An Improved Island Algorithm and Its Application in Model Optimization of Micro Soft Robot

MA Ji-ming\textsuperscript{a}, SHAN Shi-jiao\textsuperscript{b}, SU Ri-jian\textsuperscript{c}* \textsuperscript{,} WEN Xu-qin and Xu Hong-sheng
Zhengzhou University of Light Industry, Zhengzhou 450000, China

\textsuperscript{a}1988001@zzuli.edu.cn, \textsuperscript{b}184890396@qq.com, \textsuperscript{c}Srj@zzuli.edu.cn

Abstract. Aiming at the shortcomings of the original island algorithm (IA), which has a slow convergence speed and is prone to local optimality, the island algorithm with the characteristics of Levy flight (LevyIA) was proposed by introducing the Levy flight strategy, which replaced the position update method in the original algorithm and made use of the occasional long jump of Levy flight strategy to jump out of the local optimal solution. The simulation test of the improved algorithm is carried out with 6 test functions, and the experimental results show that the improved algorithm LevyIA can effectively solve the problems of slow convergence speed and local optimization of island algorithm. For the micro-soft robot model with multi-mode movement, IA and LevyIA algorithms were used to optimize the size of the robot and the appropriate magnetic field intensity needed to drive the robot to deform and move. Finally, the experimental data of swimming speed of the robot obtained by simulation shows that, among the three optimization results obtained by LevyIA algorithm and IA algorithm, LevyIA algorithm can make the robot swim faster when moving forward with minor perturbations.

1. Introduction
In recent years, in order to avoid side effects on patients during medical examinations and surgery, some small robots used for object transportation have entered the medical field. We focus on an unconstrained multi-modal motion micro soft robot. Such micro soft robot can enter the closed space in a non-invasive way, change its shape under the control of time-varying magnetic field, and generate different motion modes. However, the parameters of the robot cannot determine the optimal value, resulting in the failure to obtain fast and stable speed. Swarm intelligence algorithm can complete the optimization task of robot parameters. Many scholars, inspired by the laws of nature and the living behaviours of biological groups, have proposed lots of swarm intelligent algorithms. For example, genetic algorithm(GA)[1], bat algorithm(BA)[2], particle swarm optimization (PSO)[3]-[4], ant colony optimization (ACO)[5], firefly algorithm (FA)[6], and other algorithms[7]-[9] and so on. These algorithms can be effective methods to solve the problem of finding the optimal value, so they are widely used in the fields of engineering technology and scientific calculation.

Island algorithm (IA)[10] is a new meta-heuristic algorithm based on the phenomenon that the growing position of plants on islands is more and more concentrated at the highest point as the sea level rises and the island area shrinks. While the sea level rises and the island area shrink, the total number of plants remains unchanged, so as to find the global optimal position. However, it is found through simulation that with the increase of the number of iterations, the variation of island range decreases, which limits the search space and affects the number of elimination. Due to the interference
from these aspects, IA algorithm has the problem of slow convergence speed and easy to fall into the local optimization, which can lead to the algorithm performance degradation.

Based on the analysis of the principle and limitation of the original algorithm, this paper proposes a sea island algorithm with Levy’s flight characteristics (LevyIA). The unique random walk behaviour of Levy’s flight can effectively improve the diversity of the algorithm population, expand the search range, and make the algorithm jump out of the local optimal value. After generating new plants in a new range during the sea-level rise phase, the Levy flight strategy was used to replace the position movement mode in the original algorithm, which essentially improved the optimization performance of the algorithm. Six test functions in CEC2013 function set are used to simulate the proposed algorithm. The experimental results show that LevyIA algorithm can jump out of local optimum position in time, and its algorithm stability and robustness are better than the original island algorithm.

The LevyIA algorithm is used to optimize the micro soft robot model, and the optimal robot size and external magnetic field strength parameters are obtained. When the robot moves with small disturbance, the vertical disturbance is the smallest, and the swimming speed of the robot is the fastest. Compared with IA algorithm, the simulation results show that levyIA algorithm is better than IA algorithm, refer to § 5.3 for details.

