

An improved crow search algorithm with multi strategy disturbance

Huadong Huang, and Heng Liao*

Guangxi International Business Vocational College, Nanning 530006, China

Abstract. As a new meta heuristic intelligent algorithm, crow search algorithm simulates the behavior of crows following each other and stealing food. Due to the simplicity and robustness of crow search algorithm, it has been successfully applied in many fields. However, like other swarm intelligence optimization algorithms, crow search algorithm also has the disadvantages of slow convergence speed and easy to fall into local optimization. In order to improve the convergence accuracy and later search ability of the algorithm, a new hybrid crow search algorithm called multi strategy disturbance improved crow search algorithm (MSD-CSA) is proposed based on the traditional crow search algorithm. In MSD-CSA, the sharing mechanism is added to improve the location update mode of random tracking in the original algorithm, reduce the search blindness and improve the convergence speed. In addition, the global optimal location is perturbed with different sizes in different iterative stages, which effectively improves the probability of jumping out of the local optimal and ensures the balance between the global search ability and the local search ability of the algorithm. In order to evaluate the effectiveness of MSD-CSA algorithm, it is applied to 20 basic test functions for optimization experiments, and compared with other intelligent optimization algorithms. Experimental results show that the average convergence and robustness of the proposed algorithm are better than other algorithms, and the overall performance is good.

Keywords: Crow search algorithm, Sharing mechanism, Multi Strategy disturbance, Function optimization.

1 Introduction

In recent years, meta heuristic algorithm, as an effective evolutionary computing technology, has attracted the attention of many scholars [1]. Meta heuristic algorithm is a kind of algorithm inspired by the real environment. which core idea is to balance the random behavior and local search in the search process. Common meta heuristic algorithms include Particle Swarm Optimization (PSO) [2], Bat Algorithm(BA)[3], Butterfly Optimization Algorithm (BOA)[4], Flower Pollination Algorithm(FPA) [5], Teaching-learning-based Optimization(TLBO) [9], Pigeon Inspired Optimization(PIO) [6], Whale

* Corresponding author: liaheng@stu.gxun.edu.cn

Optimization Algorithm(WOA)[7], Gray Wolf Optimizer(GWO)[8], etc. These algorithms have the characteristics of simplicity, few parameters and short running time. Therefore, meta heuristic algorithms show excellent operability and optimization ability in solving many nonlinear and multimodal practical optimization problems.

Crow search algorithm (CSA) is a new meta heuristic algorithm proposed by Ailreza Askarzadeh in 2016 [10]. Based on the long-term research on crow habits, the algorithm simulates the social behavior of tracking and stealing food among the groups. This characteristic makes the diversity of the algorithm population not greatly reduce with the iteration, and increases the probability of jumping out of the local optimization. At present, CSA has been successfully applied to many engineering optimization and function optimization problems, such as chemical engineering and QSAR[11], image processing[12], feature selection[13], neural network and support vector machine[14], aircraft maintenance inspection [15], wireless sensor network[16] and other major engineering fields. However, like most optimization algorithms, crow search algorithm itself also has the defects of slow convergence speed and easy to fall into local optimal. In order to improve the convergence speed of CSA, many scholars have proposed improved algorithms. Wu Hao [17] proposed a crow search algorithm incorporated with Levy flight (LFCSA), and applied it to update the finite element model. The chaotic sequence was used to initialize the population, so that the particles were evenly distributed in the solution space and the diversity of the population was increased. Therefore, Liu Xuejing et al. proposed the chaotic binary crow search algorithm (CBCSA)[18] for the discrete space, use it to solve the problem of {0-1} knapsack. By introducing adaptive step size, Mohammadi et al. proposed an self-adaptive step size crow search algorithm (MCSA)[19] and applied it to non-convex economic load scheduling. For the second kind of improvement, Xiao Ziya [20] mixed optimization of crow search algorithm and sine and cosine algorithm to this algorithm to pressure vessel design. Arora [21] et al. combined the crow search algorithm with the gray Wolf algorithm and used it to solve the problem of feature selection. Pratiwi [22] et al. combined the crow search algorithm with the cat swarm algorithm and applied it to the vehicle routing problem with time Windows. However, the above algorithms are mainly aimed at improving the global search performance, do not focus on balancing the global search performance and local search performance of crow algorithm, and the problem of slow convergence speed of the algorithm has not been well solved.

In this paper, the basic crow algorithm is improved from three aspects: making full use of the optimal individual, increasing population diversity and adaptive step size, so that it can better balance exploration and development, so as to propose an improved method called multi strategy disturbance improved crow search algorithm (MSD-CSA). In MSD-CSA, the idea of learning from the optimal particle and adding disturbances in different stages of iteration will be introduced to improve the convergence speed and accuracy of the algorithm at the same time. In order to deeply study the performance of MSD-CSA algorithm, this paper uses 20 commonly used benchmark functions to test, and compares it with the classical and newly proposed meta heuristic algorithms (differential evolution algorithm, particle swarm optimization algorithm, bat algorithm, gray wolf algorithm, firefly algorithm and crow search algorithm), so as to verify the effectiveness of the improved algorithm.

