ISSN: 2664-9640

The neural network-based Ant Lion Optimization algorithm and its applications

Chen Liang[1,a], Lipu Zhang [1,2,b,] *

[1] College of Media Engineering, Communication University of Zhejiang, Hangzhou, 310018, China

[2] Key Lab of Film and TV Media Technology of Zhejiang Province, Hangzhou, 310018, China

[a]215702116@stu.cuz.edu.cn, [b]zhanglipu@cuz.edu.cn

Abstract

This article uses neural networks to optimize the parameter optimization problem in the ant lion algorithm, designs a new algorithm, and compares the newly designed algorithm with three existing algorithms. Numerical examples show that the newly designed algorithm has considerable advantages. Furthermore, we use the new algorithm to solve the Fresnel refractive index problem, and examples show that the newly designed algorithm can better solve complex engineering problems.

Keywords

Optimization algorithm, applications.

1. Introduction

Group intelligent optimization algorithm[1] is a biomimetic optimization algorithm that mimics the behavioral rules and evolution mechanisms of natural group collaboration to accomplish the given task through the interaction of a group of simple individuals. Based on the principles of collaboration and information sharing, this type of algorithm explores local optimal solutions in the optimization space from multiple perspectives to achieve the efficiency of global optimal solutions. Therefore, when dealing with complex high-dimensional optimization problems, group intelligent optimization algorithms are more robust and efficient than traditional optimization algorithms.

The Ant Lion Optimizer (ALO) was first proposed by Seyedali Mirjalili[2]. This algorithm mimics the hunting method of ant lions and has the advantages of few parameters and fast convergence speed, making it widely applied. However, the algorithm still has some drawbacks such as tendency to get stuck in local optima and failure to achieve global optimization. In large-scale or high-dimensional optimization problems, the algorithm may take a long time to find satisfactory solutions. To address the issues of ant lion algorithm, some improved schemes have been proposed. For example, Hu Hongping et al. [3] were inspired by particle swarm optimization algorithm to improve the way of updating elite ant lion positions, thereby enhancing the performance of the algorithm. In addition, Li Zirui and Ouyang Jiali [4]introduced collision-free factors and elimination mechanism in ant lion algorithm, which has been well applied in engineering.

The earliest proposal of Artificial Neural Network (ANN) can be traced back to 1943, when Warren McCulloch and Walter Pitts first introduced it in their paper "A Logical Calculus of the Ideas Immanent in Nervous Activity" [5]. It has strong robustness and fault tolerance, and its self-learning, self-organizing, and self-adaptive characteristics enable the network to handle unknown problems.

ISSN: 2664-9640

In reality, the Ant Lion obtains past hunting experiences and selects the most easily accessible locations to lay its traps based on actual situations, making it easier to catch ants. To simulate this phenomenon, an artificial neural network is introduced into the traditional Ant Lion optimization algorithm. The neural network model provides an approximate estimation of the objective function value, and the ALO algorithm uses these estimated values to determine how to update the Ant Lion's position in order to find better solutions. In this way, the neural network and ALO algorithm collaborate to reduce the risk of the original optimization algorithm falling into local optima and make the optimization process more accurate and efficient.

To demonstrate the effectiveness of the algorithm designed above, numerical examples were tested, including several commonly used CEC test functions. Furthermore, to test the application of the algorithm, we used the improved Ant Lion optimization algorithm to estimate the directivity of the medium.

The structure of this article is as follows: Chapter 1 briefly introduces the problem; Chapter 2 provides a detailed explanation of the key steps of the classic Ant Lion algorithm; Chapter 3 combines artificial neural networks with the classic Ant Lion algorithm and uses CEC2005 test functions to compare the newly designed algorithm with three existing Ant Lion algorithms; Chapter 4 provides a specific application example, where we use the improved algorithm to estimate the refractive index of a medium based on Fresnel equations. Finally, we provide some conclusions and future research directions.

