Algorithm (IAOA) [2], hybrid algorithm based on artificial bee colony and Fibonacci indicator algorithm (ABFIA) [15], boosted Harris Hawks gravitational force algorithm (HHMV) [4], improved gradual change-based Harris Hawks optimization (HHSC) [3], novel PSO using prey–predator relationship (PP-PSO) [57]. The experiment uses ten CEC2019 test functions, with details shown in Table A.14 of Appendix. Table 9 shows the results obtained by the proposed

Table 4
Results of a comparison with the algorithms based on non-convex variable dimension benchmark functions.

	HHOBM	HHO	WOA	PSO	FPA	BAT
f_{28}	**1.05E+01**	2.16E+01	1.25E+02	3.36E+02	238.3937	417.1359
f_{29}	**0**	**0**	4.97E+00	129.3814	449.9257	681.1741
f_{30}	**0.9**	**0.9**	2.01136591	5.043522	9.8736	1.000413
f_{31}	1167.01183	**1164.333**	5173.23764	9.3E+10	2.02E+11	3991.573
f_{32}	**1.39E−13**	4.74E−05	4.93E−01	8.828189	65.1749	20.3474
f_{33}	**2.50E−11**	6.58E−10	2.43E+00	8.577722	1.07E+14	8.07E+19
f_{34}	8.02E−13	**5.15E−13**	7.15E−07	7.726958	19.96677	11.95109
f_{35}	**1.0003731**	1.5259108	94.3167624	1 206 866	1 652 290	719.6931
f_{36}	1.54E−10	**1.09E−12**	0.15488601	5.183064	27.07417	2.689873
f_{37}	**−1174.971**	−1174.683	−964.10527	−845.667	−430.837	1578.14
f_{38}	**0**	**0**	1.57E−11	1.384457	17.38934	1.098581
f_{39}	**−1**	**−1**	−3.50E−01	3.92E−12	1.09E−07	3.18E−10
f_{40}	**3.51E−12**	3.63E−12	4.10E−12	1.45E−08	3.38E−11	1.86E−10
f_{41}	**5.16E−05**	1.97E−03	1.24154913	308.6886	1.09E+09	155.8715
f_{42}	**8.26E−06**	1.55E−04	2.20E−01	6.31E+00	5.71E+08	5.94E+00
f_{43}	−11.68908	−9.342493	−9.6488477	**−13.217**	−6.43815	−7.10102
f_{44}	**8.30E−04**	1.19E−03	0.01809964	1.94968	126.5081	109 416.1
+/=/−	∼	3/4/10	0/0/17	1/0/16	0/0/17	0/0/17
AVR	1.35	1.76	3.35	4.47	4.82	5.24
Rank	1	2	3	4	5	6

Table 5
Results of a comparison with the algorithms based on modern CEC benchmark test functions.

	HHOBM	HHO	WOA	PSO	FPA	BAT
$CEC01$	**6.97E+04**	7.12E+04	2.64E+11	4.32E+12	6.66E+12	929 205.9
$CEC02$	17.3874314	**17.3815**	1.76E+01	15 617.35	16 577.03	17.6889
$CEC03$	**12.7024415**	12.702465	12.70242017	12.70253	12.70696	12.70617
$CEC04$	3040.793676	3978.1495	2330.482372	**14 526.3**	31 673.83	56 857.23
$CEC05$	2.98E+00	3.08E+00	**2.30E+00**	3.793746	8.272616	7.075883
$CEC06$	**1.03E+01**	1.07E+01	1.07E+01	11.49985	1.60E+01	7.91E+00
$CEC07$	**4.93E+02**	5.22E+02	7.41E+02	1216.239	2191.787	591.1975
$CEC08$	**6.00734794**	6.1565878	6.223374337	6.581902	8.608144	6.35461
$CEC09$	2.61E+02	2.73E+02	**189.126102**	1277.14	6569.225	4616.421
$CEC10$	20.34485308	20.438669	20.43935272	20.57631	21.13335	**20.1899**
+/=/−	∼	1/0/9	2/0/8	1/0/9	0/0/10	1/0/9
AVR	1.65	2.7	2.4	4.25	5.6	4.3
Rank	1	3	2	4	6	5

HHOBM and best-published results using CEC2019. Also, the average ranking value and the final ranking of the algorithm are recorded in Table 9. Result of average ranking value is shown in Fig. 9.

The results clearly demonstrate that HHOBM is excellent compared to the other state-of-the-art methods taken from previous studies. Except for $CEC03, CEC04, CEC05, CEC06, CEC09$ and $CEC10$, HHOBM produced better or similar results compared to other state-of-the-art algorithms. The Friedman test ranking of the algorithms is counted in the "AVR" row and the "Rank" row, which show that the HHOBM algorithm ranks first with 3. This also demonstrates that HHOBM has a strong competitive edge even comparing with some of the state-of-the-art methods.

4.5. HHOBM for solving engineering design problems

To verify the generalizability and effectiveness of the proposed algorithm, HHOBM is applied to solve two engineering design problems: Pressure vessel design problem [56] and Cantilever beam design problem [33].

Table 6
The p-values of the Wilcoxon signed rank test based on modern CEC benchmark test functions.

