Antlion Algorithm Based on Immune Cloning and Its Truss Optimization

Liyancang Li
Hebei University of Engineering
Wuyue Yue Wu (1065073479@qq.com)
Hebei University of Engineering https://orcid.org/0000-0003-0184-8516

Research

Keywords: * computer application technology, Antlion optimization algorithm, immune clonal algorithm, combinatorial optimization

Posted Date: December 6th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-1098073/v1

Antlion algorithm based on immune cloning and its truss optimization

LI Yancang[1](ID) WU Yue[2]

(College of Civil Engineering, Hebei University of Engineering
Handan 056038,Hebei,P.R.China)

Abstract Antlion optimization algorithm has good search and development capabilities, but the influence weight of elite ant lions is reduced in the later stage of optimization, which leads to slower algorithm convergence and easy to fall into local optimization. For this purpose, an antlion optimization algorithm based on immune cloning was proposed. In the early stage, the reverse learning strategy was used to initialize the ant population. The Cauchy mutation operator was added to the elite antlion update to improve the later development ability of the algorithm; finally, the antlion was cloned and mutated with the immune clone selection algorithm to change the position and fitness value of the antlion, and further improve the algorithm's global optimization ability and convergence accuracy. 10 test functions and a 0~1 backpack were used to evaluate the optimization ability of the algorithm and applied to the size and layout optimization problems of the truss structure. The optimization effect was found to be good through the force effect diagram. It is verified that ICALO is applied to combinatorial optimization problems with faster convergence speed and higher accuracy. It provides a new method for structural optimization.
This article is submitted as original content. The authors declare that they have no competing interests.

Keywords * computer application technology; Antlion optimization algorithm; immune clonal algorithm, combinatorial optimization

1. Introduction

The swarm intelligence optimization algorithm is a method for global optimization in the solution space by simulating the behavior of some animals or plants in the natural world. In recent years, scholars have studied a series of new intelligent optimization algorithms, such as Ant Colony Optimization (ACO), Loin Swarm Optimization (LSO), Grey Wolf Optimizer, (GWO), whale optimization algorithm (WOA), etc. These algorithms are robust [1], have a wide range of applications [2] and are simple to operate, and are widely used in various fields [3]. Among them, inspired by the process of ant lion capturing ants, in 2015, the Australian author Seyedali proposed a swarm intelligence algorithm, the ant lion optimizer algorithm (ALO), which aims to solve the function optimization problem [4]. Although the proposed ALO algorithm takes a short time, it has the advantages of global optimization, high convergence accuracy, and better robustness. It is compatible with the new meta-heuristic algorithm Whale Optimization algorithm (WOA) and the sine cosine algorithm, (SCA) and Slap swarm algorithm, (SSA) have a certain degree of comparability. At present, the Antlion optimization algorithm has been applied to improve UAV [5], SVM (Supported Vector Machine) [6], Elman neural network and parallel machine scheduling [7], and other directions, and have achieved good results.

However, in the later stage of the Antlion optimization algorithm, the influence factor of the elite Antlion is small, and it is vulnerable to the influence of the poor Antlion, which causes the algorithm to fall into the local optimum and the convergence speed is slower to solve the problems of low accuracy. The optimization algorithm has been improved. To improve the algorithm’s global search ability and optimization accuracy, Li Zongni et al. [8] changed the ant’s search boundary through the position of the ant lion so that it can better search the effective area with the elite ant lion; Wu Weimin et al. [9] Adjusting the range of antlion shrinkage through adaptive feedback improves the accuracy of the algorithm's convergence; Zhao Xiaoguo et al. [10] improved the antlion optimization

algorithm through ant dynamic search and range shrinking, and improved the convergence speed of the algorithm. To increase the diversity of the population: Wang Yadong et al. [11] introduced the idea of differential evolution into the Antlion optimization algorithm; and improved the update position of the ants through a reverse learning strategy [12]. To solve the premature and local optimal problem of the Antlion optimization algorithm: Haydar Kilic et al. [13] used the tournament selection strategy to change the update formula of the ants and the ant lion in the ALO algorithm while adaptively changing its step length; so that it can better Balance global search ability and local search ability; Alaa Tharwat et al. [14] proposed an ant lion optimization algorithm based on chaotic mapping and penalty function, which reduces the search range of ants and speeds up the optimization of the algorithm.