2. Basic principles of island algorithm

2.1. Phase of elimination
The main task of the phase-out phase is to generate the number of plants that need to be phased out in this iteration according to the variation in the scope, in preparation for the iterative sea-level rise phase. When the range change is small, the number of elimination should be increased to expand the range change of the next iteration and accelerate the convergence of the algorithm. When the range variation is large, the number of elimination should be reduced to narrow the range variation of the next iteration. So the elimination function needs to satisfy two conditions:

<1> when the range change is 0, the number of elimination is the largest number of elimination.

<2> the elimination function should be a decreasing function. The elimination function is expressed as formula (1):

\[ f_0 \phi(h) = \left( A_n - A_1 \right) \cdot e^{-\frac{h}{A_n}} + A_1 \]  

In the formula: \( h \) is the range change, \( A_n \) is the maximum number of elimination, and \( A_1 \) is the minimum number of elimination.

2.2. Phase of sea level rise
After the number of elimination in the phase of elimination is generated, it enters the phase of sea level rise, which leads to the reduction of island range, while the growth range of plants that are not eliminated generates new island range and range change, which are respectively prepared for the balance stage and the phase of elimination in the next iteration. The formula for the sea-level rise phase and the range change formula are as follows (2), (3) and (4):

\[ X_{\max} = X_{\max} + \text{rand} \cdot (X_{\max}^{old} - X_{\max}), \]  
\[ X_{\min} = X_{\min} + \text{rand} \cdot (X_{\min}^{old} - X_{\min}), \]  
\[ h = \text{norm}(X_{\max}^{old} - X_{\min}^{old} - (X_{\max} - X_{\min})), \]

In the formula: \( X_{\max} \) and \( X_{\min} \) respectively represent the maximum and minimum values of each dimension in the current iteration; \( X_{\max}^{old} \) and \( X_{\min}^{old} \) represent the maximum and minimum values of each dimension in the last iteration, respectively.

2.3. Phase of equilibrium
The equilibrium stage is to produce A new plant in the new range at the same time of sea level rise, to replace the worst plant in the population, so as to maintain the population total. Then, according to
formula (5), move each new plant to the globally optimal position,

\[ x(j,:) = x(j,:) + 2 \cdot \text{rand}(1, D) \cdot (x(1,:) - x(j,:)) \]  

(5)

In the formula: \( x(j,:) \) is the position of the \( j \)th new plant in the new range; \( x(1,:) \) is the globally optimal position; \( D \) is the dimension; 2 is the parameter.

After the completion of the moving operation, the plant enters the evaluation stage. If the plant is superior to the globally optimal plant, the position and fitness values of the two are exchanged.

The flow chart of the basic island algorithm is shown in figure 1. \( \text{fit}(1) \) represents the optimal value of position.

**Figure 1. Island algorithm flow chart**

3. An island algorithm with Levy's flight characteristics

3.1. Levy flight characteristics

The Levy distribution is a probability distribution[11]-[12]. In recent years, some scholars[13-18] have found that Levy distribution also exists in nature. Studies have shown that in the process of searching for food in an uncertain environment, animals follow Levy's flight search strategy, and Levy's flight follows Levy's distribution law.

Levy flight is a random step size that obeys the Levy distribution. The step size \( s \) to calculate the random distribution of Levy is usually realized by Mantegna algorithm[19]. The step size formula is:

\[ s = \alpha \times \frac{u}{|v|^{\beta}} \]  

(6)

Among them: \( \beta=1.5 \), \( \alpha \) is the step factor. \( u, v \) obey normal distribution:

\[ u = \text{rand}(n,1) \times \sigma \]  

(7)

\[ v = \text{rand}(n,1) \]  

(8)
\[ \sigma = \left( \frac{\Gamma(1 + \beta) \times \sin(\pi \times \beta / 2)}{\Gamma((1 + \beta) / 2) \times \beta \times 2^{(\beta-1)/2}} \right)^{1/\beta}, \]  
\tag{9}

Where \( n \) is the dimension of the solution, and the gamma function is:
\[ \Gamma(b) = \int_{0}^{\infty} \exp(-t) \times t^{b-1} d(t) \]  
\tag{10}

### 3.2. Levy’s flight characteristics optimize the island algorithm

Inspired by Levy flight strategy, the strategy are adopted to simulate the island plant growth behaviour, make full use of the properties of random walk, in the process of plant growth position focused island peak, Levy flight characteristics of long jump is used to effectively avoid plant individual into local optimal value, increase the search space, improve the effect of optimization in the high dimension space, optimize the island from the essence the performance of the algorithm. The improved algorithm USES the following formula to replace the position update formula in the original algorithm (see formula (5)):
\[ x(j,:) = x(1,:) + 2 \cdot \text{rand}(1,D) \cdot (x(1,:) - x(j,:)) + (x(1,:) - x(j,:)) \otimes \text{Levy}(\lambda) \]  
\tag{11}

