A New Bionic Swarm Intelligence Optimization: Construction and Application of Modified Moth-Flame Optimization Algorithm

Ge-ning XU, Shuang-yun HUANG*, Qi-song QI and Zhong-yuan KANG

College of Mechanical Engineering, Taiyuan University of Science and Technology, Taiyuan, 030024

*Corresponding author

Keywords: Moth-Flame optimization algorithm, Modified, Adaptive mechanism, Study.

Abstract. The Moth-flame optimization algorithm is a new bionic swarm intelligence algorithm. But the moth’s behavior has a large number of random states and need to repeatedly test in the algorithm, which takes longer. In this paper, the basic principle of the Moth-flame algorithm is analyzed deeply, and proposed a modified Moth-flame algorithm. Its core is to improve and optimize the adaptive mechanism for the number of flames, and to change the flame adaptive mechanism along the straight line to decrease along the curve, so as to improve the convergence speed of the adaptive flame number; Given the ability of "study" to the moths when moths update position, that all moths update the position with reference to the best flame, so as to improve the search accuracy. By testing 8 classical test functions and 1 engineering example, it is proved that the modified Moth-flame algorithm has the advantages of faster convergence speed, higher search precision and avoiding local optima. The significant computational efficiency and precision of the improved moth-flame algorithm can be used to improve the ability to solve practical engineering problems.

Introduction

In the long-term evolution of animals, after a long period of natural selection, created a lot of wonderful nature of the group phenomenon, these phenomena for the solutions to the problem of human thinking has brought a lot of inspiration and encouragement. For example, the genetic algorithm (GA) mentioned in reference [1] is a method to imitate the natural biological evolutionary way to seek the optimal value; In reference [2], by simulating the firefly luminescence behavior, put forward the glowworm swarm optimization (GSO); In reference [3], a fruit fly algorithm (FOA) was proposed by foraging behavior of drosophila. In reference [4], the fish swarm algorithm (FSA) is proposed by simulating the foraging and clustering behavior of fish. In reference [5], by simulating the tight organization system of wolves and the exquisite cooperative hunting method, a new swarm algorithm called wolf pack algorithm (WPA) is proposed. In reference [6], the population behavior of bird is studied, and Particle swarm optimization (PSO) is proposed. Even some scholars have put forward the idea of artificial intelligence algorithm (ABC), bat algorithm (BA), dolphin echo localization algorithm (DE) and so on.

The Moth-Flame Optimization (MFO) is a novel bionic swarm intelligence algorithm proposed by Seyedali Mirjalili in 2015. The algorithm is the imitation of nature moth lateral positioning navigation mechanism, which is based on stochastic optimization algorithm based on population search. This paper analyzes the principle of bionics algorithm and moths to flame, from the mathematical point of view to improve the algorithm. Through the 8 classic test functions and 1 engineering examples of the test, verify the feasibility of the improved algorithm of flame moth and effectiveness.

Moth-Flame Optimization Algorithms

Algorithm Bionics Principle
There are over 160,000 various species of this insect in nature, and they will follow the phenomenon of "moths dart into the fire". Moth is a nocturnal insect, in the long-term natural selection evolution, the moth flying in the night and the moon light to maintain an angle for
horizontal positioning and navigation, as shown in Figure 1. Using this mechanism, moths can maintain long straight-line flight. In this mechanism, the most critical factor is the distance between the moth and the moon is "far." When the moth sees a light source, it tries to maintain a similar angle to the light source to fly in a straight line. As the light source is closer to the moon, the moth continues to maintain a similar angle that can lead to deadly spiral flight Path, as shown in Figure 2, they final convergence to the light source.

Figure 1. Transverse orientation. Figure 2. Spiral flying path around close light sources.

Mathematical Description and Analysis of the Algorithm

Moth-Flame Optimization (MFO) is a novel bionic swarm intelligence algorithm proposed by Seyedali Mirjalili in 2015. the algorithm is inspired by the nature of the moth lateral positioning navigation mechanism, and the mathematical model of spiral flight update the position of moths and flame, and ultimately converge to the flame position of new bionic swarm intelligence algorithm.