The above improvements to the Antlion optimization algorithm, although within a certain range, reduce the possibility of the algorithm falling into the local optimum. However, the improved algorithm still has shortcomings such as low convergence accuracy and poor development capability. Further research is still needed. In view of this, considering that Cauchy mutation[15] can disturb the current poor Antlion, prevent the algorithm from falling into the superiority of the situation, it will be reversed. The learning strategy[16] is applied to the ant population to enrich the diversity of the population. Finally, the ALO algorithm with strong exploration ability and the ICS algorithm[17] with good development ability are combined to propose an ant lion optimization algorithm based on immune cloning. Antlion has the ability to explore and develop effective balance algorithms for cloning, mutation, and selection. Through 10 benchmark test functions [18] simulation experiment and combination optimization experiment [19], it is demonstrated that the improved Antlion optimization algorithm has stronger convergence and optimization performance than other algorithms, and the extended algorithm has better performance in actual engineering. Application areas.

2. Basic Antlion optimization algorithm

This article studies a method for ant lions to prey on ants. The core of the ALO algorithm is that the ant lion can represent the optimal solution of the candidate, and the ants walk around the ant lion to search for the best solution in the search space. Whenever the ant searches for a better solution than the ant lion, the ant lion occupies the position of the ant. By comparing the fitness value of the ant lion, the elite ant lion (the optimal solution) is selected to achieve the optimal solution and the local optimal solution. The update process,

The specific calculation steps are as follows [20,21]

1) First, initialize the position parameters of the ants, the custom dimension is *D*, the number of iterations is *T*, and the solution of *N* ant positions is randomly initialized.
2) Select an antlion through the roulette strategy, arrange an ant around the antlion, and record its position at the same time

$$X(t) = [0, cumsum(2p(t_1) - 1, cumsum(2p(t_2) - 1, ..., cumsum(2p(t_{n-1}) - 1] \quad (1)$$

Among them, *cumsum* is the sum of the moving positions of the existing ants, *n* is the maximum number of iterations, and *t* represents the number of random walking steps (in this study, the number of iterations), $p(t)$ is a random function.

3) Normalize the formula

$$X_i^t = (X_i^t - a_i)(d_i^t - c_i^t)) / (b_i - a_i) + c_i^t \quad (2)$$

Among them, X_i^t is the position of the ant in the *t* iteration of the *I* dimension ; a_i, b_i are the upper and lower bounds of the ant's step length, and d_i^t, c_i^t are the upper and lower bounds of the ant's walking. In order to make the iterated ants move within the specified range, *I* is used to restraining them.

$$I = 10^q t / T \quad (3)$$

Among them, I is a ratio, indicating that there is a minimum vector of variables in the t iteration, T refers to the maximum number of iterations, q is related to the number of iterations, and it also defines the maximum vector of variables in the t iteration.

4) The last process is to assume that when the ant has a better fitness than its corresponding ant lion (into the sand), there will be a corresponding ant lion occupying the current ant position so that the ant lion can capture the post-iteration Ants with better adaptability.

5) Since the elite ant lion is the ant lion with the best fitness value, it is assumed that each ant randomly walks around the selected ant lion through the roulette wheel and the elite strategy at the same time, as shown below:

$$Ant_i^t = (R_A^t + R_E^t)/2 \tag{4}$$

Where R_A^t represents the random walk of the ants around the ant lion selected by the roulette in the tth iteration and R_E^t is the random walk around the elite ant lion in the t iteration.

6) Calculate the fitness of the corresponding objective function after updating the position of the ant and compare it with the elite ant lion. Choose a better fitness value as the global optimal solution. If the ant's fitness is greater than the value of the ant lion, update the ant lion's position

7) When the algorithm reaches the maximum number of iterations, the search will be stopped and the optimal solution will be output, otherwise the above process will be looped.

3. Antlion optimization algorithm based on immune cloning

3.1 Reverse learning strategy

The initial population of ALO is randomly generated. Due to the uneven distribution of ants, the population diversity is reduced, which leads to a decrease in the convergence speed of the algorithm. Inspired by the reverse learning strategy proposed by Tizhoosh [22] (2005), the initial population is randomly generated first, and then through the initial. Population generates a reverse population, from which the better population is selected as the next-generation population. The reverse learning strategy can select the population closer to the individual as the initial individual. Since each individual is closer to the optimal solution position, it is helpful to improve the algorithm convergence speed, and because the reverse learning strategy can search for effective areas, it can improve The diversity of the population enhances the global search capability of the algorithm.

