Research on chaotic flying sparrow search algorithm

Xiaoxiao Chen\textsuperscript{1*}, Xueyu Huang\textsuperscript{1}, Donglin Zhu\textsuperscript{1} and Yaxian Qiu\textsuperscript{1}

\textsuperscript{1}School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou, Jiangxi 341000, China

*Corresponding author’s e-mail: 6120180099@mail.jxust.edu.cn

Abstract: The sparrow search algorithm has attracted much attention due to its excellent characteristics, but it still has shortcomings such as falling into the local optimum and relying on the initial population stage. In order to improve these shortcomings, the chaotic flying sparrow search algorithm is proposed. In the initialization, the chaotic mapping based on random variables is introduced to make the population distribution more uniform and speed up the optimization efficiency of the population. In the discoverer stage, the dynamic adaptive search strategy and levy flight mechanism are used to increase the search range and flexibility, and the random walk strategy is introduced to make the follower's search more detailed and avoid premature phenomenon. The effectiveness of the improved algorithm is verified by six standard test functions, and the introduction of a variety of strategies greatly enhances the optimization ability of the algorithm.

1. Introduction

In recent years, swarm intelligence optimization algorithms have been favored by researchers. Its characteristic is to find the optimal solution in a certain range according to some things in nature and the behavior of organisms. Scholars have put forward a series of swarm intelligence optimization algorithms through the behavior of ants, wolves, longicorn, cats, chickens, sparrows and so on. The more typical algorithms are: ant colony optimization, grey wolf optimization, beetle antennae search algorithm, sparrow search algorithm and so on. The Sparrow search algorithm (SSA) is a new group intelligence optimization algorithm proposed by two scholars, Jian Kai Xue and Bo Shen\textsuperscript{1} in 2020, which searches for the optimal solution through the sparrow feeding process. The algorithm is simple in principle, has fewer population roles and parameters, and is robust. It has higher accuracy and better searching ability than traditional particle swarm optimization (PSO) and grey wolf optimization (GWO) in function optimization. The application of SSA in practical projects has also started to increase. For example, the author Jian Kai Xue has successfully applied it to UAV track planning and achieved good results. However, SSA also has some drawbacks of swarm intelligence algorithm, such as the probability of falling into local optimum and dependent on the initialization of population stage.

In order to solve the shortcomings of sparrow search algorithm, Lu Xin\textsuperscript{2} proposed a chaotic sparrow search algorithm, which used the improved chaotic map to initialize the population, and then introduced Gaussian mutation to prevent the "clustering" of sparrow individuals, thus avoiding the local optimum to some extent. Mao Qinghua\textsuperscript{3} et al. proposed an improved sparrow algorithm combining Cauchy mutation and reverse learning. Sparrow population is initialized by sin chaos to enrich the diversity of solutions. By introducing a dynamic adaptive weighting factor, the global and local mining capabilities of the algorithm are effectively balanced. Combining Cauchy mutation and reverse learning strategies reduces the probability of the algorithm falling into local extremes and improves the global exploratory
performance. At present, the improved sparrow search algorithm has been proposed gradually, but most of the improved ideas do not change the optimization mechanism of the sparrow algorithm. The improved strategy is num

Based on previous studies, this paper presents a multi-strategy improved sparrow search algorithm (CDLSSA). First, tent chaos based on random variables is used to initialize the population so that sparrows are evenly distributed and the complexity of initial optimization is reduced. In the discoverer stage, dynamic adaptive weights and Levy flight mechanism are combined to dynamically adjust the sparrow’s search mode, which provides a more detailed and extensive search, thus avoiding premature puberty. Finally, a reverse learning strategy based on lens imaging is introduced, which makes follower location update more diverse and reduces the probability of falling into local optimum.