In the moth-flame optimization algorithm, the candidate solution and the problem variable are the spatial position of the moth, and the matrix (1) is used to represent and the matrix (2) to store the corresponding fitness of moths.

\[
M = \begin{bmatrix}
  m_{1,1} & m_{1,2} & \cdots & \cdots & m_{1,d} \\
  m_{2,1} & m_{2,2} & \cdots & \cdots & m_{2,d} \\
  \vdots & \vdots & \vdots & \vdots & \vdots \\
  m_{n,1} & m_{n,2} & \cdots & \cdots & m_{n,d}
\end{bmatrix}
\]

(1)

\[
OM = [OM_1, OM_2 \cdots OM_n]^T
\]

(2)

Another key component of the algorithm is the flame, represented by a matrix (3) similar to the moth-matrix:

\[
F = \begin{bmatrix}
  F_{1,1} & F_{1,2} & \cdots & \cdots & F_{1,d} \\
  F_{2,1} & F_{2,2} & \cdots & \cdots & F_{2,d} \\
  \vdots & \vdots & \vdots & \vdots & \vdots \\
  F_{n,1} & F_{n,2} & \cdots & \cdots & F_{n,d}
\end{bmatrix}
\]

(3)

And use the matrix (4) to store the corresponding flame fitness value:

\[
OF = [OF_1, OF_2 \cdots OF_n]^T
\]

(4)

where \( n \) is the number of moths and \( d \) is the number of variables (dimension).

It should be noted here that moths and flames are both solutions. The difference between them is the way treated and updated in iteration. The moths are actual search agents that move around the search space, whereas flames are the best position of moths that obtains so far. Therefore, each moth searches around a flame and updates it in case of finding a better solution. (MFO) can be defined as Eq. (5):
\[(MFO) = (I, P, T)\]  

The function \(I\) have to generate initial solutions and calculate the objective function values. The \(P\) function, which is the main function, moves the moths around the search space. This function received the matrix of \(M\) and returns its updated one eventually. The \(T\) function returns true if the termination criterion is satisfied and false if the termination criterion is not satisfied.

As mentioned above the inspiration of this algorithm is the transverse orientation. In order to mathematically model this behavior, the position of each moth is updated with respect to a flame using the following equation:

\[M_i = S(M_i, F_j)\]  

where \(M_i\) indicates the \(i\)-th moth, \(F_j\) indicate the \(j\)-th flame, and \(S\) is the spiral function.

A logarithmic spiral is chosen as the main update mechanism of moths in this paper. However, any types of spiral can be utilized here subject to the following conditions:

1) Spiral’s initial point should start from the moth;
2) Spiral’s final point should be the position of the flame;
3) Fluctuation of the range of spiral should not exceed from the search space.

Considering these points, a logarithmic spiral is defined for the MFO algorithm as follows:

\[S(M_i, F_j) = D_i \cdot e^{bt} \cdot \cos(2\pi t) + F_j\]  

where \(D_i\) indicates the distance of the \(i\)-th moth for the \(j\)-th flame, \(b\) is a constant for defining the shape of the logarithmic spiral, and \(t\) is a random number in \([-1, 1]\). \(D\) is calculated as follows:

\[D_i = |F_j - M_i|\]

In order to ensure the fast convergence speed of MFO algorithm, an adaptive mechanism is proposed for the number of flames. That is the number of flames is decreased over the course of iterations. The following formula is utilized in this regard:

\[\text{flame no} = \text{round}(N - l \times \frac{N-1}{T})\]

where \(l\) is the current number of iteration, \(N\) is the maximum number of flames, and \(T\) indicates the maximum number of iterations.