In the formula (11):
- \( x(j,:) \) is the position of the jth new plant produced in the new range;
- \( x(1,:) \) is the globally optimal position;
- \( \text{Levy}(\lambda) \) is the random disturbance vector whose step size \( s \) follows the distribution of Levy;
- \( \lambda \) is the scale parameter \( 1 < \lambda < 3 \);
- \( \otimes \) is for vector operations.

The island algorithm with Levy’s flight characteristics is as follows:

**Step 1:** Initialization parameter: set \( D \) as the dimension of the island, \( E \) as the maximum evaluation times, \( X_{\text{min}} \) and \( X_{\text{max}} \) are the island range, \( h \) is the range variation, \( N \) is the total number of plants, The growing position of the plant is \( x_i = (i=1,2,...,N) \), \( A_{\text{max}} \) and \( A_{\text{min}} \) are the maximum and minimum values of the elimination number.

**Step 2:** Phase of elimination, the number of elimination is generated according to formula (1).

**Step 3:** In the phase of sea level rise, a new island range is generated. In order to avoid falling into local optimization, the new range is extended according to formula (2) and formula (3), and the change of island range \( h \) is generated according to formula (4).

**Step 4:** In the equilibrium stage, new plants are generated within the new island range according to the number of culled plants, and the worst plants in the population are eliminated.

**Step 5:** According to formula (11), move each newly produced plant toward the optimal plant position, and find the value of applicability of the new plant. If the new plant is better than the optimal plant, it will be exchanged, and then all plants will be sorted.

**Step 6:** If the termination condition is met, jump out of the iteration, terminate the operation, and output the result. Otherwise, proceed to step 2 and the next iteration begins.

### 4. Algorithm analysis and comparison

#### 4.1. The experiment design

Based on literature [10], the algorithm is not suitable for island has many sharp peak, peak and the surrounding area is relatively flat function. Therefore, in order to verify the performance of LIA algorithm, this paper selects the following 6 test functions for simulation test. The test function Table is shown in Table 1, \( f_1 \) is unimodal function, used to test the accuracy of the algorithm; \( f_2-f_5 \) is a multimodal function; \( f_6 \) is the combination function. The experimental environment is in Windows10 system, the experimental tool is Matlab R2015a software, CPU is i5-3470 3.20 GHz, RAM is 4 GB.
The parameters of IA algorithm and LevyIA algorithm are set as: the number size of plant population N=100; the maximum number and the minimum number of elimination are 80 and 20, respectively. The change in the initial range is 1.

For each function, in order to maintain the accuracy of the experimental results, the two algorithms were independently run on the search dimension of 25, 10 and 30 for 30 times respectively, and the calculated results were analyzed to compare the performance of the algorithm.

Table 1. Test function

| Function | The name of the function                   | The Optimal value |
|----------|-------------------------------------------|-------------------|
| f_1      | Sphere Function                           | -1400             |
| f_2      | Rotated Rastrigin’s Function              | -300              |
| f_3      | Non-Continuous Rotated Rastrigin’s Function| -200              |
| f_4      | Lunacek Bi_Rastrigin Function             | 300               |
| f_5      | Rotated Lunacek Bi Rastrigin Function     | 400               |
| f_6      | Composition Function 1 (n=5,Rotated)      | 700               |

4.2. Comparison of experimental results

The optimal value, the worst value, the median value, the mean value and the standard deviation obtained after 30 evaluations are used as the measurement indexes of the algorithm's accuracy and robustness. The running results are shown in Table 2, Figure 2 and Figure 3. If the result is less than 10^{-8}, the value is considered to be 0.