First, initialize the definition of the reverse population of the ant population $X_i^* = \left[x_{i1}^*, \ldots, x_{id}^*, \ldots x_{iD}^* \right]$

$$X_i^* = rand \cdot (c_i^t + d_i^t) - X_i \tag{5}$$

$$M=N^2/N+1 \tag{6}$$

Combine the randomly generated ant population and its reverse population into a new population, find the fitness function of the new population, arrange the fitness values in ascending order, and take the first M optimal initial solutions as the new ant initial population. The generated reverse population of ants and the randomly generated population are reorganized into a new population, the fitness value of the new population is recalculated, and the ranking is performed according to the fitness value, and the first M items of the ranking are taken as the initial population of ants.

3.2 Disturbance mechanism of elite ant lions based on Cauchy mutation

Judging the computational efficiency and accuracy of an algorithm mainly weighs its development ability and searchability. According to the ant update formula, it can be known that the probability of being affected by the ant lion and the elite ant lion is the same, so the optimization ability in the later stage is poor and vulnerable

The influence of the poor ant lion has fallen into the superiority prematurely. Therefore, this paper introduces the Cauchy mutation operator in the update of the elite ant lion [23]. Cauchy mutation can search in the area around the elite ant lion. By randomly perturbing around the optimal solution, the ability of the ants to find the local optimization is enhanced. The Cauchy probability density is shown in Equation 2-1

$$g\left(x;x_0,\gamma\right)=\frac{1}{\pi\gamma\left[1+\left(\frac{x-x_0}{\gamma}\right)^2\right]} \tag{7}$$

Among them, x_0 is a positional parameter, γ is a random variable greater than 0, and γ is a real number. In this paper, x_0 =0, γ =1, which is the standard Cauchy distribution. By analyzing its probability density function, it can be known that there is no specific mean and variance, but the mode and median are equal to the position parameter, that is, x_0 its distribution function is shown in equation (8):

$$G(x)=\frac{1}{2}+\frac{1}{\pi}\arctan(x) \tag{8}$$

Compared with the normal distribution and the Gaussian distribution, the Cauchy distribution has a more uniform overall distribution; the extreme value is higher and the probability of the tail is greater, it has a larger dispersion characteristic, and can better search around the extreme point. The perturbation formula for the elite ant lion is as follows:

$$\boldsymbol{X}_{ibset}(t)=\boldsymbol{X}_i(t)+\boldsymbol{X}_i(t)*G(x) \tag{9}$$

$$\boldsymbol{X}_i(t+1)=\begin{cases}\boldsymbol{X}_{ibest}(t), f\left(\boldsymbol{X}_{ibest}(t)\right)>f\left(\boldsymbol{X}_i(t)\right)\\ \boldsymbol{X}_i(t), f\left(\boldsymbol{X}_{ibest}(t)\right)\le f\left(\boldsymbol{X}_i(t)\right)\end{cases} \tag{10}$$

Among them, $f\left(\boldsymbol{X}_i(t)\right)$ represents the fitness value of the elite ant lion in the *t* iteration. Through the local disturbance, the algorithm is guided to jump out of the local optimum.

The individual migration of elite ant lions is conducive to the accelerated convergence of the algorithm, but if the replacement frequency of elite ant lions is high, the optimal ant colony will migrate to another ant colony. In the subsequent iterations, better elites will appear. Antlion captures all the ant populations and makes the algorithm fall into local optimization. Therefore, designing a new strategy to achieve a balance between convergence and diversity is the focus of the next improvement.

3.3 Algorithm fusion strategy

3.3.1 Immune Clone Algorithm, (ICA)

Immune Clone Algorithm(ICA) [24]: First, clone the initial antibody population and select excellent antibodies from it, improve the diversity of excellent antibodies through immune gene manipulation, and use antibody correction operations to change poorer ones. Antibodies, improve the local optimization ability of the algorithm. If the poor ant lion in the middle and late stages of optimization is regarded as an antibody, and the candidate solution of the ant lion is regarded as an antibody, then the immune recognition and immune response process of the biological immune system can be linked with the ant lion optimization algorithm, and the ant lion can be improved through cloning operations. The diversity of the population in the lion optimization algorithm is improved by the use of immune operations to change the poor Antlion to improve the convergence of the algorithm, as shown in Figure 1.