The structure of this paper is as follows. In section 2, the basic crow search algorithm will be briefly introduced. In section 3, the improved strategy will be described to produce an improved crow search algorithm. In section 4, the proposed improved algorithm is tested with 20 benchmark functions and compared with 6 algorithms. Finally, the conclusion is to given in Section 5.

2 Crow search algorithm

It is found that crows have a much larger proportion of brain relative to body than other birds. They have a higher IQ than other birds, so crows are considered to be very smart birds. By understanding the habits of crows, researchers found that crows will observe where other birds hide their food. Once the owner leaves, crows will steal the food. If a crow steals, it will take additional precautions, such as moving its hiding place, to avoid becoming a later victim. Crows use their experience of stealing food from other birds to predict the behavior of other thieves, and can find the safest hiding place to keep their stores from being stolen. CSA is a population-based technology to find the best solution to the optimization problem by simulating the above biological behavior of crows. CSA abides by the following four principles:

- (1) crows live in groups.
- (2) Crows can remember where they hide.
- (3) Crows follow each other and steal.
- (4) Crows protect their stores from theft with the greatest probability.

Based on the four principles, the basic process of CSA is described as follows:

Step 1: Initializing the parameters of CSA. Such as: population size (n), maximum iteration number ($Maxiter$), flight step size (fl), awareness probability (AP), etc.

Step 2: Initializing the individual crows and memory matrix. n crows are generated in the search space of d -dimension, and each crow $x_i = (X_{i,1}, X_{i,2}, X_{i,3}, \dots, X_{i,d})$ represents a feasible solution of a problem. Since the initial population has no experience, it is assumed that the initial memory matrix is the initial position.

Step 3: Evaluate the quality of each crow according to the fitness function.

Step 4: A new location is generated for each crow in the d -dimensional search space. Assuming that crow i randomly follows a crow (for example, crow j) in order to find the place of the hidden food of crow j , the position update of crow i can be divided into the following two situations.

Case 1: crow j does not find that crow i is following it. In this case, the position update formula of Crow i is:

$$x^{i,iter+1} = x^{i,iter} + r_i + fl^{i,iter} \times (m^{j,iter} - x^{i,iter}) \quad (1)$$

Case 2: Crow j finds that crow i is following it, and crow j will take crow i to a random position.

To sum up, the position update formula of Crow i is:

$$x^{i,iter+1} \begin{cases} x^{i,iter} + r_i + fl^{i,iter} \times (m^{j,iter} - x^{i,iter}), r_j \geq AP \\ a \text{ random position, otherwise} \end{cases} \quad (2)$$

In the formula, r_i, r_j are random numbers that obey the uniform distribution of $[0,1]$. AP represents the perception probability. When the AP is smaller, the probability of occurrence of case 1 is greater, and the algorithm tends to search locally. When the AP is larger, the probability of finding in case 2 is greater, and the algorithm tends to search globally. $fl^{i,iter}$ is the flight step length of crow i . When $fl^{i,iter} < 1$, the next position of crow i is between x_i^t and m_j^t , as shown in Figure 1. When $fl^{i,iter} > 1$, the next position of crow i is outside the line between x_i^t and m_j^t , as shown in Figure 2. Therefore, fl will affect the search ability of the algorithm. If the value is too large, it tends to search globally, and the algorithm has poor convergence. If the value is too small, it is easy to fall into the local optimum.

Step 5: Checking whether the new position of each crow is feasible. If possible, change the crow's position. Otherwise, it is not updated.

Step 6: Calculating the fitness value of the new position of each crow.

Step 7: Updating the memory matrix of each crow

$$m_i^{t+1} = \begin{cases} x_i^{t+1}, & f(x_i^{t+1}) \text{ is better than } f(m_i^t) \\ x_i^t & \text{other} \end{cases} \quad (3)$$

Step 8: Repeating steps 4-7 until the termination condition is reached.

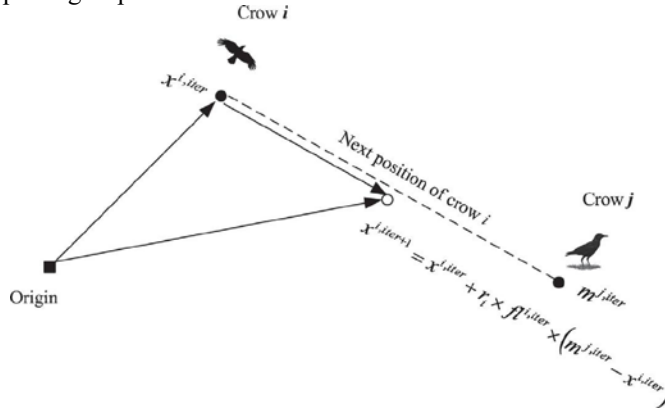


Fig. 1. Schematic diagram for updating the position of crow i when $fl < 1$.