It can be seen from Table 2, Figure 2 and Figure 3 that the results of the test functions f_2, f_3, f_4, and f_6 in 5 dimensions are that the results of the LevyIA algorithm are superior to the IA algorithm, and the results of f_1 is close. The test result of IA algorithm in f_5 is better than LevyIA algorithm; The test function f_3, f_5 in the 10 dimensions result is that the LevyIA algorithm is better than the IA algorithm, and the results of f_1, f_3, f_4 are close; The test function f_1, f_3, f_5 in the 30-dimensional result is that the LevyIA algorithm is better than the IA algorithm. The results of IA algorithm at f_2, f_4, f_6 are better than LevyIA algorithm. In summary, the computational accuracy and robustness of the LevyIA algorithm is generally superior to the IA algorithm. The island algorithm with Levy flight characteristics is better than the island algorithm. Y1 to Y5 represent standard deviation, average value, median value, worst value and optimal value, respectively.

Table 2. Performance comparison of 6 functions for 5 dimensions

| Function | The name of the algorithm | The optimal value | The worst value | The median | The average | The standard deviation |
|----------|--------------------------|-------------------|-----------------|------------|------------|------------------------|
| f_1      | IA                       | 0                 | 0               | 0          | 0          | 0                      |
|          | LIA                      | 0.9950            | 10.9445         | 3.9798     | 5.0069     | 2.8315                 |
| f_2      | IA                       | 0.8396            | 4.8844          | 2.4435     | 2.4859     | 0.8847                 |
|          | LIA                      | 1.3905            | 16.5648         | 8.2777     | 8.5198     | 4.7011                 |
| f_3      | IA                       | 1.6574            | 10.4315         | 6.6886     | 6.9412     | 3.2796                 |
|          | LIA                      | 0.4557            | 10.1243         | 6.6544     | 6.4199     | 2.4359                 |
| f_4      | IA                       | 0.5912            | 13.5219         | 7.8077     | 6.9412     | 3.2796                 |
|          | LIA                      | 6.3891            | 9.5320          | 7.8219     | 7.7960     | 0.7114                 |
| f_5      | IA                       | 0                 | 400             | 300        | 270.9677   | 90.1611                |
|          | LIA                      | 100               | 300             | 300        | 267.7419   | 74.7757                |
5. Optimization of miniature soft robot model

5.1. Model is introduced

An unconstrained multi-modal micro-soft robot is placed in the middle of three pairs of orthogonal coils (figure 4). Its internal cavity size is 65 mm×65 mm×65 mm[20]. A specific current is introduced into the coil, which is superimposed on each other by each coil to generate uniform magnetic field intensity $B$ in the central field area. By changing the current, you can change the magnetic field strength $B$.

The principle is that the small soft robot is preprogrammed as a single-wavelength harmonic magnetic profile, which can change its shape under the control of a time-varying magnetic field, and can generate different motion modes according to the different terrain. Because the base materials for the small soft robots contain NdFeB magnetic particles, the high remanent magnetization of NdFeB particles allows us to generate effective magnetic actuation for the robots, and their high magnetic coercive force ensures that our robots remain magnetized during the magnetization. The shape of the robot is shown in figure 5, where $L$, $w$ and $h$ represent the length, width and height of the robot respectively, and the unit is m. $\beta$ is the phase shift generated by the robot during the magnetization process. The magnetization profile $M$ along the length $s$ of the robot is described in as:

$$M(s) = m \begin{bmatrix} \cos(\omega_s s + \beta) \\ \sin(\omega_s s + \beta) \\ 0 \end{bmatrix},$$

(12)

Where $m$ and $\omega_s$ respectively represent the magnetic moment and spatial angular frequency of $M(s)$, $\omega_s=2\pi/L$. 

![Figure 4. Coil device diagram](image4.png)

![Figure 5. Robot shape](image5.png)
5.2. Model calculation

How does the robot change its posture under the control of the external magnetic fields, \( \mathbf{B} = [B_x, B_y, B_z]^T \)? This problem can be described by quasi-static analysis. The interaction between \( \mathbf{B} \) and \( \mathbf{M}(s) \) makes the robot produce the desired deflection. In Figure 6, the deflection can be described by the Euler-Bernoulli equation for beams, namely, the bending moment \( M_s = EI \frac{\partial^2 \theta}{\partial s^2} \). Rotation deviation \( \theta(s) \) along the axis of the robot can be expressed as:

\[
\begin{align*}
\tau_m A &= EI \frac{\partial^2 \theta}{\partial s^2}(s) \\
\tau_m &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \{RM(s) \times \mathbf{B}\} \\
R &= \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\
\sin \theta & \cos \theta & 0 \\
0 & 0 & 1 \end{bmatrix}
\end{align*}
\]

Where \( \tau_m \) represents the driving magnetic moment; \( A = h w \) is the cross-sectional area of the robot; \( l = h w / 12 \) is the quadratic moment of the area; \( E \) is Young's modulus; \( R \) is the standard z-axis rotation matrix; \( \mathbf{B} = [B_x, B_y, B_z]^T \).