Eq. (7) is where the spiral flying path of moths is simulated. As may be seen in this equation, the next position of a moth is defined with respect to a flame. The \(t\) parameter in the spiral equation defines how much the next position of the moth should be close to the flame (\(t = -1\) is the closest position to the flame, while \(t = 1\) shows the farthest). Therefore, a hyper ellipse can be assumed around the flame in all directions and the next position of the moth would be within this space. The spiral movement is the main component of the proposed method because it dictates how the moths update their positions around flames. The spiral equation allows a moth to fly “round” a flame and not necessarily in the space between them. Therefore, the exploration and exploitation of the search space can be guaranteed.

To sum up, the MFO algorithm optimization steps can be summarized as:

Step 1. Set the parameters and the range to be optimized and algorithm termination conditions;
Step 2. Determining the fitness function of MFO algorithm. Fitness function is an important index to describe the quality of population;
Step 3. Initialize the algorithm parameters. Set the maximum number of iterations \(T\), moth population size \(n\), the search space dimension \(d\) and the maximum number of flame \(N\), the current number of iterations \(l = 0\);
Step 4. Individual fitness value of the moth is calculated and the individual position of the current best moth is found and saved as the flame fitness value matrix. Judging whether the algorithm iteration termination condition is satisfied, if it is, then go to step 8, otherwise execute step 5;
Step 5. Iterative process. The flame number is updated by Eq. (9); the distance between flame and moth is calculated, and the moth-flame position is updated by Eq. (7);
Step 6. The calculation of fitness value of individual moths, the Eq. (2), Eq. (4) preservation of moths and flame space respectively;
Step 7. To finds the current best moth individual spatial location. If the current position is better than the previous position, the current flame position is retained as optimal. Judging whether the iterative termination condition of the algorithm is satisfied, if yes, then go to step 8; otherwise, set \( l = l + 1 \); and perform steps 5 to 8;
Step 8. The optimal individual value and the global extreme value are output. That is the final position of the flame and the corresponding fitness value. The algorithm ends.

![Diagram](image)

**Figure 3. Steps of MFO.**

**Improvement of the Algorithm**

In the 2.2 algorithm, the algorithm proposed by the algorithm to ensure that the MFO algorithm to achieve a faster convergence speed, the number of adaptive flame renewal mechanism formula by Eq. (8), by the formula can know the flame number update mechanism is linear decline, this bold assumption, if the number of new flame mechanism along the curve decreased whether the convergence speed will be faster? Assume that the adaptive flame number update mechanism is as follows:

\[
\text{flame } \text{no} = \text{round}(T / (l + T / N))
\]  

(10)

At the same time draw function image of the two update mechanism, suppose the maximum number of flame N is 50 and the maximum number of iterations T is 1000. As can be seen from the image in Figure 4, the improved adaptive convergence curve of flame number will be which leads to faster convergence speed. Therefore, it can be predicted that the improved MFO algorithm will have a faster convergence speed.
In addition, "study" is a universal phenomenon in nature, within the same population, between different species, there is learning phenomenon, human beings have higher than the general species of study ability, which is the main reason humans can dominate the world\cite{8}. Similarly, if all individuals in a population have good learning ability, then this population will quickly evolve to a certain heights. Inspired by this, this paper will be biological learning ability into the moth flame algorithm.

In the introduction of the algorithm 2.2, carefully observe the Eq. (7) found that each moth to update its position allows the use of Eq. (7) in the flame, each iteration and update the flame list, sort the fitness value of the flame, And then the moths are updated with respect to the flame. The first moth always updates the position with respect to the best flame, and the last moth updates the position with respect to the worst flame of the list. In this paper, introduced the concept of learning, that is, all the moths update the location of reference to the best flame, this will greatly improve the search efficiency of moths.

Through the above statements, the update mechanism of moth’s position made the following improvement:

$$S(M_i, F_i) = D_i \cdot e^{bt} \cdot \cos(2\pi) + F_1$$

where $F_1$ indicates the best individual in each generation.