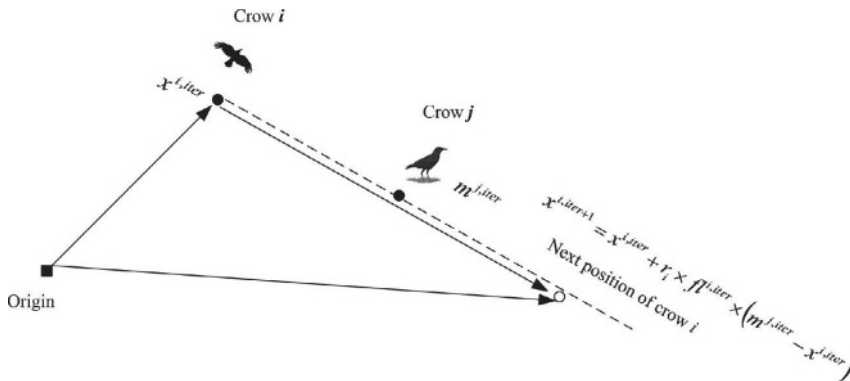


Fig. 2. Schematic diagram for updating the position of crow i when $fl > 1$.

3 Multi strategy disturbance improved crow search algorithm (MSD-CSA)

3.1 Location update method of joining the sharing mechanism

In the crow search algorithm, the crow i randomly tracks a crow j for location update. This update method can maintain the good diversity of the population in the early stage of the iteration, but it will also make the algorithm more blind, slow down the convergence speed, and can't converge to the optimal value in a short time.

In 1995, Kennedy scholar proposed particle swarm optimization (PSO) [2], which cooperatively searches the solution of the problem through information sharing among

individuals in the group, that is, the flight speed of particles will not only refer to their historical optimal position, but also refer to the position of the optimal particles in the group for action adjustment, which makes the algorithm have the characteristics of fast convergence speed. Therefore, in order to enhance the leadership of the algorithm, particle swarm optimization is introduced into the location update part of crow search algorithm. The idea of learning from the optimal particle in the algorithm makes the crow population share the optimal hiding position in the environment, that is, it updates itself with reference to the optimal hiding position in the population while randomly following an individual. Under the leadership of the optimal hiding and feeding position, the algorithm greatly improves the convergence speed. The new location update formula is as follows:

$$sm = r_i \times s \times (m^{gbest,iter} - x^{i,iter}) \tag{4}$$

$$x^{i,iter+1} = x^{i,iter} + r_i \times fl \times (m^{j,iter} - x^{i,iter}) + sm \tag{5}$$

Where, sm represents the formula of sharing mechanism and $m^{gbest,iter}$ represents the global optimal hiding position; s represents the sharing factor, which determines the influence of the optimal hiding and feeding position on the current crow position. After the improvement, the crow i moves further away from the optimal position every time, which effectively improves the convergence speed of the algorithm.

3.2 Adding disturbance strategy to update the optimal hiding position

CSA is a non-greedy algorithm, that is, if the current solution is not better than the previous generation, it will not be retained. Therefore, crow search algorithm is easy to fall into local optimization in the later stage of iteration. Especially when applied to complex multimodal functions, the disadvantages of premature convergence and low accuracy are more obvious. Secondly, the improved position update formula in Section 3.1 also affects the accuracy of the optimal solution of the algorithm in multimodal function. In this paper, the multi strategy disturbance is used to update the optimal Tibetan food position $m^{gbest,iter}$, that is, the global optimal Tibetan food position is disturbed according to the normal random distribution with adjustable variance to obtain a new global optimal Tibetan food position $m^{gbest',iter}$. The speed update formula is as follows:

$$m^{gbest',iter} = N(m^{gbest,iter}, \delta) \tag{6}$$

where, δ represents the variance of normal random distribution. δ is updated as follows:

$$\delta = \begin{cases} \delta_1 & \alpha_1 T < t < \alpha_2 T \\ \delta_2 & t > \alpha_2 T \end{cases} \tag{7}$$

where, δ represents the radius parameter of normal disturbance, and $\delta_1 > \delta_2$. α_1, α_2 is the control parameter of change of radius and $\alpha_1 < \alpha_2$. T is the calculation times of the current function value, and t is the maximum calculation times of the function. Early in the iteration δ_1 , if the value is large, the group will search in the neighborhood with large global optimal food hiding location; Interference radius at the end of iteration δ_2 if the value is small, search in the neighborhood where the global optimal hiding and eating position is

small, so that the current solution hardly jumps out of the better region, so as to ensure that the algorithm group only learns from the global optimal solution, so as to ensure that the algorithm has good convergence. Combined with the improvement of the algorithm in Sections 3.1 and 3.2, that is, while strengthening the leadership of the global optimal Tibetan food position, relying on the disturbance of different sizes in different stages of the global optimal Tibetan food position can greatly increase the probability of the algorithm jumping out of the local optimization, so that the algorithm achieves the balance between global search and local search, which not only improves the search accuracy but also speeds up the convergence speed.