2. The Ant Lion Optimizer

Ant lions are generally found in arid, sandy areas. They dig cone-shaped sand pits with steep edges in suitable locations to serve as their nests and hunting traps. These pits are easy for prey (primarily ants) to fall into, so ant lions spend most of their time waiting for prey at the bottom of the trap. When hunting, if an ant lion senses that prey has fallen into the trap, it will attempt to grab it, while the prey will also try to escape. In this case, the ant lion throws sand to the edge of the trap, making the slope steeper and harder for the prey to escape. This allows the ant lion to push the prey towards the bottom of the pit and consume it. After each successful hunt, the ant lion repairs the trap and waits for the next prey. The Ant Lion Optimization Algorithm simulates the process of ants wandering and ant lions hunting. The algorithm sets up two groups of clusters, ants and ant lions. In each iteration, an ant lion will be selected through the roulette wheel selection strategy and used as the temporary optimal solution. Ants will randomly wander around the selected ant lion, searching for better solutions. When an ant finds a better solution than the selected ant lion, the ant lion will eat the ant and replace its position, updating the optimal solution. As the number of iterations increases, the range of random wandering of ants gradually decreases, and they gradually approach the global optimal solution. The accuracy of the algorithm gradually improves.

2.1. Main steps of ALO algorithm

The specific steps of ALO are as follows:

Initialize the data. Set the parameters such as the number and dimension of the two populations, the iteration times, etc., initialize the positions of the ant lions and ants within the specified range, and calculate their respective fitness.

Determine the elite ant lions. Select the ant lion with the best fitness from the initialized ant lion population as the elite ant lion.

Use the roulette wheel selection strategy to select an ant lion for each ant, update the parameters based on the position of the selected ant lion, and allow the ant to randomly wander around the selected ant lion to update its position.

ISSN: 2664-9640

Calculate the fitness values of both ants and ant lions. Update the position of the ant lions based on the positions and fitness of the ants, and the position with the best fitness becomes the new elite ant lion.

Determine whether the algorithm termination condition is met. If it is, output the result, return the elite ant lion and its fitness, and end the iteration; otherwise, go back to step (3).

3. Ant Lion Optimization with Artificial Neural Networks

In reality, the antlion relies on its experience and perception of the environment to determine the direction and length of its movement, and constructs and repairs traps and nests at appropriate locations. Neural networks can calculate and process input data to produce predicted results. By combining neural networks with antlion optimization algorithms, the improved algorithm can predict the position of the optimal solution based on the input environmental information and the objective function value, which improves the simulation of the above phenomenon. This simulation can improve the search ability and efficiency of the antlion optimization algorithm, so as to better find the optimal solution to the problem.

3.1. Main Steps of ALO Algorithm Based on Neural Network

The improved algorithm steps are as follows:

Initialize data and neural network parameters. Set the population size (number of antlions and ants), problem dimension, maximum iteration times, and other parameters. Initialize the neural network architecture (number of layers, number of neurons, etc.) and initial weights.

Train the neural network. Generate a training dataset and use it to train the neural network.

Initialize the antlions' and ants' positions. In the search space of the problem, randomly initialize the positions of each antlion and ant, and calculate their fitness according to the objective function of the problem.

Global search. Use the neural network to calculate the predicted value of each antlion's position, and calculate its fitness based on the difference between the predicted value and the target value. Then sort the antlions by fitness and update the elite antlions (current optimal solutions).

Neural network parameter update. Take the selected elite antlions' positions as inputs, calculate their corresponding predicted values through the neural network, calculate the loss (difference between the predicted value and the target value), perform gradient backpropagation based on the loss, and update the neural network's weight parameters.

Local search. Generate new search areas centered around the updated neural network parameters, calculate the fitness of each ant, select an antlion for each ant based on the roulette wheel selection strategy, and randomly wander around the selected antlion to update the ant's position.

Update antlion positions and fitness. Calculate new fitness values based on the new positions of ants and antlions. If an ant's fitness exceeds an antlion's fitness, update the antlion's position to that of the ant. Re-evaluate the fitness of all antlions and update the elite antlions.

Check if the algorithm stop condition is met. If it is met, output the results, return the elite antlions and their fitness, and end the iteration; otherwise, return to step (d) for continued iteration.