	HHOBM	HHO	WOA	PSO	FPA	BAT
$CEC01$	N/A	1.87E−02	8.86E−05	8.86E−05	8.86E−05	8.86E−05
$CEC02$	N/A	1.87E−02	1.40E−04	8.86E−05	8.86E−05	2.19E−04
$CEC03$	N/A	3.04E−02	2.50E−03	8.86E−05	8.86E−05	8.86E−05
$CEC04$	N/A	1.87E−02	6.74E−02	8.86E−05	8.86E−05	8.86E−05
$CEC05$	N/A	9.30E−02	3.59E−03	1.32E−03	1.03E−04	8.86E−05
$CEC06$	N/A	3.04E−02	8.86E−05	8.92E−04	8.86E−05	7.80E−04
$CEC07$	N/A	8.86E−05	6.42E−03	8.86E−05	8.86E−05	8.86E−05
$CEC08$	N/A	8.86E−05	2.06E−02	2.06E−02	8.86E−05	8.86E−05
$CEC09$	N/A	8.86E−05	1.26E−01	8.86E−05	8.86E−05	8.86E−05
$CEC10$	N/A	4.49E−04	2.82E−03	8.86E−05	8.86E−05	1.87E−02

Table 7
The p-values of the Wilcoxon signed rank test based on non-convex variable dimension benchmark functions.

	HHOBM	HHO	WOA	PSO	FPA	BAT
f_{28}	N/A	1.32E−04	8.86E−05	8.86E−05	8.86E−05	8.86E−05
f_{29}	N/A	1.32E−04	8.84E−05	8.86E−05	8.86E−05	8.86E−05
f_{30}	N/A	1.32E−04	1.32E−04	8.86E−05	7.74E−06	8.86E−05
f_{31}	N/A	9.30E−02	8.86E−05	8.86E−05	8.86E−05	8.86E−05
f_{32}	N/A	9.30E−02	1.51E−03	8.86E−05	8.86E−05	8.86E−05
f_{33}	N/A	9.30E−02	1.03E−04	8.86E−05	8.86E−05	8.86E−05
f_{34}	N/A	8.86E−05	8.86E−05	8.86E−05	8.55E−05	8.86E−05
f_{35}	N/A	8.86E−05	8.86E−05	8.86E−05	8.86E−05	8.86E−05
f_{36}	N/A	7.31E−02	8.86E−05	8.86E−05	8.86E−05	8.86E−05
f_{37}	N/A	2.06E−02	8.86E−05	8.86E−05	8.86E−05	8.86E−05
f_{38}	N/A	9.30E−02	8.86E−05	8.86E−05	8.86E−05	8.86E−05
f_{39}	N/A	8.86E−05	8.86E−05	8.86E−05	8.86E−05	8.86E−05
f_{40}	N/A	2.54E−04	8.86E−05	8.86E−05	8.86E−05	8.86E−05
f_{41}	N/A	1.40E−04	8.86E−05	8.86E−05	8.86E−05	8.86E−05
f_{42}	N/A	8.92E−04	8.86E−05	8.86E−05	8.86E−05	8.86E−05
f_{43}	N/A	1.03E−04	1.89E−04	3.04E−02	8.86E−05	8.86E−05
f_{44}	N/A	8.86E−05	1.63E−04	8.86E−05	8.86E−05	8.86E−05

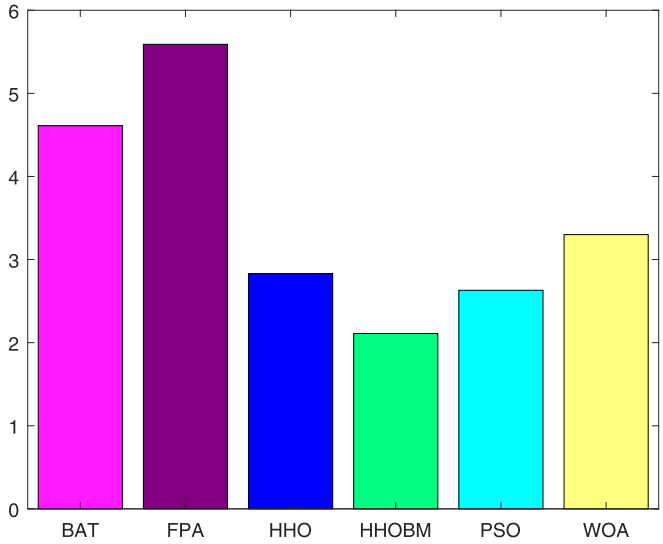

Fig. 6. Result of average ranking value for non-convex fixed dimension benchmark functions.

Table 8
The p-values of the Wilcoxon signed rank test based on non-convex fixed dimension benchmark functions.

