Viser Technology Pte. Ltd.
Mechanical Engineering Science
DOI: 10.33142/mes.v6i1.13224

Research Article Open Access

An Improved Harris Hawk Optimization Algorithm

GuangYa Chong, Yongliang YUAN*

He'nan Polytechnic University, Jiaozuo, He'nan, 454150, China

****Corresponding Author:*** *Yongliang YUAN, E-mail: yuanyongliang@hpu.edu.cn*

Abstract

Aiming at the problems that the original Harris Hawk optimization algorithm is easy to fall into local optimum and slow in finding the optimum, this paper proposes an improved Harris Hawk optimization algorithm (GHHO). Firstly, we used a Gaussian chaotic mapping strategy to initialize the positions of individuals in the population, which enriches the initial individual species characteristics. Secondly, by optimizing the energy parameter and introducing the cosine strategy, the algorithm's ability to jump out of the local optimum is enhanced, which improves the performance of the algorithm. Finally, comparison experiments with other intelligent algorithms were conducted on 13 classical test function sets. The results show that GHHO has better performance in all aspects compared to other optimization algorithms. The improved algorithm is more suitable for generalization to real optimization problems.

Keywords: *Harris Hawk optimization algorithm; chaotic mapping; cosine strategy; function optimization*

1 Introduction

With the advancement of technology and the complexity of problems, optimization tasks often exhibit characteristics such as multi-objective, large-scale, uncertainty, and complexity, which are difficult to resolve [1]. In the real world, most problems have multiple constraints and optimization objectives, while traditional optimization algorithms [2-3] mainly focus on a single objective, which makes it difficult to solve real-world problems [4-7]. It is due to these shortcomings of traditional optimization algorithms that metaheuristic optimization algorithms have emerged, which are better able to solve complex engineering problems [8]. Metaheuristic algorithms are mostly inspired by natural phenomena and are considered as the best optimization algorithms globally due to their superior performance. With the continuous exploration of optimization algorithms, many metaheuristic algorithms have been generated, such as Dragonfly Algorithm [9], Snake Optimizer [10], White Shark Optimize [11], Sine cosine algorithm [12], Atomic Orbital Search [13], etc. Harris Hawk optimization algorithm is a metaheuristic optimization algorithm proposed by Haidari et al. in 2019. [14]. The algorithm can utilize simpler and practical processing methods to simplify the problem and has relatively good performance. However, the Harris Hawk optimization algorithm suffers from low precision of the results and low values of convergence speed. Therefore, to solve these problems, we improved the algorithm and named it GHHO. Firstly, population initialization is a crucial step in the optimization algorithm, which plays a vital role in both the performance and convergence speed of the algorithm. The Gauss mapping method is used to obtain the initial HHO population, which solves the problem of non-uniform distribution of initial positions in the search space, and makes the population distribution more uniform and diverse. Secondly, the computational accuracy of the algorithm is improved by changing the original parameters through Gauss chaotic mapping. Finally, a cosine strategy is introduced into the algorithm, which prevents the algorithm from falling into a local optimum.

2 Harris Hawks Optimization (HHO)

The Harris' hawk is found mainly in the United States and is a fierce bird. They will cooperate with each other in the hunting process, and increase the success rate through coordinated hunting. During the hunting process, Harris's hawks often use "raids", where they attack through multiple individuals and multiple angles of encirclement. This "raiding" behavior highlights the foraging characteristics of this species, and makes it difficult for prey to escape from encirclement due to the rapidity and frequency of their attacks. This species also changes its hunting mode according to the escape characteristics of the prey, so that the hunting process can be adapted to the situation and the prey can be obtained efficiently. Heidari et al. fully studied the hunting behavior of Harris's hawk and constructed the mathematical model of HHO by combining with Gray

Received on May 8, 2024; Accepted on May 29, 2024

Wolf Optimization (GWO) algorithm. We assume the behavior of Harris Hawk in acquiring prey as two phases: exploration and exploitation. The amount of escape energy E will determine whether the algorithm is in the exploratory or exploitation stage; when in the exploitation stage, the attack strategy can be categorized into soft and hard besiege according to the escape energy when the random number r is not less than 0.5. On the contrary, when r is less than 0.5, it can also be categorized into soft besiege with progressive rapid dives and hard besiege with progressive rapid dives based on the escape energy.