In order to verify whether the above improvements are effective, 8 classical test functions with different characteristics are selected. The mathematical expressions of the 8 test functions, the dimension, the convergence region and the global optimal solution are listed in table 1. The uni-modal functions\cite{7} (F1-F4) are suitable for bench-marking the exploitation of algorithms since they have one global optimum and no local optima. In contrary, multi-modal functions\cite{7} (F5-F8) have a massive number of local optima and are helpful to examine exploration and local optima avoidance of algorithms.

The eight test functions shown in Table 1 were subjected to 30 independent optimization calculations. The comprehensive evaluation can be conducted from the solution of the properties and time, and many other aspects of the performance of the algorithm. Set the maximum number of iterations to $10^3$ and the initial population size of the algorithm to 50. Table 2 lists the improvement before and after the eight test function of the optimal statistical results, Figure 5,6 shows eight test function improvement before and after the convergence curve.
### Table 1. Test function.

<table>
<thead>
<tr>
<th>Number</th>
<th>Function</th>
<th>Dim</th>
<th>Range</th>
<th>$f_{\text{min}}$</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$f_1(x) = \sum_{i=1}^{n} x_i^2$</td>
<td>30</td>
<td>(-100,100)</td>
<td>0</td>
<td>Uni-modal</td>
</tr>
<tr>
<td>2</td>
<td>$f_2(x) = \sum_{i=1}^{n}</td>
<td>x_i</td>
<td>+ \prod_{i=1}^{n}</td>
<td>x_i</td>
<td>$</td>
</tr>
<tr>
<td>3</td>
<td>$f_3(x) = \max {</td>
<td>x_i</td>
<td>, 1 \leq i \leq n }$</td>
<td>10</td>
<td>(-100,100)</td>
</tr>
<tr>
<td>4</td>
<td>$f_4(x) = \sum_{i=1}^{n} i x_i^4 + \text{random}(0,1)$</td>
<td>10</td>
<td>(-1.28,1.28)</td>
<td>0</td>
<td>Uni-modal</td>
</tr>
<tr>
<td>5</td>
<td>$f_5(x) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$</td>
<td>10</td>
<td>(-5.12,5.12)</td>
<td>0</td>
<td>Comp site</td>
</tr>
<tr>
<td>6</td>
<td>$f_6(x) = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i)) + 20$</td>
<td>10</td>
<td>(-32,32)</td>
<td>0</td>
<td>Comp site</td>
</tr>
<tr>
<td>7</td>
<td>$f_7(x) = \frac{1}{400} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$</td>
<td>10</td>
<td>(-600,600)</td>
<td>0</td>
<td>Comp site</td>
</tr>
<tr>
<td>8</td>
<td>$f_8(x) = \frac{\pi}{n} (10\sin(x_{y_1}) + \sum_{i=1}^{n-1} (y_{i+1} - 1)^2 [1 + 10\sin^2(x_{y_{i+1}})] + (y_n - 1)^2] + \sum_{i=1}^{n} u(x_i,10,100,4)$</td>
<td>10</td>
<td>(-50,50)</td>
<td>0</td>
<td>Comp site</td>
</tr>
</tbody>
</table>

### Table 2. Optimized results.

<table>
<thead>
<tr>
<th>Function</th>
<th>Best value</th>
<th>Worst value</th>
<th>Optimization average value</th>
<th>Iteration time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MFO</td>
<td>GMFO</td>
<td>MFO</td>
<td>GMFO</td>
</tr>
<tr>
<td>F1</td>
<td>3.47E-06</td>
<td>9.04E-17</td>
<td>2.95279E-05</td>
<td>1.116901</td>
</tr>
<tr>
<td>F2</td>
<td>4.38E-21</td>
<td>2.25E-42</td>
<td>3.41143E-19</td>
<td>0.693869</td>
</tr>
<tr>
<td>F3</td>
<td>4.26E-10</td>
<td>1.28E-23</td>
<td>3.60487E-06</td>
<td>1.757321</td>
</tr>
<tr>
<td>F4</td>
<td>0.000975</td>
<td>0.000355</td>
<td>0.003005501</td>
<td>0.996717</td>
</tr>
<tr>
<td>F5</td>
<td>3.979836</td>
<td>3.979836</td>
<td>19.20739933</td>
<td>0.900864</td>
</tr>
<tr>
<td>F6</td>
<td>4.44E-15</td>
<td>4.44E-15</td>
<td>4.44E-15</td>
<td>1.879977</td>
</tr>
<tr>
<td>F7</td>
<td>0.034458</td>
<td>0.022141</td>
<td>0.157710647</td>
<td>1.142977</td>
</tr>
<tr>
<td>F8</td>
<td>4.71E-32</td>
<td>4.71E-32</td>
<td>5.39915E-32</td>
<td>1.938054</td>
</tr>
</tbody>
</table>