	HHOBM	HHO	WOA	PSO	FPA	BAT
F1	N/A	4.05E−03	5.17E−04	8.86E−05	8.86E−05	8.86E−05
F2	N/A	4.05E−03	1.16E−03	8.83E−05	8.86E−05	2.19E−04
F3	N/A	8.86E−05	8.86E−05	8.86E−05	8.86E−05	8.86E−05
F4	N/A	1.03E−04	8.86E−05	7.31E−02	8.86E−05	1.37E−02
F5	N/A	4.05E−03	4.79E−02	1.32E−04	8.86E−05	1.20E−04
F6	N/A	5.17E−04	1.40E−04	8.86E−05	8.86E−05	1.94E−03
F7	N/A	1.03E−04	1.03E−04	5.17E−04	8.86E−05	8.86E−05
F8	N/A	5.17E−04	7.31E−02	2.96E−01	8.86E−05	2.54E−04
F9	N/A	5.17E−04	1.94E−03	3.19E−03	8.86E−05	1.40E−04
F10	N/A	3.04E−02	4.79E−02	8.86E−05	8.86E−05	8.23E−01
F11	N/A	5.17E−04	1.88E−01	8.86E−05	8.86E−05	8.86E−05
F12	N/A	5.17E−04	5.17E−04	5.17E−04	5.17E−04	8.86E−05
F13	N/A	1.71E−03	4.38E−02	1.00E−02	8.86E−05	3.13E−01
F14	N/A	9.46E−02	1.87E−02	4.47E−04	8.86E−05	1.79E−01
F15	N/A	9.30E−02	9.30E−02	8.86E−05	8.86E−05	7.09E−01
F16	N/A	9.30E−02	8.86E−05	4.33E−01	8.86E−05	8.86E−05
F17	N/A	9.30E−02	8.86E−05	6.74E−02	8.86E−05	8.86E−05
F18	N/A	9.30E−02	2.76E−02	1.02E−03	8.86E−05	5.17E−04
F19	N/A	1.87E−02	5.17E−04	8.86E−05	8.86E−05	2.93E−04
F20	N/A	2.76E−02	1.00E−02	2.76E−02	8.86E−05	3.33E−02
F21	N/A	2.82E−03	8.59E−02	8.86E−05	8.86E−05	1.08E−01
F22	N/A	1.02E−03	1.03E−04	8.86E−05	8.86E−05	1.16E−03
F23	N/A	2.76E−02	1.00E−02	7.31E−02	8.86E−05	3.19E−03
F24	N/A	2.76E−02	2.51E−02	6.01E−01	8.86E−05	1.40E−04
F25	N/A	2.76E−02	1.71E−03	8.86E−05	8.86E−05	2.82E−03
F26	N/A	8.86E−05	5.93E−04	2.76E−02	8.86E−05	1.51E−03
F27	N/A	7.93E−02	8.86E−05	8.86E−05	8.86E−05	8.86E−05

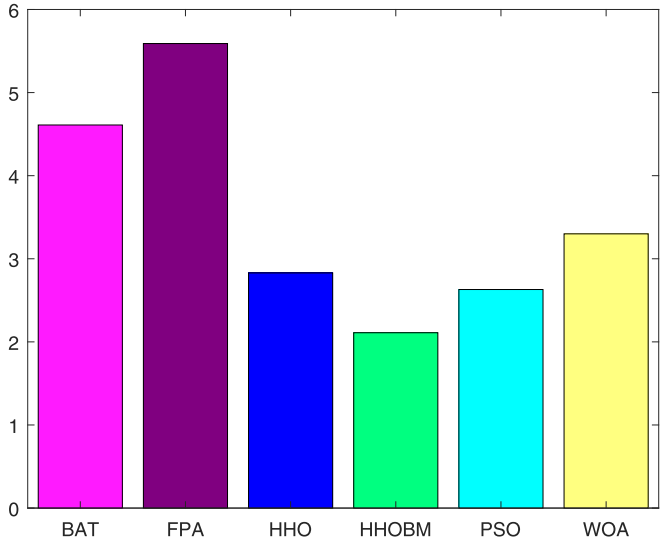

Fig. 7. Result of average ranking value for non-convex variable dimension benchmark functions.

4.5.1. Pressure vessel design problem

The pressure vessel design problem (PVD) is a well known engineering design problem proposed by Kannan and Kramer [38]. The structure of the pressure vessel is shown in Fig. 10. In this problem, the thickness of the shell (T_s),

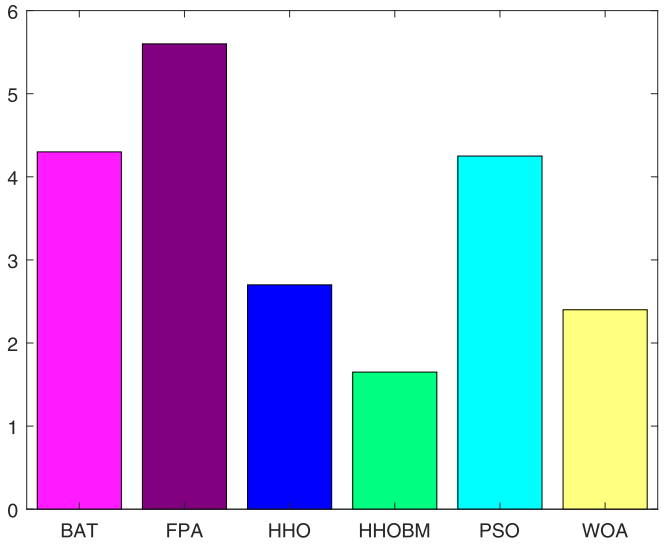

Fig. 8. Result of average ranking value for modern CEC benchmark test functions.

Table 9
The results obtained by the proposed HHOBM and best-published results using CEC2019.