2. Fundamentals of Sparrow Search Algorithm

In the process of searching for food, the sparrow population has two roles: discoverer and follower. The discoverer mainly searches for food for sparrow companions and conveys information to followers about where the food is located. At the same time, more adaptable discoverers prefer to get food during the feeding process. So the discoverer searches for food more widely than the follower. In each iteration, the location update formula for the discoverer is as follows:

\[ X_{ij}^{t+1} = \begin{cases} X_{ij}^t \cdot \exp \left( \frac{t - t_i}{\alpha \cdot \text{iter}_{\text{max}}} \right) & \text{if } R_2 < ST \\ X_{ij}^t + Q \cdot L & \text{if } R_2 \geq ST \end{cases} \]  

In equation (1), \( t \) denotes the current number of iterations, and \( \text{iter}_{\text{max}} \) is the maximum number of iterations. \( X_{ij} \) denotes the position of the current i-th sparrow in dimension j. \( \alpha \in (0,1] \) is a random number. \( R_2 (R_2 \in [0,1]) \) and \( ST (ST \in [0.5,1]) \) denote a warning value and a safety value, respectively. \( Q \) is a normally distributed random number. \( L \) represents a 1 x d matrix in which all elements are equal to 1. When \( R_2 < ST \) indicates that no predators are found around the discoverer's location, it can perform a wide range of search operations. When \( R_2 < ST \) indicates that some sparrows are aware of predators and alert the entire sparrow community, all must flee to other locations quickly before foraging.

Followers' location updates are described below:

\[ X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp \left( \frac{X_{\text{worst}}^t - X_{i,j}^t}{T^2} \right) & \text{if } i > n/2 \\ X_{p}^{t+1} + \left| X_{i,j}^t - X_{p}^{t+1} \right| \cdot A^+ \cdot L & \text{otherwise} \end{cases} \]  

In equation (2), \( X_p \) is the optimal position currently occupied by the discoverer, and \( X_{\text{worst}} \) represents the current worst position. \( A \) is a 1 x d matrix whose elements are only 1 or -1, where \( A^+ = A^T (AA^T)^{-1} \). When \( i > n/2 \), it means that the i-th follower with a lower fitness value cannot get any food, is very hungry and must go elsewhere to find more food.

When the sparrow population is aware of the danger, it acts as an anti-predator, and its mathematical expression is as follows:

\[ X_{i,j}^{t+1} = \begin{cases} X_{\text{best}}^t + \beta \cdot \left| X_{i,j}^t - X_{\text{best}}^t \right| & \text{if } f_i > f_g \\ X_{i,j}^t + K \cdot \left( \frac{x_{i,j}^t - x_{\text{worst}}^t}{(f_i - f_{\text{worst}}) + \epsilon} \right) & \text{if } f_i = f_g \end{cases} \]  

In equation (3), \( X_{\text{best}} \) is the current global optimal position. As a step control parameter, \( \beta \) is a random number which obey the normal distribution with mean 0 and variance 1. \( K \in [-1,1] \) is a random number, and \( f_i \) is the current fitness value for sparrows. \( f_g \) and \( f_{\text{worst}} \) are the highest and lowest fitness values in the whole sparrow population. \( \epsilon \) is the minimum constant to avoid zero in the denominator. For simplicity, when \( f_i > f_g \), it means that the sparrows are currently on the periphery of their feeding position and are very vulnerable to predators. When \( f_i = f_g \), it means that sparrows in the center of the feeding range find predators and need to move closer to nearby sparrows to reduce their chance of being caught. \( K \) represents the direction of sparrow migration and is also a step control parameter.
3. Improved sparrow search algorithm

3.1. Tent mapping based on random variables

Tent mapping has the advantages of randomness, uniformity and order, and it is a common method for scholars to find the optimal solution. However, tent chaos has the disadvantage of small periods and instability. To improve this effect, tent mapping based on random variables in literature\[2\] is introduced. Therefore, this paper uses tent mapping based on random variables to initialize the population, improve the diversity of the population. The tent mapping expression based on random variables is as follows:

\[
Z_{i+1} = \begin{cases} 
2z_i + \text{rand}(0,1) \times \frac{1}{N}, & 0 \leq z \leq \frac{1}{2} \\
2(1-z_i) + \text{rand}(0,1) \times \frac{1}{N}, & \frac{1}{2} \leq z \leq 1 
\end{cases}
\]

(4)

The expression after the Bernoulli transformation is:

\[
Z_{i+1} = (2z_i) \mod 1 + \text{rand}(0,1) \times \frac{1}{N} 
\]

(5)

In equation (5), \(N\) is the number of particles of chaotic sequence.