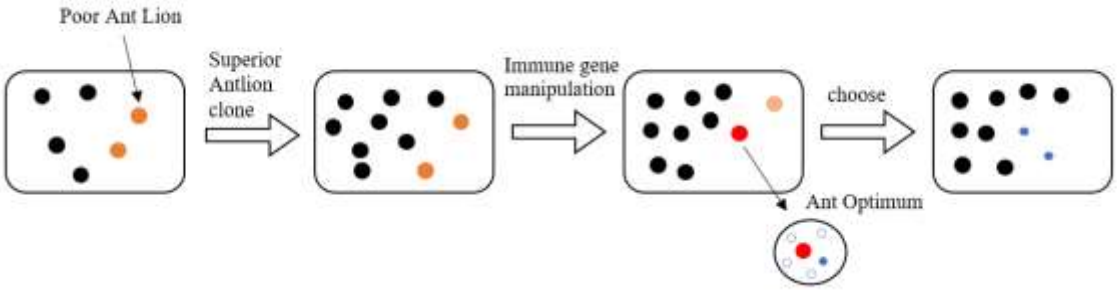


Figure 1 The search process of the Antlion optimization algorithm based on immune cloning

3.3.2 Execution of ICALO

The algorithm uses the basic Antlion optimization algorithm as the framework, and takes the number of stagnant evolutionary steps n>m (the threshold of the number of stagnant evolutionary steps) as the condition for calling the ICA algorithm. When the number of stagnant steps is greater than the threshold of the stagnant evolutionary steps, it will clone and immune Operations such as selection, etc. are introduced into the update of the ant lion. By continuously calling the ICA algorithm, the ant lion mutates, updates the ant's search space, expands the ant's search range, and finally obtains a feasible solution set. The specific operation is as follows:

1) Cloning operation: Through the tournament selection strategy, the ant lions with higher adaptability are selected to ensure that the search space of the ants can surround the better position, and the selected better ant lions are copied according to a certain proportion, and the local area is increased. Diversity of optimization and expansion of ants' optimization space. The implementation process of the tournament selection strategy is to first set the team size to B1, and the selection scale provided by the Antlion group to B2. By randomly selecting B1 Antlions in B2 for comparison, the non-dominant level and the crowded distance are retained. Antlions perform the next round of selection, repeat selection B2 times, and finally selected B2 Antlions are the antibody groups selected by the clone.

The cloned Antlion is as follows

$$\boldsymbol{A}^{(1)}(t)=\left\{\boldsymbol{A}_1^{(1)}(t),\boldsymbol{A}_2^{(1)}(t),\ldots,\boldsymbol{A}_{B_2}{}^{(1)}(t)\right\} \tag{11}$$

The ant-lion antibody group that exists after the cloning operation is shown in formula (12):

$$\begin{aligned}\boldsymbol{A}^{(2)}(t)&=\boldsymbol{A}^{(1)1}(t)+\boldsymbol{A}^{(1)2}(t)+\ldots+\boldsymbol{A}^{(1)m_c}(t)\\&=\left\{\boldsymbol{A}_1^{(1)1}(t),\ldots,\boldsymbol{A}_{B_2}{}^{(1)1}(t)\right\}+\left\{\boldsymbol{A}_1^{(1)2}(t),\ldots,A_{B_2}{}^{(1)2}(t)\right\}+\ldots+\\&\quad\left\{\boldsymbol{A}_1^{(1)m_c}(t),\ldots,\boldsymbol{A}_{B_2}^{(1)m_c}(t)\right\}\\&=\left\{\boldsymbol{A}_1^{(2)}(t),\boldsymbol{A}_2^{(2)}(t),\ldots,\boldsymbol{A}_{n(B_2)}^{(2)}(t)\right\}\end{aligned} \tag{12}$$

Among them, the number of ant lions is $N(a_1)=A_1\times k$, A_1 is the number of ant lions retained during clone selection, and k is the proportion of clones. After a large number of tests, it is proved that k=3 is the best.