	HHOBM	IAOA	ABFIA	HHMV	HHSC	PP-PSO
CECO1	**5.57E+04**	5.00E+04	6.98E+04	**5.57E+04**	6.97E+04	2.35E+08
CECO2	**1.73E+01**	**1.73E+01**	**1.73E+01**	1.74E+01	1.74E+01	2.66E+04
CECO3	1.27E+01	1.27E+01	1.27E+01	1.27E+01	1.27E+01	**6.64E+00**
CECO4	3.04E+03	5.22E+01	**2.09E+01**	6.17E+01	4.24E+02	4.77E+01
CECO5	2.98E+00	3.42E+00	3.48E+00	2.92E+00	2.98E+00	**1.66E+00**
CECO6	1.03E+01	7.06E+00	5.22E+00	4.76E+00	**4.73E+00**	8.16E+00
CECO7	**4.93E+02**	5.48E+02	5.01E+02	5.01E+02	5.88E+02	1.12E+03
CECO8	**3.91E+00**	4.33E+00	4.37E+00	4.79E+00	5.15E+00	4.40E+00
CECO9	2.21E+02	3.18E+02	3.18E+02	3.89E+02	4.72E+02	**1.53E+02**
CECO10	1.93E+01	1.99E+01	**1.63E+01**	2.05E+01	2.00E+01	2.14E+01
+/=/−	∼	2/2/6	3/2/5	1/2/7/	2/1/7	4/0/6
AVR	3	3.45	3.2	3.55	4.3	3.8
Rank	1	3	2	4	6	5

thickness of the head (T_h), and the length of the cylindrical part (L) need to be optimized in order to minimize the cost of forming and welding materials for the pressure vessel. The mathematical model is as follows:

$$
\begin{aligned}
&\text{Consider } \vec{x} = [x_1, x_2, x_3, x_4] = [T_s, T_h, R, L] \\
&\text{Minimize } f(\vec{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_2^3 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \\
&\text{Subject to} \\
&g_1(\vec{x}) = -x_1 + 0.0193x_3 \leq 0 \\
&g_2(\vec{x}) = -x_3 + 0.00954x_3 \leq 0 \\
&g_3(\vec{x}) = -\Pi x_3^2 x_4 - \tfrac{4}{3}\Pi x_3^3 + 1,296,000 \leq 0 \\
&g_4(\vec{x}) = x_4 - 240 \leq 0
\end{aligned}
\tag{18}
$$

The range of the decision variables x_1, x_2, x_3, x_4 are $[0, 99]$, $[0, 99]$, $[10, 200]$, $[10, 200]$. HHOBM, HHO and some recently proposed algorithms are used to solve this optimization problem. Table 10 shows the results of comparing HHOBM with other algorithms for pressure vessel design problems. The best design solution is obtained by the HHOBM. This shows that the improved algorithm is promising in dealing with optimization problems with

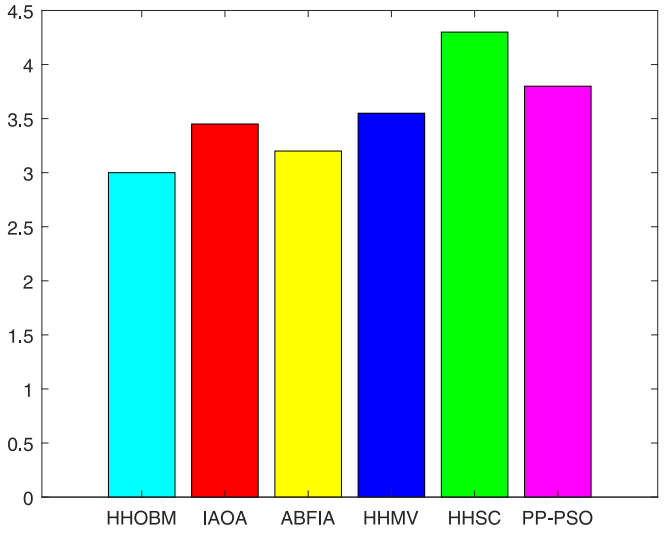

Fig. 9. Result of average ranking value for comparing with the state-of-the-art methods.

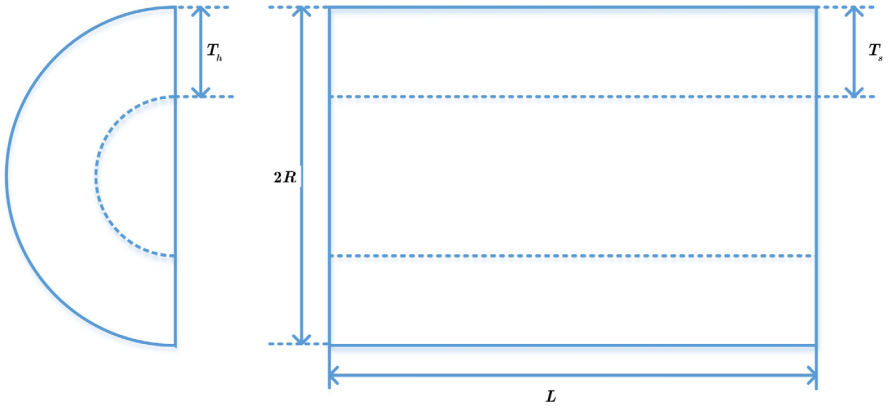

Fig. 10. The structure of the pressure vessel.

constraints, and again proves that the algorithm is highly capable of dealing with complex nonlinear optimization problems.