3.2. Dynamic Adaptive Weighting Strategy and Levy Flight

Discoverers lead other individuals to find the best solution, which requires a flexible search mechanism. Otherwise, they are prone to fall into local extremes, which hinders the overall search ability. The dynamic adaptive weighting strategy and the Levy flight mechanism are introduced to make the searching scope of the discoverer broader and more detailed. Experiments show that the dynamic adaptive weighting strategy alone can not effectively improve the searching ability of the discoverer, and there is blindness in the search. Therefore, the combination of Levy flight mechanism enlarges the search range, and the regularity of weight strategy improves the randomness of Levy flight. The two complementary advantages improve the efficiency of the discoverer, and each iteration obtains a quality solution.

The dynamic weight factor \(w\) is introduced in the discoverer stage, and changes dynamically with the number of iterations, which makes the algorithm have a better search range at the beginning, reduces the search range adaptively at the later stage, improves the search ability, balances the impact of the Levy flight mechanism on the later stage, and guarantees the optimization accuracy of the algorithm to a certain extent. The discoverer's formula for introducing the weight factor \(w\) is as follows:

\[
w = e^{\frac{t}{\text{iter}_{\text{max}}}} \cdot e^{-\frac{t}{\text{iter}_{\text{max}}}} 
\]

\[
X_{i,j}^{t+1} = \begin{cases} 
X_{i,j}^t \cdot \exp\left(-\frac{t}{\alpha \text{iter}_{\text{max}}^2}\right) \cdot w & \text{if } R_2 < ST \\
X_{i,j}^t + Q \cdot L \cdot w & \text{if } R_2 \geq ST 
\end{cases}
\]

(6)

(7)

The sparrow search algorithm has fewer population roles, so the initial stage of population work is complex and the same food may be occupied by multiple roles. The performance of the algorithm is hindered by reduced search efficiency and the dependence of the algorithm performance on the initialization phase. Because of the wide search scope and global nature of the discoverer, the Levy flight strategy is used to dynamically adjust the location of the discoverer in the initial stage, which makes the search scope wider, improves the search efficiency and search speed to a certain extent, and overcomes the impact of the initial stage population.

Levy distribution is a probability distribution proposed by Levy, a French mathematician in the 1930s. Levy flight, as a random search method obeying Levy distribution, uses the search mechanism of long-distance and short-distance flight to make the search range wider, and effectively reduces the probability of falling into local optimum, thus enhancing the search ability of the algorithm.

Levy's flight position is updated as follows:

\[
x_i(t) = x_i(t) + l \oplus \text{levy}(\lambda) 
\]

(8)
Where \( x_i(t) \) represents the discoverer location of the above introduced formula (7). \( \oplus \) denotes point-to-point multiplication. \( l \) represents the weight of the control step, \( l=0.01(x_i(t)-x_b) \), \( x_b \) is the current optimal solution. Levy (\( \lambda \)) represents a path that follows the Levy distribution and satisfies the following conditions: \( \text{levy} \sim u=t-\lambda, 1<\lambda \leq 3 \).

In Mantegna's algorithm, we can define Levy flight step as:

\[
S = |u|^{1/\beta} \quad (9)
\]

In equation (9), \( S \) is the Levy flight path; \( u \) and \( v \) are random numbers with normal distribution. \( u \sim N(0, \sigma_u^2), v \sim N(0, \sigma_v^2) \). Where \( \sigma_u, \sigma_v \) are derived from equation (10).

\[
\begin{align*}
\sigma_u &= \left( \frac{\Gamma(1+\beta)}{\Gamma(\frac{1+\beta}{2})\beta \Gamma(\beta/2)} \right)^{1/\beta} \\
\sigma_v &= 1
\end{align*} \quad (10)
\]

In equation (10), the value range of parameter \( \beta \) is \( 0 < \beta < 2 \), generally \( \beta = 1.5 \).