4 Experimental simulation and result analysis

In order to verify the effectiveness of proposed algorithm in solving various optimization problems, this paper selects 20 typical benchmark functions [23] to compare the optimization performance of DE [24], PSO [25], FA [26], BA [27], GWO [28], CSA [10]. Because the population size corresponding to the best effect of each algorithm in each study is inconsistent, in order to ensure the fairness of the experiment, the evaluation times are used as the termination condition of the algorithm, and all parameters of each algorithm are set according to the parameters proposed in the corresponding literature (including population size and key parameters). For the sake of fairness, we used a fixed population size for all algorithms: $n = 50$ individuals, all algorithms are executed in 30 independent runs. All algorithms are implemented in Matlab (version R2020a) and executed on HP computer (Windows 10, Inter Core i5-6300HQ, 2.3 GHz, 8GB RAM). The 20 well-known benchmark test functions for validation. The set of test functions is explained in Table 1. These benchmark functions are divided into three categories. High-dimensional unimodal test function ($f_1 \sim f_7$), High-dimensional multi-modal test function ($f_8 \sim f_{14}$), and fixed-dimensional multi-modal test function ($f_{15} \sim f_{20}$). The number of iterations in each run for each algorithm equals 1000 iterations. And the experimental data obtained after running, as shown in Tables 2~4, where Best, Worst, Mean, and Std represent the optimal value, worst value, average value, and Std value obtained by the algorithm independently running 30 times. The algorithm performance is ranked according to the accuracy of the optimal value. The best results are for matted in bold. In order to observe the convergence speed and convergence accuracy more intuitively.

As can be seen from the experimental results in Table 2, except that the result of testing f_6 is not as good as GWO. Among the remaining seven functions $f_1, f_2, f_3, f_4, f_5,$ and f_7 , the optimal value of MSD-CSA ranks first. At the same time, the worst value, average value and standard deviation are all smaller than other algorithms, indicating that the stability and convergence accuracy of MSD-CSA are very high. This also shows that MSD-CSA has better optimization ability for high-dimensional unimodal functions. As can be seen from the experimental data results in Table 3, Except for the test data result of f_9 , which is ranked behind the BA algorithm, the data of the remaining seven benchmark functions $f_{10} \sim f_{14}$ show that the indicators of MSD-CSA are better than the other four algorithms. It shows the feasibility of MSD-CSA in solving high-dimensional multi-modal functions. At the same time, compared to the other four algorithms, MSD-CSA's global search capabilities are stronger.

It can be seen from the experimental data results in Table 4 that the effect of MSD-CSA in optimizing the function is slightly lower than that of GWO. For the optimization of the remaining four functions f_{17}, f_{18}, f_{20} , MSD-CSA has the smallest optimal value and standard deviation. In general, the optimization problem of fixed-dimensional multi-modal functions can be well solved by MSD-CSA.

Table 1. Benchmark test functions.

Benchmark function	Dim	Range	optimal value
$f_1(x) = \sum_{i=1}^n x_i^2$	30	$x_i \in [-100,100]$	0
$f_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	$x_i \in [-10,10]$	0
$f_3(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	30	$x_i \in [-100,100]$	0
$f_4(x) = \max \{ x_i , 1 \leq i \leq n \}$	30	$x_i \in [-100,100]$	0
$f_5(x) = \sum_{i=1}^{D-1} \left[100(x_i + 1 - x_i^2)^2 + (x_i - 1)^2 \right]$	30	$x_i \in [-30,30]$	0
$f_6(x) = \sum_{i=1}^n (x_i + 0.5)^2$	30	$x_i \in [-100,100]$	0
$f_7(x) = \sum_{i=1}^n x_i^4 + \text{random}(0,1)$	30	$x_i \in [-1.28,1.28]$	0
$f_8(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	30	$x_i \in [-500,500]$	0
$f_9(x) = \sum_{i=1}^{n-1} \left[x_i^2 - 10 \cos(2\pi x_i) + 10 \right]$	30	$x_i \in [-5.12,5.12]$	0
$f_{10}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + 2$	30	$x_i \in [-32,32]$	0
$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^n (x_i^2) - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$x_i \in [-600,600]$	0
$f_{12}(x) = \frac{x}{n} \left\{ 10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_i + 1)] + (y_n - 1)^2 \right\} + \sum_{i=1}^n \mu(x_i, 10, 100, 4)$	30	$x_i \in [-50,50]$	0
$f_{13}(x) = 0.1 \sin 2(3\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + \sin^2(3\pi y_{i+1}) + (y_n - 1)^2 (1 + \sin^2(2\pi y_n))] + \sum_{i=1}^n \mu(x_i, 5, 100, 4)$	30	$x_i \in [-50,50]$	0
$f_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \right)^{-1}$	2	$x_i \in [-65,65]$	0.998
$f_{15}(x) = \sum_{i=1}^n \left(\sum_{j=1}^j x_j \sin(x_i) + 0.1 x_j \right)$	4	$x_i \in [-10,10]$	0.0003
$f_{16}(x) = \sum_{i=1}^n x_i^2 \left(\sum_{j=1}^n 0.5 i x_j \right)^2 + \left(\sum_{i=1}^n 0.5 i x_i \right)^4$	2	$x_i \in [-5,10]$	0.398
$f_{17}(x) = \sum_{i=1}^n \left(\sum_{j=1}^i \text{rand} \left x_j - \frac{1}{j} \right \right)$	2	$x_i \in [-5,5]$	3
$f_{18}(x) = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_2 - 3x_2)^2 (18 - 32x_1 + 15x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	$x_i \in [-5,5]$	-3.86
$f_{19}(x) = -\sum_{i=1}^4 C_i \exp \left[\sum_{j=1}^3 a_{ij} (x_j - p_{ij})^2 \right]$	4	$x_i \in [0,1]$	-3.32
$f_{20}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_i (b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	$x_i \in [-10,10]$	-10.1532
$f_{21}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	4	$x_i \in [-5,5]$	-10.4029
$f_{22}(x) = (x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 + 6)^2 + 10(1 - \frac{1}{8\pi}) \cos x_1 + 10$	4	$x_i \in [-5,5]$	-10.5364