4.5.2. Cantilever beam design problem

Cantilever beam design problem is a well-known engineering optimization problem that is composed of five parts, each of which is a square. The structure of the cantilever beam is shown in Fig. 11. The optimization objective of this design problem is to satisfy the load carrying capacity while minimizing the total mass of the cantilever beam. The mathematical model of the problem is as follows:

$$\begin{aligned} &\text{Consider } \vec{x} = [x_1, x_2, x_3, x_4, x_5] \\ &\text{Minimize } f(\vec{x}) = 0.6224(x_1 + x_2 + x_3 + x_4 + x_5) \\ &\text{Subject to} \\ &g(\vec{x}) = \frac{61}{x_1^3} + \frac{27}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0 \end{aligned} \tag{19}$$

Table 10
Results of HHOBM versus other algorithms for pressure vessel design problem.

Algorithm	Decision variables				Cost
	T_s	T_h	R	L	
HHOBM	0.8312	0.4084	44.6683	147.1969	5849.9629
HHO	0.8181	0.4080	42.0919	175.6138	5987.3830
EWOA [52]	0.8112	0.4249	42.0808	176.8759	5862.3411
ASOINU [43]	0.8845	0.4220	45.3368	140.2538	6090.5262
IEHHO [38]	0.8229	0.4068	42.6371	170.0741	5966.3010
DSLC-FOA [14]	0.7804	0.3849	40.3888	199.1172	5894.5981
QOCSOS [45]	0.7782	0.3848	40.322	199.9667	5885.3327

Table 11
Results of HHOBM versus other algorithms for cantilever beam design problem.

Algorithm	Decision variables					Cost
	x_1	x_2	x_3	x_4	x_5	
HHOBM	5.8830	4.8123	4.6712	3.4479	2.1476	13.0468
HHO	6.0689	4.8789	4.3746	3.6559	2.0033	13.0589
WLSSA [39]	6.1349	5.3604	4.4390	3.5105	2.0103	13.3880
EWOA [52]	6.0187	5.3148	4.4913	3.4991	2.1523	13.3996
ELPSO [59]	6.0160	5.3092	4.4943	3.5015	2.1527	13.4000
MMA [12]	6.0100	5.3000	4.4900	3.4900	2.1500	13.4000
SMA [30]	6.0177	5.3108	4.4937	3.5011	2.1501	13.3996

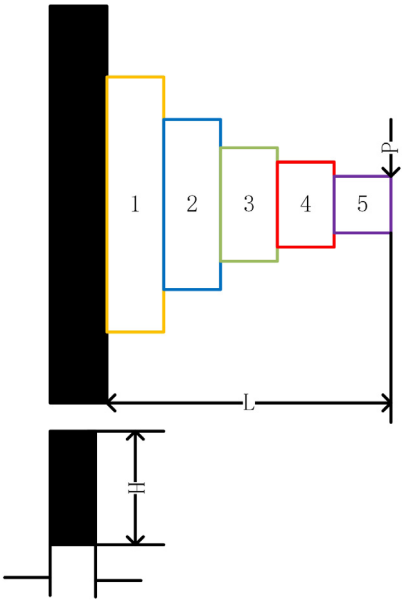

Fig. 11. The structure of the pressure vessel.

where, x_1, x_2, x_3, x_4, x_5 are required to satisfy $0 \leq x_1, x_2, x_3, x_4, x_5 \leq 100$. Table 11 shows the results of HHOBM versus other algorithms for the cantilever beam design problem. The best design solution is obtained by the HHOBM algorithm, which once again proves that the HHO algorithm combined with the Brownian motion-based mutation strategy performs well in robustness and convergence and can effectively solve engineering optimization problems.

5. Conclusions and future work

The HHO algorithm is a population-based meta heuristic algorithm with a simple structure and fast convergence, and it does not require a lot of user input parameters. However, when dealing with non-convex optimization problems, the original HHO algorithm tends to fall into the local optimum trap. In this paper, based on Brownian motion, new mutation strategy was proposed. It is then combined with the basic HHO to form an improved algorithm: Harris hawks optimization based on Brownian Motion mutant strategy (HHOBM). The Brownian motion-based mutation strategy uses the randomness of Brownian motion to help the algorithm jump out of the local optimum trap and thus help the algorithm find the global minimum.

In order to verify the performance of the HHOBM, we compared the performance of our algorithm with a series of optimization algorithms using 54 non-convex function optimization problems and two engineering design optimization problems. The numerical results show that in general, the HHOBM is able to find the global optimal solution more efficiently when dealing with non-convex optimization problems and to obtain a better design solution when dealing with engineering optimization problems. Although the HHOBM has better results in numerical experiments, we found that there is still room for improvement in balancing the exploration and exploitation phases of the improved algorithm. In our future work, we will study the balance between the exploration phase and exploitation phase of the HHO algorithm to design more effective mutation strategies.