2) Adaptive immune gene mutation operation

Immune operation refers to the reorganization and mutation of genes, which can better improve the local fine-tuning ability and search accuracy of the algorithm. Therefore, this article adds an adaptive disturbance immune mutation operator to the Antlion position. The calculation formula is as follows:

$$\begin{cases} \mathrm{AL}_i' = \left(\mathrm{AL}_i - \dfrac{\mathrm{AL}_i - \mathrm{AL}_{\min}}{\mathrm{AL}_{\max} - \mathrm{AL}_{\min}} \cdot r(t) \right) \times (1+\lambda) \text{ , rand} \geq 0.5; \\ \mathrm{AL}_i' = \left(\mathrm{AL}_i + \dfrac{\mathrm{AL}_i - \mathrm{AL}_{\min}}{\mathrm{AL}_{\max} - \mathrm{AL}_{\min}} \cdot r(t) \right) \times (1+N(0,1)) \text{ , rand}<0.5; \end{cases} \tag{13}$$

$$r(t) = 1 - \frac{e^{2(1-t/T)} - e^{-2(1-t/T)}}{e^{2(1-t/T)} + e^{-2(1-t/T)}} rand^{(1-t/t_{\max})^3} \tag{14}$$

Among them: λ is the random variable of [-1,1], N(0,1) is the Cauchy variable that obeys the standard normal distribution, t is the current cycle number, and $\mathrm{AL}_{\max}$, $\mathrm{AL}_{\min}$ is the current optimal and worst antlion positions .

3) Editor's choice

After cloning and mutation, the better ant lion maintains its current position, and the ants corresponding to the worse ant lion continue to search for the best. The compression of the poor ant lion is achieved through local optimization. At the same time, combined with the immune network adjustment mechanism [25], evolution is stagnant. When the number n>3, adjust the poor Antlion, after calculation, choose the following formula:

$$a_c(\mathrm{AL}_i) = \left(\sum_{n}^{j=1} fitness(\mathrm{AL}_j) \right) / fitness(\mathrm{AL}_j) \tag{15}$$

Among them: $fitness(\mathrm{AL}_j)$ is the nectar affinity value in the elite colony, $a_c^{\min}$,, $a_c\mathrm{AL}_i$,, $a_c^{\max}$,and this strategy can make the better ant lion be selected and cloned multiple times in the competition with other ant lions, thereby improving the global convergence speed of the algorithm.

Each time the superior ant lion is cloned and selected, combined with the immune system memory mechanism, the position and fitness of the elite ant lion are shared among the ants and the ant lion colony. When the following formula is satisfied, the position of the elite ant lion is determined to renew.

$$\left| fitness(AL) - fitness(elite\text{-}AL) \right|,, \varepsilon \tag{16}$$

The immune clone algorithm has a faster convergence speed and higher objective function diversity evolutionary algorithm, which can provide more choices for elite ant lions.

3.3.3 Antlion optimization algorithm fused with immune cloning，(ICALO）

For the basic ALO, the reverse learning strategy is used to initialize the population in the early stage. As the basis of the algorithm optimization, the search performance and development performance of the algorithm is improved by adding Cauchy mutation. Finally, the fusion of the clone immune algorithm makes the algorithm avoid falling into the local optimum. The accuracy of the algorithm, the algorithm flow chart is shown in Figure 4:

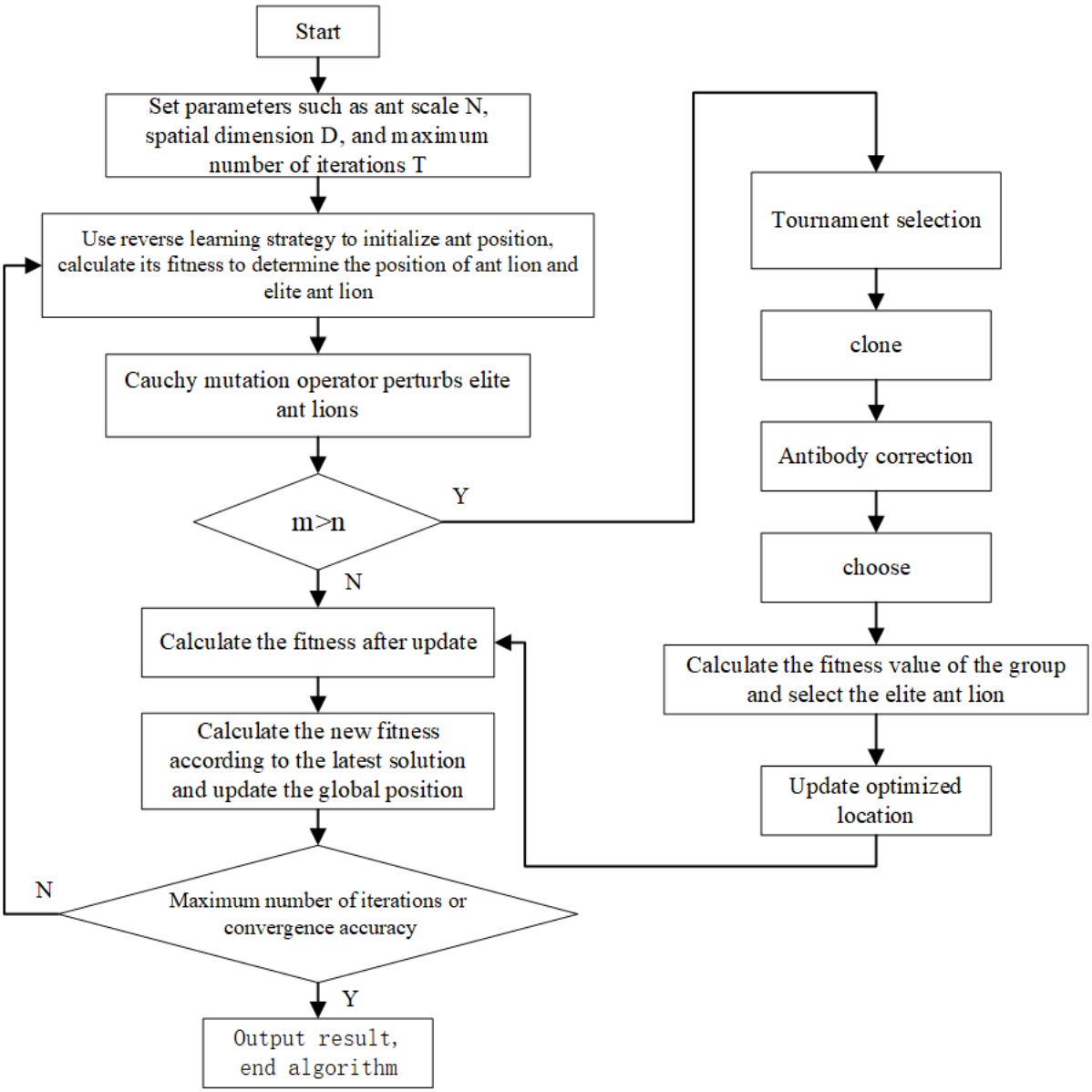


Figure 2 Flow chart of ICALO algorithm

4. Simulation

4.1 Test function

To verify the effectiveness of the improved Antlion optimization algorithm, 10 standard test functions, and classic NP problems are selected to numerically test the performance of the improved algorithm, and the calculated results are compared with the basic ALO; Rajan et al. proposed an Antlion optimization based on a weighted elite strategy Algorithm (Weighted Elitism Based ALO, WALO); Huang Changqing et al. proposed Anti Adjust Chaos Based ALO (HALO); ICA for comparison. The experimental environment is Win10 system, 8GB memory, Matlab2020a software. The parameter settings are: the initial population is 500, and the maximum number of iterations is 500. To avoid contingency, each algorithm is independently run 30 times in Matlab2020a software.

Table 1 shows the basic parameters of each test function. Single-peak [26] and multi-peak [27] are selected to evaluate the convergence speed of the algorithm [28] and the ability to jump out of the local optimum. In the formula, the F1 function and F2 function are single-peak functions, which are mainly used to test the convergence

performance of the improved algorithm in the text; F3~F10 functions are all multi-peak functions, among which F8 is a complex multi-peak function, and the global minimum is located at In a very narrow valley, F9 is moving, rotating, inseparable, and expandable multimodal function. There are countless minimum points in the multimodal function of F10. The function has a strong oscillating shape, and it is difficult to find the global optimal value. The above functions have their characteristics in the optimization feature, which can not only test the ability of the improved algorithm to deal with "premature" problems, but also be suitable for comprehensive testing of the improved algorithm's calculation accuracy, convergence stability, and convergence speed.