3.3. Inverse learning strategy based on lens imaging

Followers follow the discoverers to find the optimal value. When the discoverer is in the local optimal, the whole population is clustered in the local optimal area, which reduces the diversity of the population and makes the algorithm easy to fall into the local optimal. In order to improve the followers' optimization method, this paper presents a reverse learning strategy based on lens imaging \[4\], which is applied to followers' individuals so that new individuals can be obtained in each iteration and population diversity can be improved. The principle is described as follows:

Definition 1. Reverse Point: Set \( X=(x_1, x_2, \ldots, x_D) \) is a point in \( D \)-dimensional space, and \( x_j \in [a_j, b_j], j=1, 2, \ldots, D \), then the reverse point of \( X \) is \( X' = (x'_1, x'_2, \ldots, x'_D) \), and \( x'_j = a_j + b_j + x_j \).

Definition 2. Base point: Suppose there are several points \( O_1, O_2, \ldots, O_m \), for any point \( X=(x_1, x_2, \ldots, x_D) \) with its reverse point \( X' = (x'_1, x'_2, \ldots, x'_D) \) to \( O_i (i=1, 2, \ldots, m) \), its euclidean distances are \( d_i \) and \( d'_i \), make \( k = d_i/d'_i \), and \( k=1, 2, \ldots, n \), then \( O_i \) is called the base point of \( X \) and \( X' \) when \( k = i \).

Taking one-dimensional space as an example, suppose that the projection of an individual \( P \) with height \( h \) on the coordinate axis is \( x_p \) (globally optimal individual). A lens with focal length \( f \) is placed on the base point \( O \) (the midpoint of \( [a, b] \) in this paper). Through the lens imaging process, an image \( P' \) with height \( h' \) can be obtained, and its projection on the coordinate axis is \( x'_p \). At this point, \( x'_p \) is the new individual produced by \( x_p \) through the reverse learning strategy based on the principle of lens imaging. The schematic diagram is shown in Figure 1.

![Figure 1 Schematic diagram of lens principle](image_url)
As shown in Figure 1, the global optimal individual \( x_p \) obtains its corresponding reverse point \( x'_p \) based on \( O \). According to the principle of lens imaging, it can be concluded that:

\[
\frac{a+b}{2} - x_p = \frac{h}{h'}
\]

(11)

Let \( \frac{h}{h'} = k \), \( k \) is the zoom factor. The reverse point can be obtained by transformation:

\[
x'_p = \frac{a+b}{2} + \frac{a+b}{2k} x_p
\]

(12)

Thus, when \( k = 1 \):

\[
x'_p = a + b - x_p
\]

(13)

Equation (13) is called general reverse learning strategy. It can be seen from the above formulas that general learning strategy is only a specific case of lens imaging learning strategy, and the new individuals obtained by general reverse learning strategy are fixed each time. In high-dimensional complex functions, new individuals with fixed ranges are also likely to fall into local optimum and are monotonic. By adjusting parameter \( k \), new individuals derived from lenses imaging learning strategies are dynamic, which improves population diversity.

By generalizing formula (13) to \( d \)-dimensional space, we can get that:

\[
x'^{ij}_p = \frac{a^{ij} + b^{ij}}{2} + \frac{a^{ij} + b^{ij}}{2k} x^{ij}_p
\]

(14)

In formula (14), \( x^{ij}_p \) and \( x'^{ij}_p \) are the \( j \)-th-dimensional components of \( x_p \) and \( x'_p \) respectively, and \( a^{ij} \) and \( b^{ij} \) represent the \( j \)-th-dimensional components of the upper and lower bounds of the decision variable, respectively.