Table 2. Results of high-dimensional unimodal benchmark functions.

Function	Index	DE	PSO	FA	BA	GWO	CSA	MSD-CSA
f_1	Best	3.14E-03	1.52E+00	9.64E+03	2.35E-73	1.07E-25	8.53E-04	1.28E-100
	Worst	3.56E-02	4.49E+01	3.92E+04	7.47E-69	6.52E-24	4.74E+00	1.26E-91
	Mean	1.69E-02	1.56E+00	2.12E+04	2.40E-70	8.87E-25	9.52E-01	4.49E-93
	Std	7.49E-03	1.16E+01	7.06E+03	4.95E-70	1.45E-24	1.28E+00	2.30E-92
	Rank	5	6	7	2	3	4	1
f_2	Best	3.58E-01	1.81E+00	6.80E+01	7.44E-42	7.97E-23	3.73E-02	2.12E-50
	Worst	2.46E+00	5.52E+00	1.20E+07	1.55E-40	1.80E-22	7.16E+00	1.09E-45
	Mean	1.30E+00	3.53E+00	5.25E+04	4.50E-41	1.03E-22	1.56E+00	7.86E-47
	Std	5.83E-01	9.99E-01	2.16E+06	3.31E-41	2.26E-22	1.48E+00	2.52E-46
	Rank	5	6	7	2	3	4	1
f_3	Best	8.00E-01	3.68E+00	3.18E+01	2.32E-19	9.28E-23	3.26E-03	3.44E-49
	Worst	5.92E+00	1.01E+01	6.63E+01	8.62E-17	3.15E-22	9.01E-01	1.20E-44
	Mean	2.60E+00	6.11E+00	5.52E+01	1.56E-17	1.39E-22	3.02E-01	5.63E-46
	Std	1.06E+00	1.58E+00	7.38E+00	1.82E-17	4.32E-23	2.49E-01	2.21E-45
	Rank	5	6	7	3	2	4	1
f_4	Best	2.47E+02	5.05E+00	2.69E+04	2.72E+01	8.54E-40	6.08E+00	8.77E-98
	Word	1.64E+03	5.29E+01	1.01E+06	8.69E+03	9.24E-22	4.37E+03	2.01E-85
	Mean	8.26E+02	1.99E+01	2.21E+05	3.02E+03	3.11E-23	9.29E+02	6.72E-87
	Std	3.55E+02	1.26E+01	2.14E+05	2.42E+03	1.66E-22	1.26E+03	3.69E-86
	Rank	6	3	7	5	2	4	1
f_5	Best	9.47E-03	1.66E-02	4.84E-02	1.27E-04	5.83E-05	/	1.34E-05
	Worst	4.06E-02	5.54E-02	1.45E-01	1.36E-03	1.22E-02	/	5.87E-04
	Mean	2.04E-02	3.41E-02	9.00E-02	5.10E-04	1.58E-03	/	1.47E-04
	Std	7.23E-03	1.06E-02	2.45E-02	3.20E-04	2.19E-03	/	1.25E-04
	Rank	4	5	6	3	2	7	1
f_6	Best	5.95E+02	1.95E+00	4.71E+02	1.14E-71	1.23E-25	5.35E-01	8.40E-100
	Worst	2.66E+03	4.63E+01	6.23E+03	1.24E-68	2.30E-21	1.56E+02	7.33E-88
	Mean	1.18E+03	1.71E+01	1.45E+03	1.06E-69	7.58E-23	4.04E+01	2.50E-89
	Std	4.05E+02	1.04E+01	9.41E+02	2.79E-69	4.01E-22	4.26E+01	1.33E-88
	Rank	7	5	6	2	3	4	1
f_7	Best	6.94E+03	1.57E+02	8.00E+05	7.56E-71	4.88E-25	4.07E+01	2.09E-96
	Worst	4.20E+04	9.74E+02	6.42E+07	9.26E-67	4.53E+06	5.26E+04	5.09E-88
	Mean	2.05E+04	3.70E+02	2.10E+07	8.30E-68	4.53E+06	7.79E+03	2.17E-89
	Std	8.68E+03	2.19E+02	1.49E+07	1.76E-67	8.14E+06	1.05E+04	9.37E-89
	Rank	6	5	7	2	3	4	1

Note: '/' means that the algorithm is invalid for the benchmark function.