For small deflection motions similar to undulate swimming, the rotation matrix of the robot can be reduced to a 3×3 identity matrix. The length \( s \) of the robot in formula (13) can be described as \( s \approx X'(X' \text{ along the main frame}), \theta \) can be described as \( \theta = \frac{\partial Y'}{\partial X'} \), \( Y' \) is the vertical deflection.

Formula (16) can be obtained through the above simplification:

\[
\begin{align*}
-\begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \{M(s) \times \mathbf{B}\} A &= EI \frac{\partial^3 Y'}{\partial X'^3} \\
\end{align*}
\]

The shape of the robot under small deflection can be described by a linear combination of the two main shapes. When the direction of \( \mathbf{B} \) is parallel to \( [\cos \beta \ \sin \beta \ 0]^T \) or \( [-\sin \beta \ \cos \beta \ 0]^T \), these two standard primary shapes can be performed. To obtain the vertical perturbation of the two main shapes (see figure 7), \( \mathbf{B} \) is substituted into formula (16).

\[
\begin{align*}
Y'_i(X') &= \frac{mALB}{8\pi^3EI} \cos \left( \frac{2\pi}{L} X' \right) \\
\end{align*}
\]

When \( \mathbf{B} = [\cos \beta \ \sin \beta \ 0]^T \), the vertical disturbance of the robot can be expressed as:

\[
\begin{align*}
Y'_i(X') &= \frac{mALB}{8\pi^3EI} \cos \left( \frac{2\pi}{L} X' \right) \\
\end{align*}
\]

When \( \mathbf{B} = [-\sin \beta \ \cos \beta \ 0]^T \), the vertical disturbance of the robot can be expressed as:
In order to make the robot swim in a continuous undulate motion, a rotating magnetic field needs to be applied in the $xy$ plane. The time-varying magnetic induction intensity $B(t)$ applied to the three coils can be expressed as:

$$B(t) = \begin{bmatrix} B_x(t) \\ B_y(t) \\ B_z(t) \end{bmatrix} = \begin{bmatrix} B \cos \alpha \cos(2\pi ft) \\ B \cos \alpha \sin(2\pi ft) \\ 0 \end{bmatrix}$$

(19)

Where $f$ represents the frequency of the magnetic field, $t$ represents the time, $\alpha$ represents the steering angle of the robot.

According to the content of Section 3.2, the effective undulating swimming gait of the robot can be expressed as:

$$Y'(X', t) = \left( \frac{mAL}{8\pi^2 EI} R_i \right) \left[ \cos \left( \frac{2\pi}{L} X' \right) \cos(2\pi ft) + \sin \left( \frac{2\pi}{L} X' \right) \sin(2\pi ft) \right]$$

(20)

Where $R_i$ represents the ideal Bode transfer function, which is not sensitive to gain changes of in the system. The relationship between $R_i$, frequency $f$ and the basic natural frequency $\omega_c$ of the robot was described in references [20]:

$$R_i = \begin{cases} 1, f < \frac{\omega_c}{2\pi} \\ \left( \frac{\omega_c}{2\pi f} \right)^2, f \geq \frac{\omega_c}{2\pi} \end{cases}$$

(21)

Finally, the undulating swimming speed model can be represented as:

$$V_{\text{swim}} = \frac{2\pi^2}{L} \left( \frac{mALB}{8\pi^2 EI} R_i \right)^2 f$$

(22)

5.3. experimental design

From Section 5.2, we can see the robot's undulating swimming speed model. Next, we substitute $A=hw$ and $I=h^3w/12$ into the undulating swimming speed model:

$$V_{\text{swim}} = \frac{L^5}{2} \left( \frac{3\cdot mB}{\pi^2 Eh^2} R_i \right)^2 f$$

(23)