2.1 Exploration phase

When $|E| \geq 1$, GHHO is in an exploratory phase. The location update can be categorized into two strategies based on the Harris's hawk's perching situation q, as shown in Eq (1-2):

$$X(t+1) = X_{rand}(t) - r_1 \mid X_{rand}(t) - 2r_2 X(t) \mid \quad q \geq 0.5 \tag{1}$$

$$X(t+1) = (X_{rabbit}(t) - X_m(t)) - r_3(LB + r_4(UB - LB)) \quad q < 0.5 \tag{2}$$

Where $X(t+1)$ is the coordinates of the Harris' hawk at t+1 iteration; $X_{rand}(t)$ is the coordinates of the species in the randomized case at the tth iteration; $X_{rabbit}(t)$, $X_m(t)$ are the prey-specific coordinates at the tth iteration, and the coordinates of the midpoint of the Harris' hawk; LB, UB are numerical range thresholds; r_1 , r_2 , r_3 and r_4 are random numbers inside (0,1).

During the iteration process, the escape energy E decreases gradually as shown in Eq (3):

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

Where T , t and E_0 represent the maximum number of iterations, the current number of iterations and the initial energy value of E, respectively.

2.2 Exploitation phase

When $|E| < 1$, hunting can be further categorized into four different strategies.

a. Soft besiege

When $r \geqslant 0.5$ and $|E| \geq 0.5$, the algorithm implements this strategy as shown in Eq (4), with J being the escape of the besieged species. r_5 is a random number between 0 and 1

$$X(t+1) = X_{rabbit}(t) - X(t) - E\left|JX_{rabbit}(t) - X(t)\right| \tag{4}$$

$$J = 2(1 - r_5) \tag{5}$$

b. Hard besiege

When $r \geq 0.5$ and $|E| < 0.5$, Harris' hawk adopts a hard besiege, see Eq (6). At this point, the Harris's hawk will be a one-strike winner as the prey has been escaping resulting in a decrease in escape level.

$$X(t+1) = X_{rabbit}(t) - E\left|X_{rabbit}(t) - X(t)\right| \tag{6}$$

c. Soft besiege with progressive rapid dives

When still $|E| \geq 0.5$ but $r < 0.5$, since the prey has enough stamina to escape, the Harris' hawk will start a high-speed dive, which is still a soft besiege. If the raid is unsuccessful, it will turn on random wandering Z , and if the wandering is unsuccessful it will return to the initial position, the strategy is shown in equation (7-8):

$$Y = X_{rabbit}(t) - E\left|JX_{rabbit}(t) - X(t)\right| \tag{7}$$

$$Z = Y + S \times LF(D) \tag{8}$$

Where S and D represent the random number and dimension, respectively. LF is the levy flight function, as shown in Eq (9):

$$LF(x) = 0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}}, \sigma = \left(\frac{\Gamma(1+\beta) \times sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{(\frac{\beta-1}{2})}} \right)^{\frac{1}{\beta}} \tag{9}$$

Where, u, v are random numbers in the range of (0,1) and the value of β is 1.5.

d. Hard besiege with progressive rapid dives

When $|E| < 0.5$ and $r < 0.5$, the Harris' hawk based on the location of the center of the group in relation to where the prey is located, with reference to Eq (10) for the specific strategy.

$$Y = X_{rabbit}(t) - E\left|JX_{rabbit}(t) - X_m(t)\right| \tag{10}$$

3 Improved Harris Hawk Optimization (GHHO)

3.1 Initialization of Gauss chaos mapping

HHO is randomly generated by the computer, which tends to make the individual fitness of the initial population deviate from the optimal fitness, thus making the algorithm converge slowly and generating high running time, leading to the degradation of the solution quality. The author adopts Gauss chaotic mapping to initialize the population and ensure the diversity of the population. As shown in Eq (11):

$$x_{n+1} = f(x_n) = \begin{cases} 1 & x_n = 0 \\ \dfrac{1}{\mathrm{mod}(x_n, 1)} & \text{otherwise} \end{cases} \tag{11}$$

3.2 Optimization of energy parameters

In response to the suspension of the HHO due to early convergence, the author uses Gauss chaotic mapping to optimize the energy parameter E. Finally, it is shown that this method can improve the function difficulties and improve the accuracy and robustness. The improvement is shown in Eq (11):

$$E = 2E_0(1 - \frac{t}{T}) \times f(x_n) \tag{12}$$

3.3 Cosine strategy

In order to solve the problem of low accuracy of results and easy to fall into local optimum in the use of HHO method, the author introduced the cosine strategy in Eq (2), which improves the performance of the optimization algorithm solution. As shown in Eq (13):

$$X(t+1)=(X_{rabbit}(t)-X_m(t))-r_3(LB+r_4(UB-LB))\times\cos(r_5+1) \quad q<0.5 \tag{13}$$

Where, r_5 is a random number lies in the range of (0, 1).