5 Conclusions

Through the research and analysis of the principle and update formula of the original crow algorithm, aiming at the problems of slow convergence speed and low search accuracy in the later stage, MSD-CSA is proposed in this paper. MSD-CSA adds a sharing mechanism to the location update formula and the idea of following the global optimal location to the simple random following behavior, so as to achieve the coexistence of diversity and convergence. The perturbation of the global optimal solution in different iterative periods can expand the search range of the population, and has a strong ability to jump out of the local optimal in the later stage of the iteration. Combined with the above two improvements, the algorithm achieves a balance between global search and local search. The test results also show that the optimization ability of MSD-CSA is better than the original algorithm and other algorithms. In the next step, MSD-CSA algorithm is applied to engineering problems to further verify the effectiveness of the algorithm.

This work was supported by the Basic Ability Improvement Project for Young and Middle-Aged Teachers in Colleges and Universities in Guangxi (no. 2021KY1251).

Table 3. Results of high-dimensional multimodal benchmark functions.

Function	Index	DE	PSO	FA	BA	GWO	CSA	MSD-CSA
f_8	Best	1.50E+00	4.37E-01	1.65E+01	7.99E-15	2.22E-14	3.48E-02	8.88E-16
	Worst	5.32E+05	5.01E+00	1.90E+01	1.51E-14	3.29E-14	3.37E+00	8.88E-16
	Mean	3.23E+00	2.92E+00	1.80E+01	1.34E-14	2.82E-14	1.01E+00	8.88E-16
	Std	8.57E-01	8.90E-01	6.27E-01	2.42E-15	3.57E-15	1.07E+00	0
	Rank	6	5	7	2	3	4	1
f_9	Best	3.69E-02	8.92E-01	1.53E+02	0	2.22E-16	1.62E-04	0
	Worst	1.78E-01	1.65E+00	4.45E+02	2.55E-02	1.22E-15	1.14E+00	0
	Mean	9.30E-02	1.10E+00	3.02E+02	2.51E-03	6.70E-16	5.57E-01	0
	Std	3.68E-02	7.07E+01	6.78E+01	6.96E-03	3.16E-16	4.33E-01	0
	Rank	5	6	7	1	3	4	1
f_{10}	Best	6.97E+00	7.07E+01	6.70E+01	0	5.68E-14	2.88E-02	0
	Worst	2.89E+01	1.10E+02	2.71E+02	4.36E+00	2.27E-13	6.02E+01	0
	Mean	2.05E+01	9.04E+01	1.47E+02	2.91E-01	1.25E-13	9.43E+00	0
	Std	6.32E+00	9.90E+00	5.01E+01	1.11E+00	6.29E-14	9.43E+00	0
	Rank	5	7	6	1	3	4	1
f_{11}	Best	9.76E-03	3.71E+00	1.29E+00	4.70E-41	8.11E-23	8.91E-03	1.13E-53
	Worst	1.01E+00	1.12E+01	1.51E+01	1.32E-03	1.76E-22	6.52E-01	9.51E-47
	Mean	1.48E-01	7.38E+00	7.04E+00	8.33E-05	1.19E-22	1.09E-01	5.57E-48
	Std	2.62E-01	1.72E+00	3.49E+00	2.58E-04	2.76E-23	1.43E-01	1.81E-47
	Rank	5	7	6	2	3	4	1
f_{12}	Best	1.14E-01	1.40E-03	7.40E-02	2.38E-08	9.11E-26	8.96E-05	2.16E-100
	Worst	2.03E+00	2.32E-01	2.27E-01	2.54E-05	4.74E-25	6.56E+00	9.19E-93
	Mean	7.48E-01	2.99E-02	1.36E-01	3.55E-05	2.00E-25	1.18E+00	8.81E-94
	Std	3.90E-01	4.47E-02	3.90E-01	4.96E-06	9.07E-26	1.80E+00	2.29E-93
	Rank	7	5	6	3	2	4	1
f_{13}	Best	1.85E-01	4.00E-01	1.54E+01	6.31E-01	2.88E+00	1.01E+00	7.68E-05
	Worst	3.08E+00	2.95E+00	4.65E+01	1.54E+00	3.26E+00	1.50E+00	7.93E-01
	Mean	9.07E-01	1.20E+00	2.55E+01	1.02E+00	3.17E+00	1.30E+00	4.26E-02
	Std	7.17E-01	5.02E-01	7.53E+00	1.96E-01	9.67E-02	1.23E-01	1.52E-01
	Rank	2	3	6	4	7	5	1