It can be seen that the fluctuation speed of the robot has nothing to do with the width of the robot, so the width of the robot can be set to be consistent with the width of the original robot. The length and height of the robot are within 0.005 meters. In order to find the size parameters of the robot in the model, the speed formula is optimized by Levy's improved island algorithm. To find the maximum speed, the fitness function can be defined as follows:

$$f = e^{-\frac{L^5}{2} \left( \frac{3\cdot mB}{\pi^2 Eh^2} R_i \right)^2 f} - 1$$

(24)

Where, $\mu > 0$ is a constant coefficient. When the fitness function is infinitely close to its minimum value, $u^{-\frac{L^5}{2} \left( \frac{3\cdot mB}{\pi^2 Eh^2} R_i \right)^2 f}$ tends to zero, and $V_{\text{swim}} = \frac{L^5}{2} \left( \frac{3\cdot mB}{\pi^2 Eh^2} R_i \right)^2 f$ is the largest.

In order to find the magnetic field strength in the model and the size of the robot and other parameters, the LevyIA algorithm is used to optimize the fitness function (23), and then the swimming speed of the robot is solved. The basic parameters of the robot are in table 3:
Table 3. The basic parameters of the robot

| The parameter name       | Parameter values |
|--------------------------|------------------|
| Young's modulus E        | 84500 pa         |
| Magnetization profile m  | 62000 A/m        |
| Density ρ                | 0.001T           |
| Rotation frequency f     | 61 Hz            |

Table 4. The simulation result

| Parameter algorithm      | Primitive robot | IA   | LevyIA |
|--------------------------|-----------------|------|--------|
| h                        | 0.000185        | 0.00009 | 0.00005 |
| w                        | 0.0015          | 0.0015 | 0.0015 |
| L                        | 0.0037          | 0.0048 | 0.0031 |
| \( V_{swim} \)            | 8.9812e-04      | 0.0589 | 0.0695 |

5.4. Calculation results and analysis

The LevyIA algorithm fusion algorithm solves the model of the miniature soft robot under small disturbance, and compares it with the original robot size and length and the parameters calculated by the IA algorithm. The results are shown in table 4.

Figure 8. Fitness value curve

The height h of the robot obtained by LevyIA algorithm is 0.00005 meters, the width is unchanged, w is 0.0015 meters, and the length L is 0.0031 meters. Substituting the obtained parameters into formula (22), the obtained swimming speed is 0.0695 m/s. The result is significantly faster than the swimming speed obtained by the original robot and the IA algorithm.

6. Conclusion

LevyIA algorithm is proposed in this paper. Based on the IA algorithm, Levy flight mechanism is introduced to disturb the population position, which can effectively get rid of the constraint of local extreme value, and make the algorithm find a better position quickly, thus accelerating the convergence rate. Six basic test functions were used to conduct comparison experiments in 2, 10 and 30 dimensions respectively, and the convergence speed and robustness of LevyIA algorithm was verified to be better than that of IA algorithm.

Finally, LevyIA algorithm is applied to the optimization of micro-soft robot model. By comparing with the simulation results obtained with the original robot and the IA algorithm, the experimental data swimming speed \( V_{swim} \) shows that the LevyIA algorithm is superior to the IA algorithm in terms of accuracy and optimization parameters. In further, we will consider the search for new combined swarm intelligence algorithms, such as the combination with bat algorithms[21]-[23]; and the application of swarm intelligence technology in multi-objective solution and optimization problems[24]-[25], such as micro robot motion control, magnetic fluid hyperthermia and big data dimensionality reduction.
Acknowledgments
This work is supported by the National Natural Science Foundation of China (61773018), and the Key Technologies R&D Program of Henan Province China (212102210546). Thanks to the team members for their mutual help and joint experimentation, everyone can work together to complete this paper.