![Figure 5. Convergence curve of test function 1-4.](image1)

![Figure 6. Convergence curve of test function 5-8.](image2)
First, the "best value" of the first column in Table 2 embodies the search precision of the algorithm, and the "worst value" and "average value" of the second and third columns. It is the worst value and arithmetic mean of 30 independent optimization, which can reflect the stability of the algorithm, while the convergence time of the fourth column shows the convergence speed of the algorithm. Comparing the results in Table 2, it can be seen that the improved algorithm has better searching ability due to the change of the adaptive flame quantity updating mechanism and the ability to learn to the current best individual. The better results can be obtained while take less time. Analysis of the convergence curve of Figure 5, 6 can be seen, MFO algorithm can be smooth convergence to the global optimum, there is no premature phenomenon, proved MFO algorithm can effectively avoid premature maturation, improved MFO algorithm inherited the advantages of MFO. The improved MFO algorithm improves the convergence speed and convergence precision by changing the adaptive flame updating mechanism and having the ability to learn to the current individual.

It can be seen that the improved MFO algorithm has better searching effect for most of the functions, especially for complex functions with high dimensions and multi-peaks, and has better robustness and global search ability.

Examples of Optimization Design Based on Moth-Flame Algorithm[9]

With the rapid development of the domestic economy, scientific and technological innovation has become a key factor in promoting economic development. Lifting equipment, large-scale, high-speed, specialization, intelligent, lightweight and other trends have been recognized by the industry; lightweight lifting equipment can not only reduce the production cost of lifting equipment products, reduce the lifting equipment work consumption, and can reduce the investment[10]. In order to ensure the safe and stable operation of the crane, the main beam is usually designed to box-type structure, so that it can meet the crane in the complex conditions of bending, twisting use. Because of the quality of the crane's metal structure accounted for about 60% of the total weight, taking into account the crane's manufacturing costs, saving steel and caused by the weight of the plant construction cost influence, the lightweight design of metal structure has become an important issue for the design of the crane.

The dead weight of the main girder of the bridge crane is a uniformly distributed load $q$. The lifting force acting on the middle point of the beam is $p$. Figure 7 shows the mechanical model of bridge crane under uniform and concentrated load. The section of the box girder of the bridge crane is shown in Figure 8. As shown in Figure 8 of the main girder box section, $a$ is thickness of the web; $t$ is thickness of the upper and lower cover plate; $b$ is beam width; $h$ is beam height; $d$ is distance between the web.

Taking into account the actual manufacturing conditions and the geometric relationship between the parameters of the box girder section, the geometric constraint condition of the box girder section is defined as the form shown in Eq. 11:
Strength constraint condition:

\[
\frac{M_y}{I_x} \leq [\sigma]
\]  

(13)

where \( M \) is the bending moment of the most dangerous section; \( y \) is the vertical distance between the centre of the section and the dangerous point; \( I_x \) is the section moment of inertia of box beam; \([\sigma]\) is the allowable stress, \([\sigma]=\frac{235}{1.48}=158.78\text{MPa}\).

Stability Constraints:

\[
\frac{h-t}{a+d} \leq 3
\]  

(14)

Stiffness constraint conditions:

\[
Y_e \leq [Y]
\]  

(15)

where \( Y_e \) is the maximal deformation of the main girder under the concentrated load; \([Y]\) is the structure allows static displacement, \([Y]=l/1000=16500/1000=16.5\text{mm}\).

