A multi-strategy improved sparrow search algorithm

Chengtian OUYANG¹, Yaxian QIU¹* and Donglin ZHU¹

¹School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou, Jiangxi 341000, China
*Corresponding author’s e-mail: 6120200219@mail.jxust.edu.cn

Abstract: As a novel algorithm, the sparrow search algorithm has better optimization performance than other intelligent optimization algorithms. However, in complex problems, there is still the possibility of falling into a local optimum and relying on the initial population stage. In response to these shortcomings, a multi-strategy improved sparrow search algorithm (KLSSA) is proposed. First, in the initial population stage, K-means clustering method is used to cluster and differentiate the individual positions of sparrows, which speeds up the work efficiency of the population and gets rid of the influence of randomness. Then, the levy flight mechanism and adaptive local search strategy are respectively introduced in the calculation of the location update of the discoverer and the follower, so that the discoverer can conduct a wide range of searches more flexibly, and the follower has a more detailed search method. Through the 10 standard test functions, it can be seen that the multi-strategy improved sparrow search algorithm has stronger optimization ability and better optimization speed.

1. Introduction
The Sparrow Search Algorithm (SSA) is a new swarm intelligence optimization algorithm proposed by two scholars, Jiankai Xue and Bo Shen, in 2020 to find the optimal solution through the sparrow foraging process. The algorithm is simple in principle, with fewer population roles and self-parameters. In terms of function optimization, it has higher precision and better optimization ability than traditional particle swarm algorithm and gray wolf algorithm. The application in actual engineering has also begun to increase. For example, the proponent Jiankai Xue has successfully applied it in UAV trajectory planning and achieved good results. However, the SSA algorithm also has some disadvantages to the swarm intelligence algorithm, such as the probability of falling into a local optimum and depending on the initial population stage.

In order to make up for the defects of the sparrow search algorithm, Lu Xin [2] proposed a chaotic sparrow search algorithm. The chaos strategy is used to initialize the population, and Gaussian mutation is introduced to prevent the sparrow individuals from "clustering" and premature convergence, and to a certain extent avoid local optimality. Mao Qinghua [3] discovered the importance of sparrow position initialization for global search, and proposed to use Sin chaos to initialize the sparrow population to enrich the diversity of solutions. The introduction of dynamic adaptive weighting factors effectively balances the overall and local mining capabilities of the algorithm. It also integrates Cauchy mutation and reverse learning strategies to reduce the probability of the algorithm falling into a local extreme and improve the performance of global exploration. At present, improved sparrow search algorithms are gradually being proposed, but most improvements do not improve the optimization mechanism of the sparrow algorithm itself, so that the algorithm still has a large randomness in the global search ability.
To solve the above problems, this paper proposes a multi-strategy improved sparrow search algorithm (KLSSA). The algorithm uses K-means to initialize the population stage, which reduces random interference. Then, the levy flight mechanism and adaptive local search strategy are introduced respectively in the discoverer and follower stages, which dynamically adjust the search method of the sparrow, and have a more detailed and extensive search, so as to avoid the premature algorithm. In order to verify the optimization performance of the KLSSA algorithm, simulation experiments were performed on multiple test functions. The results show that the algorithm has higher optimization accuracy and faster convergence speed.

2. Sparrow search algorithm

In the process of sparrow foraging, the sparrow population has two roles: discoverer and follower. The discoverer finds the food and transmits the information of food location to the whole sparrow population, while the follower follows the discoverer in foraging, finding and obtaining the food. Each sparrow will monitor the behavior of other members in the group. If there are sparrows with high intake in the group, the attackers in the group will compete with them for food to improve their predation rate. When danger approaches, they give up foraging and the whole colony moves to a new location.