CRediT authorship contribution statement

Helei Kang: Conceptualization, Methodology, Software, Writing – original draft. **Renyun Liu:** Visualization, Investigation, Writing – original draft, Funding acquisition. **Yifei Yao:** Visualization, Investigation. **Fanhua Yu:** Supervision, Writing – review & editing.

Acknowledgments

The authors would like to thank the editor and anonymous referees for the constructive comments and suggestions. This work was supported by the Jilin Scientific and Technological Development Program, PR China (Grant Nos. 20180101224JC and 20200201276JC).

Appendix

See Tables A.12–A.14

Table A.12
Descriptions of non-convex fixed-dimension benchmark functions.

Problem	Name	dim	Range	fmin
f_1	Egg Crate	2	[−5,5]	0
f_2	Ackley N.3	2	[−32,32]	−195.629
f_3	Adjiman	2	[−1,2]	−2.02181
f_4	Bird	2	[−2pi,2pi]	−106.765
f_5	Camel 6 Hump	2	[−5,5]	−1.0316
f_6	Branin RCOS	2	[−5,5]	0.397887
f_7	Goldstien Price	2	[−2,2]	3
f_8	Hartman 3	3	[0,1]	−3.86278
f_9	Hartman 6	6	[0,1]	−3.32236
f_{10}	Cross-in-tray	2	[−10,10]	−2.06261
f_{11}	Bartels Conn	2	[−500,500]	1
f_{12}	Bukin 6	2	[(−15,5),(−5,−3)]	180.3276
f_{13}	Carrom Table	2	[−10,10]	−24.1568
f_{14}	Chichinadze	2	[−30,30]	−43.3159
f_{15}	Cross function	2	[−10,10]	0

(*continued on next page*)

Table A.12 (*continued*).

Problem	Name	dim	Range	fmin
f_{16}	Cross leg table	2	[−10,10]	−1
f_{17}	Crowned cross	2	[−10,10]	0.0001
f_{18}	Easom	2	[−100,100]	−1
f_{19}	Giunta	2	[−1,1]	0.060447
f_{20}	Helical Valley	3	[−10,10]	0
f_{21}	Himmelblau	2	[−5,5]	0
f_{22}	Holder	2	[−10,10]	−19.2085
f_{23}	Pen Holder	2	[−11,11]	−0.96354
f_{24}	Test Tube Holder	2	[−10,10]	−10.8723
f_{25}	Shubert	2	[−10,10]	−186.731
f_{26}	Shekel	4	[0,10]	−10.5364
f_{27}	Three-Hump Camel	2	[−5,5]	0

Table A.13
Descriptions of non-convex variable-dimension benchmark functions.

Problem	Name	dim	Range	fmin
f_{28}	Schwefe's 2.26	30,50,100	[−500,500]	−418.983
f_{29}	Rastrigin	30,50,100	[−5.12,5.12]	0
f_{30}	Periodic	30,50,100	[−10,10]	0.9
f_{31}	Qing	30,50,100	[−500,500]	0
f_{32}	Alpine N.1	30,50,100	[−10,10]	0
f_{33}	Xin-She Yang	30,50,100	[−5,5]	0
f_{34}	Ackley	30,50,100	[−32,32]	0
f_{35}	Trignometric 2	30,50,100	[−500,500]	0
f_{36}	Salomon	30,50,100	[−100,100]	0
f_{37}	Styblinski-Tang	30,50,100	[−5,5]	−1174.98
f_{38}	Griewank	30,50,100	[−100,100]	0
f_{39}	Xin-She Yang N. 4	30,50,100	[−10,10]	−1
f_{40}	Xin-She Yang N. 2	30,50,100	[−2pi,2pi]	0
f_{41}	Gen. Penalized	30,50,100	[−50,50]	0
f_{42}	Penalized	30,50,100	[−50,50]	0
f_{43}	Michalewics	30,50,100	[0,pi]	−29.6309
f_{44}	Quartic Noise	30,50,100	[−1.28,1.28]	0

Table A.14
Description of CEC-C06 2019 functions.

Problem	Name	Range	dim	fmin
1	Storn's Chebyshev polynomial fitting problem	[−8192,8192]	9	1
2	Inverse Hilbert matrix problem	[−16 384,16 384]	16	1
3	Lennard-Jones minimum energy cluster	[−4,4]	18	1
4	Rastrigin's function	[−100,100]	10	1
5	Griewangk's function	[−100,100]	10	1
6	Weierstrass function	[−100,100]	10	1
7	Modified Schwefel's function	[−100,100]	10	1
8	Expanded Schwefel's F6 function	[−100,100]	10	1
9	Happy Cat function	[−100,100]	10	1
10	Ackley function	[−100,100]	10	1

References

[1] M. Abdel-Basset, W. Ding, D. El-Shahat, A hybrid Harris Hawks optimization algorithm with simulated annealing for feature selection, Artif. Intell. Rev. 54 (1) (2021) 593–637.
[2] L. Abualigah, K.H. Almotairi, M.A. Al-qaness, A.A. Ewees, D. Yousri, M. Abd Elaziz, M.H. Nadimi-Shahraki, Efficient text document clustering approach using multi-search Arithmetic Optimization Algorithm, Knowl.-Based Syst. 248 (2022) 108833.
[3] L. Abualigah, A. Diabat, M. Altalhi, M.A. Elaziz, Improved gradual change-based Harris Hawks optimization for real-world engineering design problems, Eng. Comput. (2022) 1–41.