3.2. Experiment and analysis

In this section, we compare the performance of four different ant lion algorithms on some test functions of CEC2005. These algorithms are: neural network-based ant lion optimization algorithm (NNALO), cossine-distribution-based ant lion optimization algorithm (CALO) [6], self-adaptive ant lion optimization algorithm (PSALO) [7], and original ant lion optimization algorithm (ALO). We analyze the effect of algorithm improvement by comparing the solution

ISSN: 2664-9640

results of these algorithms on the test functions. The experimental running platform is PyCharm Community Edition 2023.1.2, the operation system is Windows 11, the memory size is 16GB, and the CPU is i9-13900HX.

The test functions used are shown in the table below:

Table 1: Some test functions from CEC2005

NO.	Functions	Search space	f_{min}
F1	$F_1(x) = \sum_{i=1}^{n} x_i^2$	$[-100,10($	0
F2	$F_2(x) = \sum_{i}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	$[-10,10]^D$	0
F3	$F_3(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$	$[-100,10($	0
F4	$F_4(x) = max_i\{\lvert x_i \rvert, i \le i \le n\}$	$[-100,10($	0
F5	$F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30,30]^D$	0
F6	$F_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	$[-100,10($	0
F7	$F_7(x) = \sum_{i=1}^{n} i x_i^4 + random(0,1)$	$[-1.28,1.2$	0
F8	$F_8(x) = \sum_{i=1}^{n} [x_i^2 - 10cos(2\pi x_i) + 10]$	$[-5.12,5.1$	0
F9	$F_9(x) = -20exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} {x_i}^2}) - exp(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i)) + 20 + e$	$[-32,32]^D$	0
F10	$F_{10}(x) = \frac{1}{4000}\sum_{i=1}^{n} {x_i}^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600,60($	0

ISSN: 2664-9640

F11	$F_{11}(x) = \frac{\pi}{n}\{10sin^2(\pi y_1) + \sum_{i=1}^{n-1}(x_i-1)^2[1 + 10sin^2(\pi y_{i+1})] + (y_n-1)^2\} + \sum_{i=1}^{n} u(x_i,10,100,4)$ $y_i = 1+\frac{x_i+1}{4}$, $u(x_i,a,k,m) = \begin{cases} k(x_i-a)^m & x_i > a \\ 0 & -a \le x_i \le a \\ k(-x_i-a)^m & x_i < -a \end{cases}$	0	$[-50,50]^D$	0
F12	$F_{12}(x) = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i-1)^2[1+\sin^2(3\pi x_{i+1})] + (x_n-1)^2[1+\sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i,5,100,4)$	0	$[-50,50]^D$	0
F13	$F_{13}(x)=\left(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})}\right)^{-1}$		$[-65.536,$	0.998003837 79445
F14	$F_{14}(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1-\frac{1}{8\pi}\right)\cos x_1 + 10$		$[-5,10]$ $\times [0,15]$	0.397887357 729738
F15	$F_{15}(x) = [1+(x_1+x_2+1)^2(19-14x_1+3x_1^2 - 14x_2+6x_1x_2+3x_2^2)] \times [30 + (2x_1-3x_2)^2(18-32x_1+12x_1^2 + 48x_2-36x_1x_2+27x_2^2)]$		$[-2,2]^D$	2.999999999992

The parameter settings for this section of the experiment are as follows: both population sizes are set to 20, and the maximum iteration count is set to 10,000. The neural network's hyperparameters include a learning rate alpha of 0.5, a regularization parameter beta of 0.5, a momentum term gamma of 0.5, and a decay factor epsilon of 1e-06. To ensure the accuracy of the results, we independently ran each algorithm 20 times and calculated the average and

ISSN: 2664-9640

standard deviation of the solution results for each test function. The results are shown in Table 2.