The location of the update discoverer is calculated as follows:

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

(1)

Where \(X_{i,j}^t\) represents the position of the \(i\)-th sparrow in the \(j\)-th dimension in the \(t\) generation of the sparrow population, \(\alpha\) is the random number in \((0,1]\), and \(Q\) is the standard normal distribution random number, \(R_2\) is the random number in \([0,1]\), \(ST\) is the warning threshold, and the value range is \([0.5,1.0]\). \(L\) denotes a \(1\times d\) matrix in which all elements are 1. \(R_2 < ST\) indicates that there is no predator in the current position, and the discoverer can continue to forage; \(R_2 \geq ST\) indicates that there is a predator in the current foraging area, and the discoverer needs to avoid danger and move to a safe area.

The calculation formula for updating follower position is as follows:

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

(2)

\(X_{p}^{t+1}\) is the optimal position of the \(t+1\) generation of sparrow population, \(X_{\text{worst}}^t\) represents the worst position of the \(t\) generation of sparrow population, \(X_{p}^t\) represents a \(1\times d\) matrix, in which each element is randomly assigned to 1 or -1, and \(A^+\) represents a \(1\times d\) matrix, in which each element is randomly assigned to 1 or -1, and \(A^+ = A^T(AA^T)^{-1}\). When \(i > n/2\), the sparrow did not get food and needed to find a new location; when \(i \leq n/2\), the sparrow moved to the current optimal location to get more food.

The formula for calculating the update location of reconnaissance and warning is as follows:

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

(3)

\(X_{\text{best}}^t\) denotes the optimal position of the \(t\) generation of the sparrow population, \(\beta\) is a random number that conforms to the standard normal distribution, \(K\) is a random number of \([-1,1]\], \(f_i\) is the fitness value of the current sparrow individual, \(f_g\) and \(f_w\) is the best and worst fitness value for the current region, and \(\varepsilon\) is a smaller number to prevent the denominator from being zero. \(f_i > f_g\) indicates that this sparrow is not optimally located in the current foraging area and, when at risk, it will move near the optimal location; \(f_i = f_g\) suggests that sparrows in the middle of the population are aware of the danger and need to be close to other sparrows to minimize their risk of being caught.
3. Improved sparrow search algorithm

3.1 K-means clustering algorithm
Sparrow search algorithm has few population roles, and it has heavy work in the initial optimization. At the same time, it depends on the randomly initialized population position. Therefore, K-means is used to cluster and differentiate the initial sparrow individual position, which makes the work orderly and increases the fault tolerance rate of the population.

K-means clustering algorithm is an iterative clustering algorithm. Its basic idea is that the closer the distance between two targets is, the greater the similarity is. The implementation process is as follows:

\[\text{Function K-means()}\]
\[\text{Initialize n elements and k cluster centers } \mu_i. \text{ Each cluster } A_i \text{ and } \mu_i \text{ corresponding}\]
\[\text{Repeat}\]
\[\text{For Every data point } X_i \text{ do}\]
\[\text{Assign to cluster } A_i \text{ with the smallest distance } d \text{ from the cluster center } \mu_i\]
\[\text{For Every cluster } A_i \text{ do}\]
\[\text{All cluster centers } \mu_i \text{ are updated by using the cluster center calculation formula}\]
\[\text{Calculate the sum of squares of errors } \sigma\]
\[\text{Until } \sigma \text{ does not change significantly or cluster center } \mu_i \text{ does not change}\]

The calculation formula of the distance \(d\) from the data point to the cluster center, the cluster center \(\mu\) and the sum of square error \(\sigma\) from the element to the cluster center is as follows:

\[d(x_i, \mu_j) = \sqrt{\sum_{j=1}^{n} (x_i - \mu_j)^2}\]  \hspace{1cm} (4)

\[\mu_j = \frac{1}{n} \sum_{x \epsilon A_j} x\]  \hspace{1cm} (5)

\[\sigma = \sum_{j=1}^{k} \sum_{x \epsilon A_i \neq A_j} |x_i - \mu_j|^2\]  \hspace{1cm} (6)

The introduction of the K-means clustering algorithm into the sparrow search algorithm, which achieves initialization to the sparrow population by cyclically using different clustering centers, reduces the dependence of the algorithm on initialization and the impact of initialization to the algorithm.