3.4 Algorithm flow chart

Combining the above improvement strategies, the GHHO algorithm flow is shown in Figure 1.

Step 1: Gauss population initialization, set population size N, dimension D, initialize each parameter.

Step 2: Gauss optimizes the energy parameter E. The optimized E is applied to Eqs. (4), (6), (7) and (10) to update the population position.

Step 3: Introduce the cosine strategy into Eq. (2) in the exploratory phase, a phase in which the population will use Eq. (13) to update its position.

Step 4: Determine whether the termination condition is reached, if so, output the optimal value; otherwise, return to step 2.

The GHHO algorithm records the position of the population from the first iteration and each iteration is compared with the results of the last iteration until the last iteration is completed. The best candidate solution saved during the iterations is used as the final solution to the problem after the algorithm is completely finished.

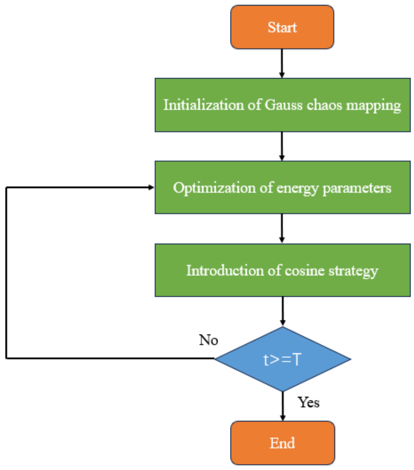

Figure 1 Flow chart of GHHO

Table 1 Statistical results of test functions

Function		GWO	BOA	MFO	GA	PSO	GHHO
F1	Avg	3.64E-174	1.93E-17	1.79E+03	1.63E-02	7.55E-04	0.00E+00
	Std	0.00E+00	9.38E-19	3.87E+03	4.59E-03	7.06E-04	0.00E+00
F2	Avg	1.22E-99	1.51E-14	2.96E+01	3.19E-01	1.43E-02	2.64E-270
	Std	2.84E-99	1.74E-15	2.20E+01	4.70E-02	7.87E-03	0.00E+00
F3	Avg	1.97E-55	2.02E-17	1.36E+04	2.44E+02	1.80E+01	0.00E+00
	Std	1.35E-54	9.95E-19	1.20E+04	4.89E+02	4.89E+00	0.00E+00
F4	Avg	8.73E-44	1.64E-14	1.42E+01	1.57E-01	8.77E-01	1.17E-268
	Std	4.17E-43	7.52E-16	8.62E+00	1.89E-02	1.48E-01	0.00E+00
F5	Avg	2.59E+01	2.90E+01	3.11E+04	3.40E+01	9.60E+01	2.32E-02
	Std	8.57E-01	2.59E-02	4.79E+04	2.11E+01	7.34E+01	2.44E-02
F6	Avg	1.28E-01	4.86E+00	2.10E+03	7.85E+00	6.80E-04	2.07E-05
	Std	1.91E-01	5.84E-01	3.43E+03	1.01E-01	7.38E-04	3.55E-05
F7	Avg	1.23E-04	2.26E-04	1.68E+00	8.56E-01	1.93E-01	1.84E-05
	Std	7.87E-05	8.01E-05	4.30E+00	2.29E-01	8.26E-02	5.46E-06
F8	Avg	-6.45E+03	-4.78E+03	-9.00E+03	-2.66E+03	-6.81E+03	-1.07E+04
	Std	6.60E+02	3.65E+02	8.50E+02	5.04E+02	8.56E+02	1.93E+03
F9	Avg	0.00E+00	1.89E+00	1.18E+02	1.82E+00	4.86E+01	0.00E+00
	Std	0.00E+00	1.89E+01	3.82E+01	5.46E-01	1.35E+01	0.00E+00
F10	Avg	8.03E-15	1.04E-14	8.03E-15	8.41E-02	1.81E-02	4.44E-16
	Std	9.44E-16	3.60E-15	9.44E-16	1.60E-02	1.10E-02	0.00E+00
F11	Avg	4.52E-04	0.00E+00	4.52E-04	5.91E-04	1.39E-02	0.00E+00
	Std	2.30E-03	0.00E+00	3.30E-03	2.75E-04	1.21E-02	0.00E+00
F12	Avg	1.28E-02	2.86E-01	2.28E-02	2.69E+00	4.08E-06	6.25E-07
	Std	9.31E-03	8.30E-02	9.31E-03	4.39E-02	4.43E-06	8.09E-07
F13	Avg	1.55E-01	1.69E+00	2.55E-01	2.93E-03	9.67E-04	7.70E-05
	Std	1.25E-01	4.39E-01	3.24E-01	7.13E-04	2.62E-03	1.46E-04