4. Improve Sparrow Search algorithm flow

| INPUT |
|---|
| \( M \): Maximum number of iterations |
| \( PD \): Population number of producers |
| \( SD \): Number of sparrows aware of danger |
| \( R_2 \): Alert value |
| \( N \): Total population quantity |

| OUTPUT: \( X_{best}, f_g \) |

\( t = 1 \)

Introduced equation (5) chaotic mapping to initialize population

\( \text{While}(t < G) \)

Find the best and worst position of the current sparrow population based on fitness values

\( R_2 = \text{rand}(1) \)

For \( i = 1 : PD \) : (1)

Update the location of the discoverer based on equation (8)

End for

For \( i = (PD + 1) : N \)

Update the location of the follower based on equation (14)

End for

For \( i = 1 : SD \)

Update dangerous sparrows’ individual locations based on equation (3)

End for

Get the location of the new optimal individual

Update optimal location if new individual location is better than previous individual location

\( t = t + 1 \)

End while

Return: \( X_{best}, f_g \)
5. Simulation and analysis

5.1. Experimental Environment and Parameter Settings

Each algorithm runs on a Window10 64bit system with 8GB of memory and Intel(R) Core (TM) i7-5500U CPU @ 2.40GHz processor. To prove the effectiveness and practicability of the improved algorithm, the algorithm (CSSA) in [2] is compared with particle swarm optimization (PSO), grey wolf optimization (GWO) and sparrow search algorithm (SSA). The population size of each algorithm is 100 and the number of iterations is 1000. Among them, the weight of PSO is 0.729, and the two learning factors $c1=c2=1.49445$. MATLAB R2018b is used for simulation.

### Table 1 Benchmark function test table

| FUNCTION | DIM | SECTION | MIN |
|----------|-----|---------|-----|
| $F_1(x) = \max_i{|x_i|}, 1 \leq i \leq n$ | 30 | [-100,100] | 0 |
| $F_2(x) = \sum_{i=1}^{n-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$ | 30 | [-100,100] | 0 |
| $F_3(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{|x_i|})$ | 30 | [-500,500] | -418.98 |
| $F_4 = 418.9829n - \sum_{i=1}^{n} x_i \sin(\sqrt{|x_i|})$ | 30 | [-500,500] | 0 |
| $F_5(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1)$ | 4 | [-10,10] | 0 |
| $F_6(x) = (0.002 + \sum_{i=1}^{25} \frac{1}{i + (x_1 - a_{1i})^6 + (x_2 - a_{2i})^6})^{-1}$ WHERE $a = (-32 -16 0 \ 16 32 -32 \ldots 0 \ 16 32 \ldots -32 -32 -32 -32 -16 \ldots 32 32 32)$ | 2 | [-65.536,65.536] | 6) | 0.998 |

The benchmark functions are shown in Table 1. In this paper, six test functions are used to verify and analyze the optimization effect of the improved sparrow search algorithm, in which there are two unimodal, multimodal and fixed dimension functions in turn. Each algorithm runs 30 times independently and counts the optimal value, average value and standard deviation of each algorithm. Three indexes are used to test the optimization ability and stability of the algorithm. The experimental results of algorithm optimization are shown in Table 2.