3.2 Levy flight mechanism
Levy distribution is a probability distribution proposed by French mathematician Levy in 1937, and levy flight is a random search method obeying levy distribution. During levy flight, there will be a larger range of motion and the direction of each change will be different. As shown in Figure 1, levy flight is active in a small area for a portion of time and then moves over a long distance.

\[\text{Figure 1. Levy flight diagram}\]
In the initial stage of sparrow search algorithm, the same high-quality position may be occupied by multiple roles, which leads to the decline of search efficiency, and the performance of the algorithm depends on the initialization stage, which greatly reduces the performance of the algorithm. It is also possible for the algorithm to fall into local optimum when facing complex problems with high dimensions. Therefore, in the initial stage, using the levy flight strategy to dynamically adjust the location of the discoverer, and using the search mechanism of long-distance and short-distance flight, to expand the search range, can effectively reduce the probability of the algorithm falling into the local optimal, thereby enhancing the search ability of the algorithm.

The location updates for Levy flights are as follows:

\[ X_{i}^{t+1} = X_{i}^{t} + l \oplus \text{levy}(\lambda) \]  

\( X_{i}^{t+1} \) represents the position of the \( i \)-th individual in the discoverer's generation \( t \), \( \oplus \) denotes point-to-point multiplication, \( l \) denotes the weight of the control step, \( l = 0.01 \cdot (X_{i}^{t} - X_p) \), and \( X_p \) is the current optimal solution. \( \text{levy}(\lambda) \) denotes the path that follows the levy distribution and satisfies: \( \text{levy} \sim u \sim t^{-\lambda}, 1 < \lambda \leq 3 \).

Because of the complexity of the levy distribution, the Mantegna algorithm is usually used to simulate it. The formula for calculating the step size is as follows:

\[ S = \frac{\mu}{|v|^{1/\lambda}} \]  

\[ \mu \sim N(0, \sigma_{\mu}^2) \]  

\[ v \sim N(0, \sigma_v^2) \]  

\[ \sigma_{\mu} = \left( \frac{\Gamma(1+\lambda) \sin(\pi \lambda/2)}{\Gamma(\frac{\lambda+1}{2}) \frac{\lambda^{-\lambda}}{2^{-\lambda}}} \right)^{1/\lambda} \]  

Where \( \sigma_v = 1 \), and \( \lambda \) is generally 1.5. So the final formula is as follows:

\[ X_{i}^{t+1} = X_{i}^{t} + l \oplus \text{levy}(\lambda) = X_{i}^{t} + lS \]  

### 3.3 Adaptive local search

To improve the ability of sparrow search algorithm to avoid local optimum, an adaptive local search is introduced. Adaptive local search is to add a parameter, which changes with the number of iterations, to the calculation of updating follower locations, so that it can be automatically adjusted without adding other variables. First, the followers take the location of the discoverer as the optimal location, then search for the neighborhood in the optimal location. Finally, the neighborhood values of the optimal solution are compared with the optimal solution one by one, and the individuals with good adaptability are selected as the optimal solution.

Larger adaptive coefficients are beneficial to global search, but the algorithm is less efficient; smaller adaptive coefficients are beneficial to accelerating the convergence of the algorithm, but are prone to falling into local optimum. Therefore, an adaptive coefficient \( \epsilon \) is added, which varies with the number of iterations, and its formula is as follows:

\[ \epsilon(t) = \frac{\text{iter}_{\text{max}} - t + 1}{\text{iter}_{\text{max}}} \]  