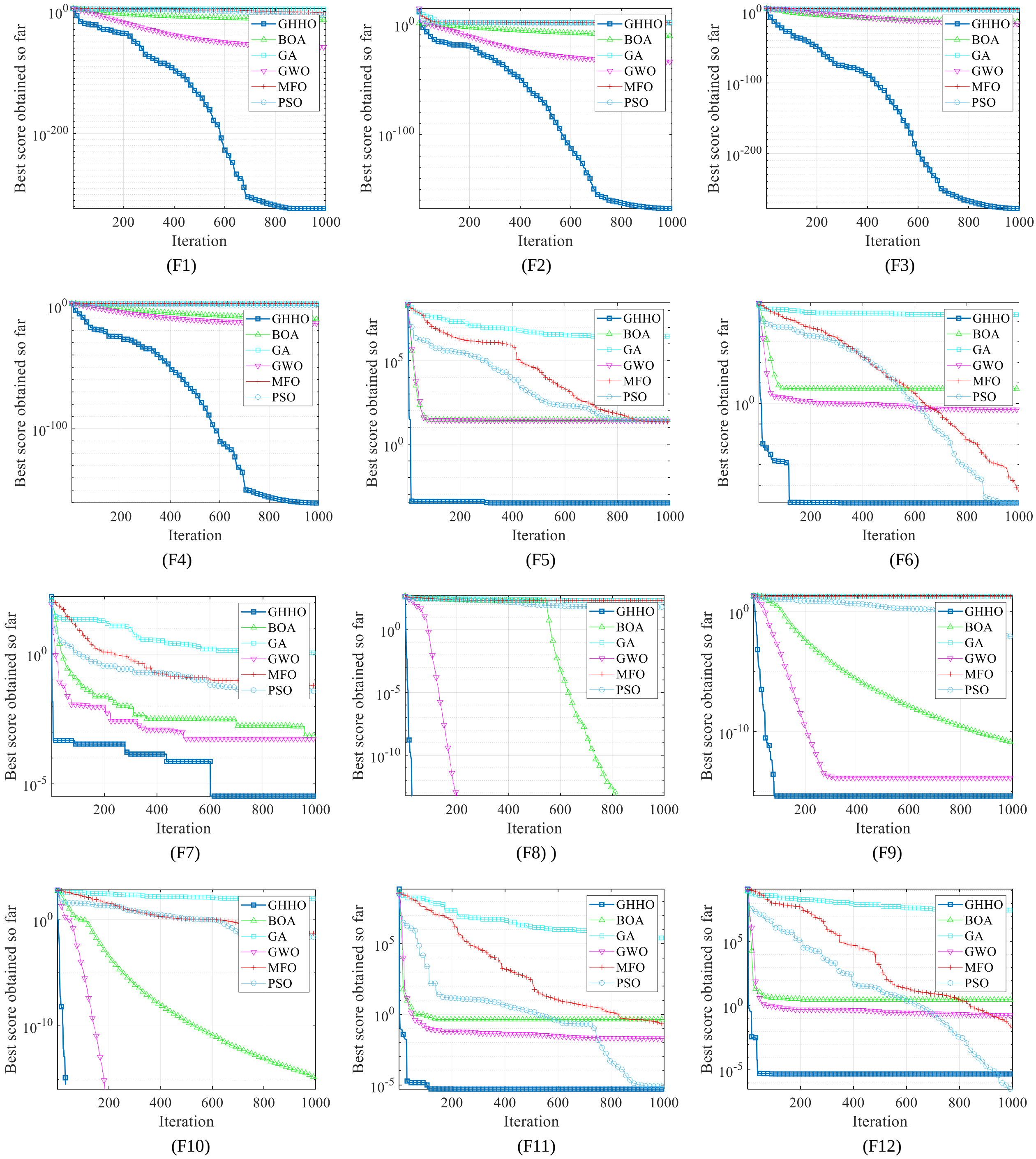

Figure 2 Convergence Curve

4 Experimental Design

This experiment is designed to examine the performance of each algorithm with 13 typical global optimization test functions from the literature [15]. These classical functions are categorized into single-peak and multi-peak functions.