[4] L. Abualigah, A. Diabat, D. Svetinovic, M.A. Elaziz, Boosted Harris Hawks gravitational force algorithm for global optimization and industrial engineering problems, J. Intell. Manuf. (2022) 1–36.
[5] I. Ahmadianfar, A.A. Heidari, A.H. Gandomi, X. Chu, H. Chen, RUN beyond the metaphor: an efficient optimization algorithm based on Runge Kutta method, Expert Syst. Appl. 181 (2021) 115079.
[6] O. Akdag, A. Ates, C. Yeroglu, Modification of Harris hawks optimization algorithm with random distribution functions for optimum power flow problem, Neural Comput. Appl. 33 (6) (2021) 1959–1985.
[7] Z. Allen-Zhu, Y. Li, Neon2: Finding local minima via first-order oracles, Adv. Neural Inf. Process. Syst. 31 (2018).
[8] Q. Askari, M. Saeed, I. Younas, Heap-based optimizer inspired by corporate rank hierarchy for global optimization, Expert Syst. Appl. 161 (2020) 113702.
[9] S.J. Badashah, S.S. Basha, S.R. Ahamed, S.P.V.S. Rao, Fractional-Harris hawks optimization-based generative adversarial network for osteosarcoma detection using Renyi entropy-hybrid fusion, Int. J. Intell. Syst. 36 (10) (2021) 6007–6031.
[10] S. Boyd, S.P. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.
[11] H. Chen, A.A. Heidari, H. Chen, M. Wang, Z. Pan, A.H. Gandomi, Multi-population differential evolution-assisted Harris hawks optimization: Framework and case studies, Future Gener. Comput. Syst. 111 (2020) 175–198.
[12] C.A.C. Coello, Use of a self-adaptive penalty approach for engineering optimization problems, Comput. Ind. 41 (2) (2000) 113–127.
[13] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[14] T.-S. Du, X.-T. Ke, J.-G. Liao, Y.-J. Shen, DSLC-FOA: improved fruit fly optimization algorithm for application to structural engineering design optimization problems, Appl. Math. Model. 55 (2018) 314–339.
[15] A. Etminaniesfahani, H. Gu, A. Salehipour, ABFIA: A hybrid algorithm based on artificial bee colony and Fibonacci indicator algorithm, J. Comput. Sci. 61 (2022) 101651.
[16] C. Fan, Y. Zhou, Z. Tang, Neighborhood centroid opposite-based learning Harris Hawks optimization for training neural networks, Evol. Intell. 14 (4) (2021) 1847–1867.
[17] D.B. Fogel, Evolutionary Computation: Toward a New Philosophy of Machine Intelligence, John Wiley & Sons, 2006.
[18] A.H. Gandomi, X.-S. Yang, S. Talatahari, A.H. Alavi, Metaheuristic algorithms in modeling and optimization, in: Metaheuristic Applications in Structures and Infrastructures, 2013, Elsevier, Amsterdam, The Netherlands, 2013, pp. 1–24.
[19] S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power, Inform. Sci. 180 (10) (2010) 2044–2064.
[20] I. Gölcük, F.B. Ozsoydan, Quantum particles-enhanced multiple Harris Hawks swarms for dynamic optimization problems, Expert Syst. Appl. 167 (2021) 114202.
[21] A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, Future Gener. Comput. Syst. 97 (2019) 849–872.
[22] D.J. Higham, An algorithmic introduction to numerical simulation of stochastic differential equations, SIAM Rev. 43 (3) (2001) 525–546.
[23] E.H. Houssein, N. Neggaz, M.E. Hosney, W.M. Mohamed, M. Hassaballah, Enhanced Harris hawks optimization with genetic operators for selection chemical descriptors and compounds activities, Neural Comput. Appl. 33 (20) (2021) 13601–13618.
[24] R.S. Jadon, U. Dutta, Modified ant colony optimization algorithm with uniform mutation using self-adaptive approach, Int. J. Comput. Appl. 74 (13) (2013).
[25] M. Jeanblanc, M. Yor, M. Chesney, Mathematical Methods for Financial Markets, Springer Science & Business Media, 2009.
[26] S. Jiao, C. Wang, R. Gao, Y. Li, Q. Zhang, Harris hawks optimization with multi-strategy search and application, Symmetry 13 (12) (2021) 2364.
[27] I. Karatzas, S.E. Shreve, Brownian motion, in: Brownian Motion and Stochastic Calculus, Springer, 1998, pp. 47–127.
[28] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95-International Conference on Neural Networks, Vol. 4, IEEE, 1995, pp. 1942–1948.
[29] Y. Kumar, V.K. Singh, Computational approach based on wavelets for financial mathematical model governed by distributed order fractional differential equation, Math. Comput. Simulation (2021).
[30] S. Li, H. Chen, M. Wang, A.A. Heidari, S. Mirjalili, Slime mould algorithm: A new method for stochastic optimization, Future Gener. Comput. Syst. 111 (2020) 300–323.
[31] C. Li, J. Li, H. Chen, M. Jin, H. Ren, Enhanced Harris hawks optimization with multi-strategy for global optimization tasks, Expert Syst. Appl. 185 (2021) 115499.
[32] J. Luo, H. Chen, Y. Xu, H. Huang, X. Zhao, et al., An improved grasshopper optimization algorithm with application to financial stress prediction, Appl. Math. Model. 64 (2018) 654–668.
[33] Q. Luo, X. Yang, Y. Zhou, Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, Math. Comput. Simulation 159 (2019) 57–92.
[34] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.
[35] K. Price, N. Awad, M. Ali, P. Suganthan, The 100-digit challenge: Problem definitions and evaluation criteria for the 100-digit challenge special session and competition on single objective numerical optimization, Nanyang Technol. Univ. (2018).
[36] K. Price, R.M. Storn, J.A. Lampinen, Differential Evolution: A Practical Approach to Global Optimization, Springer Science & Business Media, 2006.
[37] J. Qiao, Y. Niu, T. Kifer, Intelligent optimization algorithm for global convergence of non-convex functions based on improved fuzzy algorithm, J. Intell. Fuzzy Systems 35 (4) (2018) 4465–4473.
[38] C. Qu, W. He, X. Peng, X. Peng, Harris hawks optimization with information exchange, Appl. Math. Model. 84 (2020) 52–75.
[39] H. Ren, J. Li, H. Chen, C. Li, Adaptive levy-assisted salp swarm algorithm: Analysis and optimization case studies, Math. Comput. Simulation 181 (2021) 380–409.