Where \( t \) represents the number of iterations. In the iteration update process, the search range of neighborhood will gradually reduce with the number of iterations, balancing the search speed and the search accuracy to achieve the purpose of adaptive search. The mathematical expression for updating the follower location with adaptive local search is as follows:

\[ X_{i}^{t+1} = (1 - \epsilon)X_{p}^{t+1} + \epsilon \cdot \text{Rand} \]  

Where \( X_{i}^{t+1} \) is the optimal position of followers in the \( t+1 \) iteration, \( \text{Rand} \) is a \( d \)-dimensional vector, and its vector element value is a random number on \([0, 1] \). In the initial stage of the algorithm, the smaller \( t \) is, the larger \( \epsilon \) is, the wider the neighborhood search range is, so that followers can find the optimal location in a larger range; when \( t \) is larger, the calculation method of updating sparrow location will make the algorithm fall into local optimum. At this time, \( 1 - \epsilon \) increases with the increase
of T, which balances the convergence characteristics of the algorithm.

3.4 Improved sparrow search algorithm process

The sparrow algorithm had few roles and was somewhat limited at the initialization stage. Therefore, K-means is used to cluster and differentiate sparrows’ individual locations, which reduces the impact of randomness and makes the population work efficiently. By introducing the Levy flight mechanism, the discoverer stage has a broader and more flexible search range, and the resulting solution is more valuable. Finally, in the follower stage, an adaptive local search strategy is added to make the search mechanism more detailed and improve the search accuracy. The algorithm flow is as follows:

**Improved sparrow search algorithm (KLSSA) process**

**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**: \( X_{best}f_g \)

1. \( T = 1; \)
2. Initialize the location of sparrow individuals by introducing K-means clustering
   (K should not be too large and increase with population size)

3. \( \text{While}(t < G) \)
4. According to the fitness value to find the best and worst sparrow individual position
5. \( R^2 = \text{rand}(1) \)
6. \( \text{For } i = 1:PD \)
7. Update the location of the discoverer according to the formula (7)
8. \( \text{End for} \)
9. \( \text{For } i = (PD+1):N \)
10. Update the location of the follower according to the formula (14)
11. \( \text{End for} \)
12. \( \text{For } l = 1:SD \)
13. Update the position of sparrows aware of danger
14. \( \text{End for} \)
15. Get the location of the new optimal individual
16. Update optimal location if new individual location is better than previous individual location
17. \( t = t + 1 \)
18. \( \text{End while} \)
19. Return: \( X_{best}f_g \)

4. Experimental results and analysis

4.1 Experimental environment and parameter settings

In order to verify the feasibility of the improved SSA algorithm, eight test functions are used to verify the optimization effect of the three strategies. Particle Swarm Optimization (PSO) and Grey Wolf Optimizer (GWO) are introduced for comparison experiments. Each algorithm runs on a Window10 64bit system with 8GB memory and Intel(R) Core(TM) i7-5500U CPU @ 2.40GHz. The population size of each algorithm is 100, the number of iterations is 200, and the number of cluster centers K=10. MATLAB R2018b is used for simulation. The benchmark functions are shown in Table 1. The first five functions are set to 30 dimensions, the unimodal function is \( F_1-F_4 \), the multimodal function is \( F_5-F_6 \), and the remaining two functions are different fixed-dimensional functions to optimize. Each algorithm runs 30 times independently, and the optimal value, average value and Standard Deviation of each algorithm are counted. The three indexes are used to test the searching ability, searching speed and stability of the algorithm. The results of algorithm optimization and comparison are shown in Table 2 and Figure 2.
Table 1. Test function table

| FUNCTION | DIMENSION | INTERVAL | OPTIMAL VALUE |
|----------|-----------|----------|---------------|
| $F_1(x) = \sum_{i=1}^{n} x_i^2$ | 30 | [-100,100] | 0 |
| $F_2(x) = \sum_{i=1}^{n} (x_i)^2$ | 30 | [-100,100] | 0 |
| $F_3(x) = \max_i[|x_i|, 1 \leq i \leq n]$ | 30 | [-100,100] | 0 |
| $F_4(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$ | 30 | [-30,30] | 0 |
| $F_5(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{|x_i|})$ | 30 | [-500,500] | -418.9829n |
| $F_6 = 418.9829n - \sum_{i=1}^{n} x_i \sin(\sqrt{|x_i|})$ | 4 | [-10,10] | 0 |
| $F_7(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_2^2 - x_4)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1)$ | 37 | [-65.536,65.536] | 0.998 |