### Table 2 Algorithmic optimization compared with experimental results

| FUNCTION | ALGORITHM | OPTIMAL VALUE | AVERAGE VALUE | STANDARD DEVIATION |
|----------|-----------|---------------|---------------|--------------------|
| $F_1(x)$ | CDLSSA    | 0             | 0             | 0                  |
|          | CSSA      | 0             | 0             | 0                  |
FUNCTION | ALGORITHM | OPTIMAL VALUE | AVERAGE VALUE | STANDARD DEVIATION
---|---|---|---|---
\(F_2(x)\) | SSA | 4.0636E-14 | 1.9036E-07 | 2.9266E-07
 | GWO | 2.4178E-06 | 6.4757E-06 | 2.1850E-06
 | PSO | 8.1218E-14 | 6.9044E-13 | 7.0838E-13
\(F_3(x)\) | CDLSSA | 6.19408E-08 | 1.99154E-05 | 4.59655E-05
 | CSSA | 0 | 4.2858E-192 | 0
 | SSA | 0 | 7.1371E-210 | 0
 | GWO | 2.2250E-04 | 0.029431 | 0.070791
 | PSO | 22.2592 | 49.9596 | 18.1286
\(F_4(x)\) | CDLSSA | 0.056188601 | 3685.49672 | 2025.58711
 | CSSA | 2529.3974 | 3661.9962 | 725.5963
 | SSA | 3471.670439 | 4292.617883 | 529.1990723
 | GWO | 4866.1835 | 6134.3633 | 604.5177
 | PSO | 5205.7803 | 7638.9387 | 1024.9673
\(F_5(x)\) | CDLSSA | 1.25461E-14 | 1.2361E-07 | 4.06158E-07
 | CSSA | 4.55454E-14 | 5.22137E-07 | 1.36526E-06
 | SSA | 9.50199E-13 | 1.02709E-06 | 2.51357E-06
 | GWO | 0.00022591 | 1.142301517 | 2.11174329
 | PSO | 0.001130521 | 0.026019705 | 0.02267333
\(F_6(x)\) | CDLSSA | 0.998 | 1.1631 | 0.630895495
 | CSSA | 0.998 | 2.1068 | 2.9273
 | SSA | 0.998 | 2.5594 | 3.6421
 | GWO | 0.998 | 2.8013 | 2.8092
 | PSO | 0.998 | 1.1968 | 0.3976

5.2. Optimizing performance analysis
It can be seen from the data in Table 2 that CDLSSA is superior to the other four algorithms in convergence and solution accuracy in six functions. More specifically, in the F1-2, F5-6 functions, there is strong convergence accuracy and stability, and the other two functions show CDLSSA has significant optimization ability. Therefore, CDLSSA has a good convergence effect on unimodal function, CDLSSA has a strong ability to resist local extremes on multimodal function, and the introduction of a variety of strategies makes the algorithm search better and more flexible.
5.3. **Convergence Analysis**

![Figure 2 Optimizing Convergence Diagram of algorithms on F1 function](image1)

![Figure 5 Optimizing Convergence Diagram of algorithms on F2 function](image2)

![Figure 3 Optimizing Convergence Diagram of algorithms on F3 function](image3)

![Figure 6 Optimizing Convergence Diagram of algorithms on F4 function](image4)

![Figure 4 Optimizing Convergence Diagram of algorithms on F5 function](image5)

![Figure 7 Optimizing Convergence Diagram of algorithms on F6 function](image6)

As shown in Figures 2-7, CDLSSA shows good convergence. In F2, F3 and F5, CDLSSA has obvious anti-local attractiveness. In F6, PSO shows better performance and SSA has poor convergence. With the introduction of various strategies, the convergence effect of CDLSSA and PSO is almost identical. Therefore, the introduction of a variety of strategies has improved CDLSSA to a certain extent, and has flexibility in search methods.

6. **Conclusion**

The sparrow search algorithm has better search performance than other algorithms, but there are still some drawbacks. In order to overcome these shortcomings, a chaotic flight sparrow search algorithm is
proposed. In the initialization phase, a chaotic map based on random variables is introduced, which makes the individual distribution of sparrows more uniform and improves the efficiency of the population. Dynamic adaptive weights and Levy flight mechanisms are combined in the discoverer stage to increase the search range and flexibility. By introducing a reverse learning strategy based on lens imaging, followers' location updates can be varied, balancing local and global searches. The experimental results show that the improved algorithm has strong optimization ability and can effectively avoid premature convergence.