[40] S. Song, P. Wang, A.A. Heidari, X. Zhao, H. Chen, Adaptive Harris hawks optimization with persistent trigonometric differences for photovoltaic model parameter extraction, Eng. Appl. Artif. Intell. 109 (2022) 104608.
[41] R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (4) (1997) 341–359.
[42] Y. Su, Y. Dai, Y. Liu, A hybrid parallel Harris hawks optimization algorithm for reusable launch vehicle reentry trajectory optimization with no-fly zones, Soft Comput. 25 (23) (2021) 14597–14617.
[43] P. Sun, H. Liu, Y. Zhang, L. Tu, Q. Meng, An intensify atom search optimization for engineering design problems, Appl. Math. Model. 89 (2021) 837–859.
[44] R. Toscano, P. Lyonnet, A new heuristic approach for non-convex optimization problems, Inform. Sci. 180 (10) (2010) 1955–1966.
[45] K.H. Truong, P. Nallagownden, I. Elamvazuthi, D.N. Vo, An improved meta-heuristic method to maximize the penetration of distributed generation in radial distribution networks, Neural Comput. Appl. 32 (14) (2020) 10159–10181.
[46] J. Tu, H. Chen, M. Wang, A.H. Gandomi, The colony predation algorithm, J. Bionic Eng. 18 (3) (2021) 674–710.
[47] P. Vasant, T. Ganesan, I. Elamvazuthi, An improved PSO approach for solving non-convex optimization problems, in: 2011 Ninth International Conference on ICT and Knowledge Engineering, IEEE, 2012, pp. 80–87.
[48] G.-G. Wang, Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems, Memet. Comput. 10 (2) (2018) 151–164.
[49] G.-G. Wang, S. Deb, Z. Cui, Monarch butterfly optimization, Neural Comput. Appl. 31 (7) (2019) 1995–2014.
[50] X. Wang, X.Z. Gao, S.J. Ovaska, Artificial immune optimization methods and applications-a survey, in: 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583), Vol. 4, IEEE, 2004, pp. 3415–3420.
[51] A. Wunnava, M.K. Naik, R. Panda, B. Jena, A. Abraham, An adaptive Harris hawks optimization technique for two dimensional grey gradient based multilevel image thresholding, Appl. Soft Comput. 95 (2020) 106526.
[52] Z. Yan, J. Zhang, J. Zeng, J. Tang, Nature-inspired approach: An enhanced whale optimization algorithm for global optimization, Math. Comput. Simulation 185 (2021) 17–46.
[53] X.-S. Yang, Firefly algorithm, stochastic test functions and design optimisation, Int. J. Bio-Inspired Comput. 2 (2) (2010) 78–84.
[54] X.-S. Yang, A new metaheuristic bat-inspired algorithm, in: Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), Springer, 2010, pp. 65–74.
[55] Y. Yang, H. Chen, A.A. Heidari, A.H. Gandomi, Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts, Expert Syst. Appl. 177 (2021) 114864.
[56] Y. Yang, H. Chen, S. Li, A.A. Heidari, M. Wang, Orthogonal learning harmonizing mutation-based fruit fly-inspired optimizers, Appl. Math. Model. 86 (2020) 368–383.
[57] H. Zhang, M. Yuan, Y. Liang, Q. Liao, A novel particle swarm optimization based on prey–predator relationship, Appl. Soft Comput. 68 (2018) 202–218.
[58] X. Zhao, Y. Li, P. Boonen, Intelligent optimization algorithm of non-convex function based on genetic algorithm, J. Intell. Fuzzy Systems 35 (4) (2018) 4289–4297.
[59] Y. Zhou, Y. Ling, Q. Luo, Lévy flight trajectory-based whale optimization algorithm for engineering optimization, Eng. Comput. (2018).