	k							
f_{14}	Best	5.71E+00	5.85E+00	2.64E+04	4.52E-01	5.62E+00	1.09E+00	2.32E-03
	Worst	4.98E+01	4.24E+01	1.05E+08	1.69E+00	2.54E+02	2.30E+00	2.03E+00
	Mean	2.35E+01	1.71E+01	1.65E+07	9.85E-01	1.43E+01	1.71E+00	8.65E-01
	Std	1.22E+01	7.83E+00	2.93E+07	3.02E-01	4.53E+01	3.50E-01	5.95E-01
	Rank	5	6	7	2	4	3	1

Table 4. Results of fixed-dimensional multimodal benchmark functions.

Function	Index	DE	PSO	FA	BA	GWO	CSA	MSD-CSA
f_{15}	Best	0	1.01E-08	9.72E-03	0	1.67E-16	5.42E-06	0
	Worst	9.71E-03	1.90E-03	3.12E-01	9.72E-03	3.33E-16	5.42E-06	0
	Mean	9.71E-03	7.95E-05	1.18E-01	2.27E-03	2.68E-16	8.01E-03	0
	Std	3.62E-03	3.44E-04	8.14E-02	4.18E-03	4.98E-17	1.45E-03	0
	Rank	1	5	7	1	4	6	1
f_{16}	Best	3.29E-27	0	3.02E-10	1.45E-09	1.69E-04	1.73E-06	0
	Worst	5.52E-24	6.12E-27	6.58E-01	7.62E-01	5.81E-02	2.23E-04	0
	Mean	1.12E-24	2.28E-28	2.20E-02	5.08E-02	7.60E-03	5.60E-05	0
	Std	1.52E-24	1.12E-27	1.20E-01	1.93E-01	1.18E-02	5.71E-05	0
	Rank	3	1	4	5	7	6	1
f_{17}	Best	0.00030	0.00030	0.00035	0.00030	0.00040	0.00034	0.00030
	Worst	1.22E-03	0.00030	1.65E-02	2.02E-02	2.27E-03	1.10E-03	1.22E-03
	Mean	3.38E-04	0.00030	2.28E-03	2.34E-03	1.22E-03	7.74E-04	4.03E-04
	Std	1.67E-04	2.95E-12	3.32E-03	6.11E-03	6.40E-04	1.97E-04	2.80E-04
	Rank	1	1	6	1	7	5	1
f_{18}	Best	1.62E-27	1.80E-36	9.83E-11	0	8.15E-33	1.02E-08	3.54E-178
	Worst	1.70E-24	1.86E-32	2.99E-01	0	6.43E-27	2.42E-04	7.57E-168
	Mean	2.73E-25	2.20E-33	7.96E-02	0	3.60E-28	6.48E-05	3.06E-169
	Std	4.13E-25	4.58E-33	1.34E-01	0	1.17E-27	6.46E-05	0
	Rank	5	3	6	1	4	7	2
f_{19}	Best	0.398	0.398	0.398	0.398	0.399	0.398	0.398
	Worst	0.398	0.398	0.398	0.398	0.624	0.398	0.398
	Mean	0.398	0.398	0.398	0.398	0.441	0.398	0.398
	Std	0	0	2.67E-08	4.28E-08	4.94E-02	8.75E-05	0

	Ran k	1	1	1	1	7	1	1
f_{20}	Best	3	3	3	3	3.04E+00	3	3
	Wor st	3	3	3.0E+01	3	9.84E+00	3.01E+00	3
	Mea n	3	3	4.80E+00	3	5.18E+00	3.00E+00	3
	Std	2.21E-15	1.06E-15	6.85E+00	4.59E-06	2.02E+00	3.91E-03	6.39E-16
	Ran k	1	1	1	1	7	1	1