Acknowledgments
This work was supported in part by The National Key Research and Development Program of China (2020YFB1713700). Then I would like to express thanks and respect to Mr. Huang, who inspires me to keep learning new knowledge. Finally, I would like to express my sincere thanks to all the people who care about and help me.

Reference
[1] Xue J, Shen B. A novel swarm intelligence optimization approach: sparrow search algorithm[J]. Systems Science & Control Engineering An Open Access Journal, 2020, 8(1):22-34.
[2] Lu Xin, Mu Xiaodong, Zhang Jun. multi threshold image segmentation based on improved sparrow search algorithm [J / OL]. Systems engineering and electronic technology:1-12[2021-01-25].http://kns.cnki.net/kcms/detail/11.2422.TN.20201106.1803.018.html.
[3] Mao Qinghua, Zhang Qiang. Improved Sparrow Algorithm Combining Cauchy Mutation and Opposite Learning [J/ol]. Computer Science and Exploration: 1-12 [2021-01-05].UNK1.5602.tp.20201203.1601.006.html.
[4] Long Wen, Wu tiebin, Tang Mingzhu, Xu Ming, Cai Shaohong. Gray wolf optimization algorithm based on lens imaging learning strategy [J]. Acta automatica Sinica, 2020,46 (10): 2148-2164.
[5] Fu Xingwu, Hu Yang. Three dimensional path planning based on improved particle swarm optimization [J / OL]. Electro Optics and control:1-5[2021-01-25]. http://kns.cnki.net/kcms/detail/41.1227.TN.20201201.1055.016.html.
[6] Li Yang, Li Weigang, Zhao Yuntao, Liu Ao. Gray Wolf Algorithm Based on Levy Flight and Random Walk Strategy. Computer Science, 2020, 47 (08): 291-296.
[7] Lu Xin, Mu Xiaodong, Zhang Jun. multi threshold image segmentation based on improved sparrow search algorithm [J / OL]. Systems engineering and electronic technology:1-12[2021-01-25].http://kns.cnki.net/kcms/detail/11.2422.TN.20201106.1803.018.html.
[8] Yuan Yan, Cao Cuiwen. Improved Grey Wolf Algorithm and Its Application. Computer Engineering and Design, 2020, 41 (02): 513-521.
[9] Li Yali, Wang Shuqin, Chen Qianru, Wang Xiaogang. Comparative study on some new swarm intelligence optimization algorithms [J]. Computer engineering and application, 2020,56 (22): 1-12.
[10] Ren Yan, Zhao Haibo, Xiao Yongjian. Obstacle avoidance and path planning of robot based on improved potential field ant colony algorithm [J]. Electro Optics and control, 2019,26 (11): 75-79.
[11] Cheng Qian, Gao Song, Cao Kai, Chen Chaobo. Path planning of mobile robot based on PRM optimization algorithm [J]. Computer applications and software, 2020,37 (12): 254-259 + 296.
[12] Huang Chenchen, Wei Xia, Huang Deqi, ye Jiahao. Hybrid frog leaping gray wolf optimization algorithm for solving high dimensional complex functions [J]. Control theory and application, 2020,37 (07): 1655-1666.
[13] Li Tianlai, Liu fangai. Butterfly optimal localization algorithm with chaotic mapping in WSN [J]. Computer engineering and design, 2019,40 (6): 1729-1733.
[14] Xue Jiankai. Research and application of a new swarm intelligence optimization technology [D]. Donghua University, 2020.
[15] Hu Zhangfang, Sun Lin, Zhang Yi, Bao Hezhang. A robot path planning algorithm based on improved QPSO [J]. Computer Engineering, 2019,45 (04): 281-287.

[16] Kang Yuxiang, Jiang Chunying, Qin Yunhai, ye Changlong. Robot path planning and experiment based on Improved PSO algorithm [J]. Robot, 2020,42 (01): 71-78.

[17] Zhang hang, Yao Minru, Cao Kai, Gao Song. Multi robot map construction based on improved particle swarm optimization [J]. Computer applications and software, 2019,36 (10): 162-167 + 174.