Table 2: Comparison of Results for Solving Test Functions

F		ALO	NNALO	CALO	PSALO
$F_1(x)$	mean	5.9142e-09	1.3241e-11	6.4911e-09	5.0791e-09
	std	7.0441e-18	6.7361e-12	1.4942e-17	2.0195e-17
$F_2(x)$	mean	0.3562	0.1899	0.4009	0.3587
	std	0.0564	0.1453	0.0542	0.0314
$F_3(x)$	mean	0.2152	0.0376	0.2334	0.2490
	std	0.0131	0.0217	0.0129	0.0170
$F_4(x)$	mean	0.1612	0.0823	0.1464	0.1511
	std	3.1954e-03	0.0374	2.5162e-03	2.3087e-03
$F_5(x)$	mean	1.4540	0	1.3863	0
	std	40.1694	0	36.5178	0
$F_6(x)$	mean	5.2497e-09	1.2553e-11	5.2017e-09	5.4318e-09
	std	1.1620e-17	9.5134e-12	1.2743e-17	1.9766e-17
$F_7(x)$	mean	0.0918	0.0447	0.1014	0.0890
	std	1.6104e-03	0.0188	1.3771e-03	1.8421e-03
$F_8(x)$	mean	93.7746	91.5857	91.6354	75.4177
	std	521.1842	17.9573	1.052e+03	618.6692
$F_9(x)$	mean	3.5153e-04	9.5195e-05	3.7426e-04	3.4196e-04
	std	9.0499e-09	3.6086e-05	1.0448e-08	1.7344e-08
$F_{10}(x)$	mean	1.5635e-02	3.1864e-03	1.8968e-02	1.4343e-03
	std	4.7831e-04	2.6096e-03	1.4096e-03	1.7817e-06
$F_{11}(x)$	mean	4.5370	1.5705e-32	4.4605	2.9738
	std	6.5412	2.7369e-48	3.6082	2.0936
$F_{12}(x)$	mean	0.0633	0.0489	0.0487	0.0195
	std	2.1570e-03	2.4654e-17	2.3703e-03	1.5170e-03
$F_{13}(x)$	mean	0.2720	0.0359	0.4203	0.2260
	std	0.1160	0.1051	0.3035	0.0961
$F_{14}(x)$	mean	0.3978	0.3979	0.3979	0.3969
	std	6.5029e-27	4.4747e-14	1.0619e-26	2.2989e-27
$F_{15}(x)$	mean	3.0010	3.0000	3.0001	3.0000
	std	3.1338e-25	1.9915e-13	5.9927e-25	4.0530e-26

ISSN: 2664-9640

The following is a statistical analysis of the number of times each of the four algorithms obtained the optimal mean and standard deviation when solving the 15 functions mentioned above, as shown in Figure 1. Among them, the NNALO algorithm has mostly optimal solution results and strong stability, indicating that improving the ant lion algorithm with neural networks has certain effectiveness.

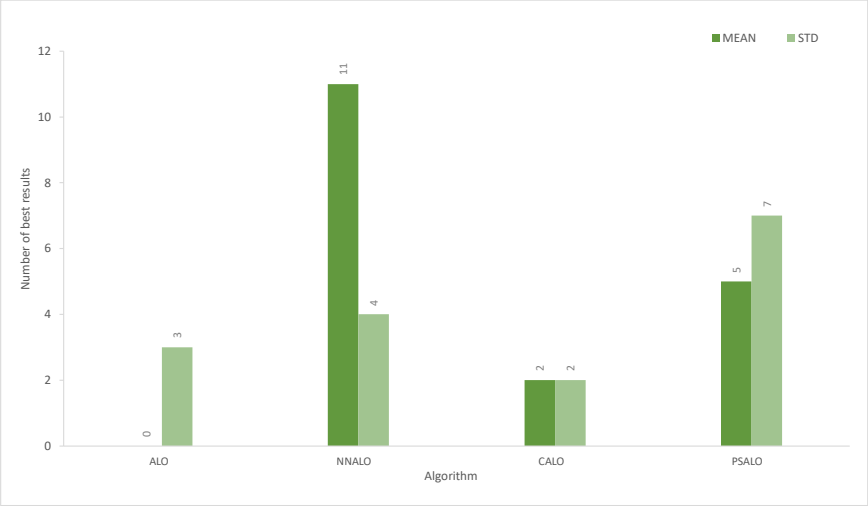

Figure 1: The statistical analysis for the number of times with best results

To better demonstrate the superiority of the NNALO algorithm, we present the convergence plots of the four algorithms (NNALO, CALO, PSALO, and ALO) after 1000 iterations for solving some of the functions mentioned above, as shown in Figure 2.