References

1. Sayed, Gehad Ismail, et al. "Feature Selection via a Novel Chaotic Crow Search Algorithm." *Neural Computing and Applications*, vol. 31, no. 1, 2019, pp. 171–188.
2. Kennedy, J., and R. Eberhart. "Particle Swarm Optimization." *Neural Networks, 1995. Proceedings., IEEE International Conference On*, vol. 4, 2002, pp. 1942–1948.
3. Yang, Xin-She, and Xingshi He. "Bat Algorithm: Literature Review and Applications." *International Journal of Bio-Inspired Computation*, vol. 5, no. 3, 2013, pp. 141–149.
4. Arora, Sankalop, and Satvir Singh. "Butterfly Optimization Algorithm: A Novel Approach for Global Optimization." *Soft Computing - A Fusion of Foundations, Methodologies and Applications Archive*, vol. 23, no. 3, 2019, pp. 715–734.
5. Yang, Xin-She. "Flower Pollination Algorithm for Global Optimization." *UCNC'12 Proceedings of the 11th International Conference on Unconventional Computation and Natural Computation, 2012*, pp. 240–249.
6. Duan, Haibin, and Peixin Qiao. "Pigeon-Inspired Optimization: A New Swarm Intelligence Optimizer for Air Robot Path Planning." *International Journal of Intelligent Computing and Cybernetics*, vol. 7, no. 1, 2014, pp. 24–37.
7. Mirjalili, Seyedali, and Andrew Lewis. "The Whale Optimization Algorithm." *Advances in Engineering Software*, vol. 95, no. 95, 2016, pp. 51–67.
8. Mirjalili, Seyedali, et al. "Grey Wolf Optimizer." *Advances in Engineering Software*, vol. 69, 2014, pp. 46–61.
9. Rao, R.Venkata. *Teaching–Learning-Based Optimization Algorithm*. 2016, pp. 211–216.
10. Askarzadeh, Alireza. "A Novel Metaheuristic Method for Solving Constrained Engineering Optimization Problems: Crow Search Algorithm." *Computers & Structures*, vol. 169, 2016, pp. 1–12.
11. Abdallah, Ghada Yousif, and Zakariya Yahya Algamal. "A QSAR Classification Model of Skin Sensitization Potential Based on Improving Binary Crow Search Algorithm." *Electronic Journal of Applied Statistical Analysis*, vol. 13, no. 1, 2020, pp. 86–95.
12. Upadhyay, Pankaj, and Jitender Kumar Chhabra. "Kapur's Entropy Based Optimal Multilevel Image Segmentation Using Crow Search Algorithm." *Applied Soft Computing*, vol. 97, 2020, p. 105522.
13. Ouadfel, Salima, and Mohamed E.Abd Elaziz. "Enhanced Crow Search Algorithm for Feature Selection." *Expert Systems with Applications*, vol. 159, 2020, p. 113572.
14. Devikanniga, D., et al. "Efficient Diagnosis of Liver Disease Using Support Vector Machine Optimized with Crows Search Algorithm." *EAI Endorsed Transactions on Energy Web*, vol. 7, no. 29, 2018, p. 164177.

15. Siswanto, Nurhadi, et al. "A Crow Search Algorithm for Aircraft Maintenance Check Problem and Continuous Airworthiness Maintenance Program." *Jurnal Sistem Dan Manajemen Industri*, vol. 3, no. 2, 2019, pp. 115–123.
16. Arora, Sankalap, et al. "A New Hybrid Algorithm Based on Grey Wolf Optimization and Crow Search Algorithm for Unconstrained Function Optimization and Feature Selection." *IEEE Access*, vol. 7, 2019, pp. 26343–26361.
17. Wu, Hao, et al. "Finite Element Model Updating Using Crow Search Algorithm with Levy Flight." *International Journal for Numerical Methods in Engineering*, vol. 121, no. 13, 2020, pp. 2916–2928.
18. Liu Xuejing, He Yichao, Wu Congcong & Li Liang. Chaotic binary crow algorithm for solving 0-1 knapsack problem. *Computer Engineering and Applications*, vol. 54, no. 10, 2018, pp. 173-179.
19. Mohammadi, Farid, and Hamdi Abdi. "A Modified Crow Search Algorithm (MCSA) for Solving Economic Load Dispatch Problem." *Applied Soft Computing*, vol. 71, 2018, pp. 51–65.
20. Xiao ZY, Liu S, Han FF & Yu JF. Research on crow search algorithm guided by sine and cosine. *Computer Engineering and Applications* 2019, pp,52-59.
21. Arora, Sankalap, et al. "A New Hybrid Algorithm Based on Grey Wolf Optimization and Crow Search Algorithm for Unconstrained Function Optimization and Feature Selection." *IEEE Access*, vol. 7, 2019, pp. 26343–26361.
22. Pratiwi, Asri Bekt. "A Hybrid Cat Swarm Optimization - Crow Search Algorithm for Vehicle Routing Problem with Time Windows." 2017 2nd International Conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE), 2017.
23. Mao Qing-Hua et al." Improved Grey Wolf Algorithm Fusing Tent Chaos and Simulated Annealing." *Mathematics in Practice and Understanding*, vol. 51, no. 05, 2021, pp.147-168.
24. Storn R, Price K. Differential evolution-A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 1997,11(4):341-359.
25. Kennedy J, Eberhart R. Particle swarm optimization// *Proceedings of ICNN'95-International Conference on Neural Networks*, IEEE, 1995.
26. Yang X S. A new metaheuristic bat-inspired algorithm. *Computer Knowledge & Technology*, 2010, 284:65-74.
27. Fister I, Fister I, Yangx S, et al. A comprehensive review of firefly algorithms. *Swarm and Evolutionary Computation*, 2013, 13(Complete):34-46.
28. Mirialili S, Mirialili S M, Lewis A. Grey wolf optimizer. *Advances in Engineering Software*, 2014, 69(3):46-61.