Table 2. Comparison table for each algorithm

| FUNCTION | Algorithm | OPTIMAL VALUE | Average value | Standard Deviation |
|----------|-----------|---------------|---------------|--------------------|
| $F_1(x)$ | KLSSA | 0 | 0 | 0 |
| | CSSA | 0 | 0 | 0 |
| | SSA | 0 | 5.192E-251 | 0 |
| | GWO | 1.4406E-103 | 1.56961E-97 | 5.36616E-97 |
| | PSO | 3.8599E-12 | 3.0697E-11 | 2.7076E-11 |
| | KLSSA | 0 | 0 | 0 |
| | CSSA | 0 | 4.2858E-192 | 0 |
| $F_2(x)$ | SSA | 2.2250E-04 | 0.029431 | 0.070791 |
| | GWO | 22.2592 | 49.9596 | 18.1286 |
| | KLSSA | 0 | 0 | 0 |
| | CSSA | 0 | 7.7137E-210 | 0 |
| $F_3(x)$ | 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 |
| | KLSSA | 1.7599E-09 | 0.000132611 | 9.7825-05 |
| | CSSA | 0.00062694 | 0.0015114 | 0.0015114 |
| $F_4(x)$ | SSA | 0.998 | 0.000132611 | 9.7825-05 |
| | GWO | 45.8565 | 47.3855 | 0.89355 |
| FUNCTION | Algorithm | OPTIMAL VALUE | Average value | Standard Deviation |
|----------|-----------|---------------|---------------|--------------------|
| PSO      | -89.4507  | 234.8606      | 104.6002      |
| KLSSA    | -12533.8181 | -9328.842212 | 754.9426      |
| CSSA     | -9859.7543 | -8700.9989    | 615.6127      |
| SSA      | -9374.4498 | -8208.1416    | 501.1037      |
| GWO      | -9055.703  | -6452.354     | 1033.4608     |
| PSO      | -8405.1316 | -6176.2342    | 793.1086      |
| KLSSA    | 377.6629992 | 3017.783766  | 1100.432732   |
| CSSA     | 2529.3974  | 3661.9962     | 725.5963      |
| SSA      | 1254.61E-14 | 5.22137E-07  | 1.36526E-06   |
| GWO      | 2505.7803  | 7638.9387     | 1024.9673     |
| KLSSA    | 1254.61E-14 | 5.22137E-07  | 1.36526E-06   |
| CSSA     | 9.18632E-11 | 7.55843E-06  | 1.55751E-05   |
| SSA      | 2.2255E-11  | 1.02709E-06   | 2.51357E-06   |
| GWO      | 1.2267E-08  | 1.142301517   | 0.00022591    |
| PSO      | 0.001130521 | 0.026019705   | 0.022667333   |
| KLSSA    | 0.998      | 0.998         | 0             |
| 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        |

![Graphs](image-url)
4.2 Optimize performance analysis

From the data in Table 2, we can see that the KLSSA algorithm has better convergence and solution accuracy than the other four algorithms in unimodal and multimodal functions. Specifically, on the functions F1-3 and F8, KLSSA can find the optimal value with good stability. On the two functions F5-6, KLSSA algorithm has better optimization ability and higher accuracy than other algorithms.