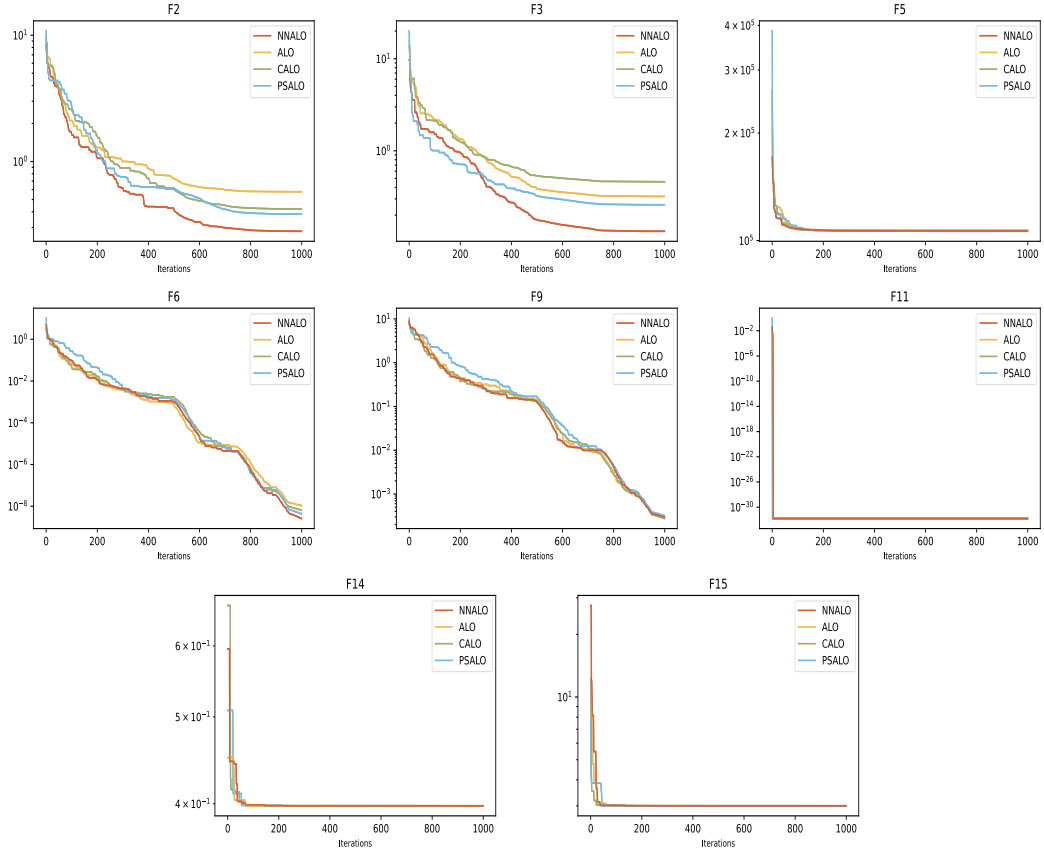

Figure 2: Partial test function convergence result

ISSN: 2664-9640

4. Estimating the refractive index of a medium using the Fresnel equation

4.1. The Fresnel equation

The Fresnel equations are a set of equations that describe the phenomenon of light reflection and refraction at the interface between two media. These equations were proposed by the French physicist Augustin-Jean Fresnel in the early 19th century and have become an important tool in optical research.

The Fresnel equations have the following expression:

$$r_{pi} = \frac{n_2 \cdot \cos(\theta_i) - n_1 \cdot \sqrt{1-(\frac{n_1}{n_2}\sin(\theta_i))^2}}{n_2 \cdot \cos(\theta_i) + n_1 \cdot \sqrt{1-(\frac{n_1}{n_2}\sin(\theta_i))^2}}, \quad (1)$$

$$r_{si} = \frac{n_1 \cdot \cos(\theta_i) - n_2 \cdot \sqrt{1-(\frac{n_1}{n_2}\sin(\theta_i))^2}}{n_1 \cdot \cos(\theta_i) + n_2 \cdot \sqrt{1-(\frac{n_1}{n_2}\sin(\theta_i))^2}}, \quad (2)$$

Where, r_{pi} represents the amplitude reflectivity of the p component at an incident angle of θ_i, r_{si} represents the amplitude reflectivity of the s component at an incident angle of θ_i, and n_1、n_2 represent the refractive index of the medium.