The objective of the lightweight design is to obtain the minimum structural quality, while satisfying the design requirements. Since the span is known and has a certain value, the objective function is simplified to the area \( A \) of the box beam section.

\[
\min \quad A = 2(ah + tb - 2at)
\]  

(16)

Based on PSO, FOA, MFO and improved MFO three intelligent algorithms, the crane optimized analysis. The performance of the improved MFO algorithm is evaluated comprehensively from two aspects: optimal value and average time-consuming. The initial population size of the algorithm is 20, and the number of iterations is 100 times. The PSO algorithm is implemented by literature [11]. The FOA algorithm is implemented by literature [12]. Table 3 lists the optimization results of the three algorithms.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Parameters (mm)</th>
<th>Iteration time (s)</th>
<th>min (mm²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSO</td>
<td>8.212</td>
<td>12.107</td>
<td>416.73</td>
</tr>
<tr>
<td>FOA</td>
<td>5.042</td>
<td>10.139</td>
<td>456.01</td>
</tr>
<tr>
<td>MFO</td>
<td>5.013</td>
<td>10.012</td>
<td>400.56</td>
</tr>
<tr>
<td>GMFO</td>
<td>5.064</td>
<td>10.016</td>
<td>402.49</td>
</tr>
</tbody>
</table>

From the data in Table 3, it is found that the optimization result of PSO is the worst from the smallest optimization value, followed by the fruit fly algorithm (FOA), the optimization result of Moth Flame Algorithm (MFO) is 13139.24, the relative PSO decreased 30.5%, the relative FOA decreased 8.04%, and the improved MFO algorithm to improve less 0.622%, from the time point of
view, FOA optimization is the longest, followed by MFO, PSO by optimization of the shortest time. From the convergence curve in Fig. 9, it can be seen that the PSO algorithm has the ability to converge quickly in the search space. However, for a problem with multiple local optima solutions, the PSO algorithm tends to fall into the local optimum and cause the optimization of the algorithm. FOA has the ability of fast convergence, but its convergence precision and easy to fall into local optimum; improved MFO algorithm can converge to the global best smooth, no premature phenomenon, it is proved that the MFO algorithm can effectively avoid premature.

![Figure 9. Convergence curve of three optimization algorithms.](image)

It is proved that the MFO algorithm has strong ability to deal with complex optimization problems, and has good robustness and global search ability. Improved MFO algorithm inherited the advantages of MFO, The improved MFO algorithm improves the convergence speed and convergence precision by changing the adaptive flame updating mechanism and having the ability to learn to the current individual.

**Conclusions**

This paper makes a theoretical analysis of a novel bionic swarm intelligence algorithm, the Moth-Flame Optimization (MFO), the algorithm is the imitation of nature moth lateral positioning navigation mechanism, which is based on stochastic optimization algorithm based on population search. However, due to the existence of a large number of moths in a random state, need to repeatedly test, which takes a long time and other defects, this paper proposes the following improvements:

1) Change the flame adaptive mechanism, which improve the convergence speed of the adaptive flame number;

2) Given the ability of "study" to the moths when moths update position, that all moths update the position with reference to the best flames, which improve the search accuracy.

As the improved MFO algorithm changes the adaptive flame updating mechanism and has the ability to learn to the current best individual, compared with the traditional moth flame optimization algorithm, improve the traditional moths flame algorithm search efficiency. The improved MFO algorithm not only inherits the advantages of MFO algorithm, such as few parameters, high convergence precision and good robust performance, but also improves the search speed and convergence precision. Through the 8 classical test functions and 1 engineering examples show that the improved algorithm of flame moth can significantly improve the efficiency optimization of flame. It has quick search speed, high convergence precision and avoids getting into local optimum, the moth flame improved algorithm has the ability to solve practical problems, it has wide application prospect.

**References**