References
[1] HONDA M. Application of genetic algorithms to modelings of fusion plasma physics[J].
[2] Iztok Fister jr., Iztok Fister, Xin-She Yang, et al. Bat algorithm: Recent advances[C].
International Symposium on Computational Intelligence and Informatics. IEEE, 2014, PP: 163-167.
[3] Y. AiJia, T. Zhuo and L. KenLi et al., Estimating parameters of muskingum model using an adaptive hybrid PSO algorithm, Int. J. Pattern Recognit. Artif. Intell. 28(1) (2014) 178–215.
[4] QI Yichao, CHEN Wei, MU Chunyuan, ZHU Ninghua. Laser temperature control system based on particle swarm self-tuning PID algorithm[J], Laser Technology, 2019, 43(5): 650-654.
[5] Greco, A. Pluchino, F. Cannizzaro. An improved ant colony optimization algorithm and its applications to limit analysis of frame structures[J], Engineering Optimization, 2019, 51(11): 1867-1883.
[6] TIGHZERT L, FONLUPT C, MENDIL B. A set of new compact firefly algorithms[J]. Swarm and evolutionary computation, 2018, 40: 92-115.
[7] Chen Xu, Li Kangji, Tianfield Huaglory. Self-adaptive differential artificial bee colony algorithm for global optimization problems[J], Swarm and Evolutionary Computation, 2019, 45: 70-91.
[8] JIANG X Y, LI S. BAS:Beetle antennae search algorithm for optimization problems[J]. International Journal of Robotics and Control, 2018.1 (1):1.
[9] Ma jiming, Chen haoyang, Zhang song. Based on chaos perturbation mechanism, longhorn whisker search algorithm and its application in image enhancement[J]. Journal of light industry, 2019, 34(3):68-76.
[10] Ma jiming, Zhang song, Su rijian, Zhang guoliang, Cheng haoyang, Shan shijiao. A meta-heuristic algorithm: island algorithm[J]. Journal of ZhengZhou University (Engineering Science), 2019, 40(04):54-60.
[11] Zhang qiantu, Fang liqing,Zhaoyulong. An optimization algorithm for Gemini drosophila with Levy’s flight characteristics[J]. Computer application, 2015, 35(05):1348-1352.
[12] REYNOLDS A M, SMITH A D, REYNOLDS D R, et al. Honeybees perform optimal scale-free searching flights when attempting to locate a food source[J]. The Journal of Experimental Biology, 2007, 210(21) : 3763-3770.
[13] REYNOLDS A M, FRYE M A. Free-flight odor tracking in drosophila is consistent with an optimal intermittent scale-free search[J]. PLoS One, 2007, 2(4) : 1-9.
[14] VISWANATHAN G M, AFANASYEV V, BULDYREV SV, et al. Lévy flight search patterns of wandering albatrosses[J]. Nature, 1996, 381: 413-415.
[15] EDMARDS A M, PHILLIPS R A, WATKINS N W, et al. Re- visiting Lévy flight search patterns of wandering albatrosses,bumblebees and deer[J]. Nature, 2007, 449: 1044-1048.
[16] Liu changping, Ye Chunming. Bat algorithm with Levy’s flight characteristics[J]. Journal of intelligent systems, 2013, 8(3) : 240-246.
[17] Xu kun, Cheng zhijun, Yanxueqin. Improved ant colony algorithm based on levy flight to solve the TSP problem[J]. Computer engineering and design, 2019, 40(01):245-249.
[18] Liu xiaoming, Shen mingyu, Houzhengfeng. Fuzzy clustering algorithm of firefly based on Levy flight[J]. Computer application, 2019, 39(11):3257-3262.
[19] Mantegna RN. Fast, accurate algorithm for numerical simulation of Levys Table stochastic processes[J]. Physical Review E, 1994,49: 4677-4683.
[20] Hu Wenqi, Lum GuoZhan, Mastrangeli Massimo, Sitti Metin. Small-scale soft-bodied robot with multimodal locomotion[J]. Nature, 2018, 554(7690).
[21] Munshi, Amr A.; Mohamed, Yasser A.-R.I. Comparisons among Bat algorithms with various objective functions on grouping photovoltaic power patterns[J], Solar Energy, 2017, 144: 254-266.
[22] Nehal Elshaboury, Tarek Attia, Mohamed Marzouk. Application of Evolutionary Optimization Algorithms for Rehabilitation of Water. Distribution Networks. 2020, 146(7).
[23] Yu jinping, Wang wei, Wu guangfu, Liang wen. Multi-objective particle swarm optimization algorithm based on game mechanism[J]. Computer engineering and design, 2020, 41(04):964-971.
[24] Wang shuohe, Liu xu, Li suchen, Zhang guoju. LoRa parameter matching optimization based on multi-objective genetic algorithm[J]. Computer engineering and science, 2020, 42(03):434-440.
[25] Cobianchi M, Guerrini A, Avolio M, et al. Experimental determination of the frequency and field dependence of Specific Loss Power in Magnetic Fluid Hyperthermia[J]. Journal of Magnetism & Magnetic Materials, 2017, 446:54-67.