4.2. Application of NNALO to the estimation of refractive index of media

By observing the laws of refraction and reflection in the Fresnel equations, we can transform the estimation of the refractive index of a medium into a parameter identification problem, where the refractive indices n_1 and n_2 of the medium are treated as parameters to be estimated.

The specific steps are as follows:

Organize the data of the reflection rate and corresponding incident angle of a certain medium in reality;

Define the loss function;

Use an optimization algorithm to minimize the loss function and obtain the corresponding refractive indices n_1 and n_2.

We define the loss function as Equation (3) and use the NNALO algorithm as the optimization algorithm for this parameter identification problem.

$$F = \sum_{i=0}^{90}(R_{pi} - r_{pi})^2 + \sum_{i=0}^{90}(R_{si} - r_{si})^2, \quad (3)$$

Where, R_{pi} and R_{pi} are the amplitude reflectivity of the p and s components when the incident angle is θ_i in practice, and r_{pi} and r_{si} are the amplitude reflectivity of the p and s components calculated using Equations and based on the estimated refractive indices n_1 and n_2.

We estimated the refractive index of the medium under three refractive conditions, and the estimation results are shown in the Table 3.

Table 3: The result of NNALO for estimating the refractive index of different media

Medium	Refractive Index				Loss Function
	n1		n2		
	True Value	Estimated Value	True Value	Estimated Value	

ISSN: 2664-9640

Air and Lodine	1.0003	1.0161	3.3400	3.3952	3.8616e-16
Alcohol and Glass	1.3900	1.4191	1.5000	1.5436	1.4437e-11
Crystal and Water	2.0000	2.0219	1.3330	1.3418	1.5279e-11

It can be seen that the result of estimating the refractive index of the medium using the NNALO algorithm is relatively ideal, which proves that the NNALO algorithm has certain value in practical applications.

5. Conclusion

We introduced a neural network to simulate the behavior of trap selection by antlions based on information, in order to improve the antlion optimization algorithm. Through extensive experiments, we found that this improved strategy effectively improves the efficiency of the original algorithm and has a positive effect on achieving global optimization of the algorithm. Subsequently, we applied the improved algorithm to estimate the refractive index of the medium and obtained ideal results.

Our next research goal is to further improve the performance of the algorithm and apply the improved algorithm to solve other problems.

References

[1] Chakraborty, A., Kar, A.K. Swarm Intelligence: A Review of Algorithms. In: Patnaik, S., Yang, XS., Nakamatsu, K. (eds) Nature-Inspired Computing and Optimization. Modeling and Optimization in Science and Technologies, vol 10. Springer, Cham., 2017. https://doi.org/10.1007/978-3-319-50920-4_19C.

[2] Seyedali, M. The Ant Lion Optimizer. Advances in Engineering Software, 83(2015): 80-98, https://doi.org/10.1016/j.advengsoft.2015.01.010.

[3] Hu Hongping, et al."The Improved Antlion Optimizer and Artificial Neural Network for Chinese Influenza Prediction." Complexity 2019.(2019). doi:10.1155/2019/1480392.Seyedali, M. The Ant Lion Optimizer. Advances in Engineering Software, 83(2015): 80-98, https://doi.org /10.1016/j. advengsoft.2015.01.010.

[4] Li Zirui,and Ouyang Jiali."Optimal Design Method for Sub-array Beamforming Based on an Improved Ant Lion Algorithm." Journal of Physics: Conference Series 2437.1(2023). doi:10.1088/1742-6596/2437/1/012106.

[5] McCulloch, W.S., Pitts, W. A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics 5, 115–133 (1943). https://doi.org/10.1007/BF02478259

[6] Jingsen, L., Yu, H., and Yu, L., Preferred Strategy Based Self-adaptive Ant Lion Optimization Algorithm, Pattern recognition and artificial intelligence, 33(2020):121-132. DOI: 10.16451/j. cnki. issn1003-6059.202002004.

[7] Kamel, N., Ouchen, I., Baali, K. A Sampling-PSO-K-Means Algorithm for Document Clustering. In: Pan, JS., Krömer, P., Snášel, V. (eds) Genetic and Evolutionary Computing. Advances in Intelligent Systems and Computing, vol 238. Springer, Cham, 2014. https://doi.org/10.1007/978-3-319-01796-9_5.