We compare GHHO with GWO [16], BOA[17], MFO[18], GA[19] and PSO [20]respectively. During the experiments, the population size and dimension of each algorithm is set to 30 and the maximum number of iterations is set to 1000.

Table 1 presents the competitive results of the GHHO algorithm on F1-F13. It is clear that GHHO performs well in terms of overall performance, specifically, GHHO obtains the best solution for all tested functions except F8. Figure 2 demonstrates the convergence process of all algorithms except F8, and GHHO exhibits higher convergence accuracy compared to other optimization algorithms, which also validates the experimental data in Table 1.

These results show the excellent performance of the GHHO algorithm in dealing with single-peak benchmark functions (F1-F7) and multi-peak functions (F8-F13), which further validates the accuracy and reliability of the GHHO algorithm as well as its excellent performance on different types of optimization problems.

5 Conclusion

In this work, an optimization algorithm GHHO with better performance is proposed based on HHO. Firstly, Gaussian chaotic mapping strategy is used to initialize the population. Secondly, by optimizing the energy parameters and introducing the cosine strategy, GHHO possesses a better performance. Finally, the effectiveness of GHHO is verified by comparing it with different intelligent optimization algorithms on test functions.

References

[1] Wan M, Ye C, Peng D. Multi-period dynamic multi-objective emergency material distribution model under uncertain demand[J]. Eng Appl Artif Intell, 2023(117):105530.

[2] Inceyol Y, Cay T. Comparison of traditional method and genetic algorithm optimization in the land reallocation stage of land consolidation[J]. Land Use Policy, 2022(115):105989.

[3] Wang W-c, Xu L, Chau K-w, et al. An orthogonal opposition-based-learning Yin–Yang-pair optimization algorithm for engineering optimization[J]. Eng Comput, 2022(38):1149–1183.

[4] Atban F, Ekinci E, Garip Z. Traditional machine learning algorithms for breast cancer image classifcation with optimized deep features[J]. Biomed Signal Process Control, 2023(81):104534.

[5] Hu G, Chen L, Wei G. Enhanced golden jackal optimizer-based shape optimization of complex CSGC-Ball surfaces[J]. Artif Intell Rev, 2023(56):2407–2475.

[6] Wang L, Gao K, Lin Z, et al. Problem feature based meta-heuristics with Q-learning for solving urban trafc light scheduling problems[J]. Appl Soft Comput, 2023(147):110714.

[7] Wang W-c, Xu L, Chau K-w, et al. Cε-LDE: a lightweight variant of differential evolution algorithm with combined ε constrained method and Lévy fight for constrained optimization problems[J]. Expert Syst Appl, 2023(211):118644.

[8] Ayyarao T S L V, Ramakrishna N S S, Elavarasan R M, et al. War strategy optimization algorithm: a new effective metaheuristic algorithm for global optimization[J]. IEEE Access, 2022(10):25073-25105.

[9] Mirjalili S. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems[J]. Neural computing and applications, 2016(27):1053-1073.

[10] Hashim F A, Hussien A G. Snake Optimizer: A novel meta-heuristic optimization algorithm[J]. Knowledge-Based Systems, 2022(242):108320.

[11] Braik M, Hammouri A, Atwan J, et al. White Shark Optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems[J]. Knowledge-Based Systems, 2022(243):108457.

[12] Mirjalili S. SCA: a sine cosine algorithm for solving optimization problems[J]. Knowledge-based systems, 2016(96):120-133.

[13] Azizi M. Atomic orbital search: A novel metaheuristic algorithm[J]. Applied Mathematical Modelling, 2021(93):657-683.

[14] HEIDARI A A, MIRJALILI S, FARIS H, et al. Harris hawks optimization: Algorithm and applications[J]. Future generations computer systems (FGCS), 2019(97):849-872.

[15] YAO Xin, LIU Yong, LIN Guangming. Evolutionary programming made faster[J]. IEEE transactions on evolutionary computation, 1999,3(2):82-102.

[16] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer[J]. Advances in engineering software, 2014(69):46-61.

[17] Arora S, Singh S. Butterfly optimization algorithm: a novel approach for global optimization[J]. Soft computing, 2019(23):715-734.

[18] Mirjalili S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm[J]. Knowledge-based systems, 2015(89):228-249.

[19] Deb K. Optimal design of a welded beam via genetic algorithms[J]. AIAA journal, 1991,29(11):2013-2015.

[20] Wang D, Tan D, Liu L. Particle swarm optimization algorithm: an overview[J]. Soft computing, 2018,22(2):387-408..