From Figure 2, it can be seen that KLSSA algorithm has better convergence accuracy and speed than other algorithms, KLSSA has better convergence speed and accuracy on unimodal function, and KLSSA has stronger ability to resist local extremes on multimodal function. Because of the Levy flight and the adaptive weight search strategy, KLSSA makes the search method more detailed and extensive, and improves the global search ability of the algorithm, so that the algorithm can break through the local optimal limit to find a new global optimal solution. Overall, KLSSA algorithm has higher search accuracy and solving ability.

5. Concluding remarks

The Sparrow search algorithm is an efficient and intelligent optimization algorithm, but there are still some limitations when dealing with complex function optimization problems. Based on the optimization characteristics of sparrows, this paper puts forward corresponding strategies for improvement. Firstly, K-means is used to cluster sparrows to reduce the interference of randomly initialized populations. Secondly, the Levy flight mechanism is introduced in the discoverer stage to make its search more extensive and flexible, and to improve the global search of the algorithm. Then, in the follower stage, an adaptive local search strategy is added, which makes the follower search more detailed and improves the convergence and local search ability of the algorithm. Through the test on 10 standard test functions, the results show that ISSA algorithm can break through the limitation of local optimal solution, obtain higher accuracy, and has stronger global search ability than other improved sparrow search algorithm and classical intelligent optimization algorithm, which proves the
effectiveness of the improved strategy.

Acknowledgments
This work was supported in part by National Natural Science Foundation of China (No.61561024). And I would like to express my heartfelt thanks and respect to Mr. Ouyang, who inspires me to keep learning new knowledge. Then I would like to thank my senior brother Donglin Zhu, who gave me a lot of assistance in the completion of this paper. 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] Lv Xin, Mu Xiaodong, Zhang Jun, Wang Zhen. Chaotic sparrow search optimization algorithm [J/ol]. Journal of Beijing University of Aeronautics and Astronautics: 1-10 [2020-09-05]. UNK1.13700/j.bh.1001-5965.2020.0298.
[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] Chen Tianpei, Wang Yuhui, Wu Qingxian, Zhou Zeyu. Three-dimensional path planning based on fuzzy logic particle swarm optimization algorithm. Electro-Optics and Control, 2020, 27 (06): 1-5.
[5] Lei Zhang, Xiangtuo Zhuan, Marcello Vanali. Vibration Control Method of an Electromagnetic Isolation System Based on LQR and Coevolutionary NGA. 2020, 2020.
[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] Song Xiaoyu, Gao Minghui, Zhao Ming. Hybrid Artificial Bee Colony Algorithm with Adaptive Search Strategy. Computer Programs and Applications, 2019, 55 (22): 53-59 + 85.
[8] Yuan Yan, Cao Cuiwen. Improved Grey Wolf Algorithm and Its Application. Computer Engineering and Design, 2020, 41 (02): 513-521.
[9] Bu Guannan, Liu Jianhua, Jiang Lei, Zhang Dongyang. An Adaptive Grouping Ant Colony Algorithm [J/ol]. Computer Engineering and Applications: 1-10 [2020-08-04]. UNK1.2127.TP.20200617.1328.022.html.
[10] Guo Wenyan, Wang Yuan, Dai Fang, Liu Ting. Alternating sine and cosine algorithm based on elitist chaotic search strategy. Control and Decision, 2019,34 (08): 1654-1662.
[11] Huang Chenchen, Wei Xia, Huang Deqi, Ye Jiahao. Hybrid Frog Leap-Grey Wolf Optimization Algorithm for Solving High Dimensional Complex Functions. Control Theory and Applications, 2020, 37 (07): 1655-1666.
[12] Li Tianlai, Liu Fangai. Butterfly Optimization Localization Algorithm with Chaotic Mapping for Wireless Sensor Network. Computer Engineering and Design, 2019,40 (6): 1729-1733.
[13] Zhang Qiang, Cheng Hui, Ye Zhencheng, Zhang Guanghui. Improved Competitive Particle Swarm Optimization Algorithm and Its Application. Computer Engineering and Design, 2019,40 (2): 376-383.