Table 1 Test function

Function name	Feature	Dimension	Interval	Optimal
$F_1(x)=\sum_{i=1}^{n}x_i^2$	Unimodal	50/100	[-100，100]	0
$F_2(x)=\sum_{i=1}^{n}[100(x_{(i+1)}-x_i^2)^2+(x_i-1)^2]$	Unimodal	50/100	[-10，10]	0
$F_3(x)=\sum_{i=1}^{n}[x_i^2-10cos(2\pi x_i)+10]$	Multimodal	50/100	[-32，32]	0
$F_4(x)=-20exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2})-exp\left\{\frac{1}{n}\sum_{i=1}^{n}cos(2\pi x_i)\right\}+20+e$	Multimodal	50/100	[6，6]	0
$F_5(x)=\frac{1}{4000}\sum_{i=1}^{n}X_i^2-\prod_{i=1}^{n}cos(x_i/\sqrt{i})+1$	Multimodal	50/100	[-600，600]	0
$F_6(x)=\sum_{i=1}^{n}X_i^2+(\sum_{i=1}^{n}0.5iX_i)^2+(\sum_{i=1}^{n}0.5iX_i)^4$	Multimodal	50/100	[-100，100]	0
$F_7(x)=\sum_{i=1}^{n}(x_{4i-2}^2-10x_{4i-2})^2+5(x_{(4i-1)}-x_4i)^2+(x_{(4i-2)}-2x_{(4i-1)})^4+10(x_{(4i-3)}-2x_4i)^4)$	Multimodal	50/100	[-1000，1000]	0
$F_8(x)=\sum_{i=1}^{D-1}(100(x_i^2-x_{i+1})^2+(x_i-1)^2)$	Multimodal	50/100	[-100，100]	0
$F_9(x)=\sum_{i=1}^{D}(10^6)^{\frac{i-1}{D-1}x_i^2}$	Multimodal	50/100	[-1300，1300]	0
$F_{10}(x)=g(x_1,x_2)+g(x_2,x_3)+...+g(x_{D-1},x_D)+g(x_D,x_1)$	Multimodal	50/100	[-600，600]	0

Table 2 and Table 3 are the experimental results of 10 functions in 50 and 100 dimensions respectively. Fig. 5 is the iterative curve of each algorithm under the 10 functions, which can be specifically analyzed for ICA, ALO,

HALO, WALO and, ICALO respectively:

ALO's unimodal function solution results are worse than other algorithms, but the optimal value obtained in F9 and F10 and the worst value obtained in F7 and F8 are better than the average value of ICA.

The best solution, worst solution, and average value of WALO in unimodal function are better than those of ALO, HALO and ICA. In the multimodal functions F4~F6 and F10, the optimal value and average value are better than other algorithms, but the worst values in F4 and F7 are both worse than HALO. It can be seen that WALO may fall into the local optimization. In the function convergence graph, it can be seen that WALO effectively improves the optimization speed of the algorithm, reducing the number of iterations by half compared to the basic ALO and ICA optimization.

The optimal value, worst value and average value of HALO in F1~F6, F9, F10 are better than ALO and ICA. In the F3, F5, F6~F10 iterative graphs, it can also be seen that the convergence speed is slow.

Compared with ALO, HALO, WALO and ICA, ICALO has better accuracy in the optimal value, worst value and the average value of F1, F3, F5~F10 functions, and it can be seen that it is less affected by dimensionality. It can also show that the solution result is far superior to other algorithms, and it can be seen in the iteration graph that the algorithm converges faster.

From the images and data, we can see that whether it is the algorithm's convergence performance on the function, the operating efficiency, the "local merit" processing problem or the quality of the solution obtained, ICALO is significantly better than other algorithms, and it can be seen that It has better stability when solving high-dimensional functions. This is mainly because the dynamic weight factor is added to the position of the ant lion, which effectively alleviates the problem of reducing the diversity of the roulette strategy, and the faster convergence speed can also be reflected. The introduction of the immune clone has greatly improved the performance of Antlion optimization algorithm out of the game, which fully demonstrates the feasibility and effectiveness of ICALO in solving high-dimensional problems.

Table 2 Comparison of 50-dimensional optimization performance of standard functions

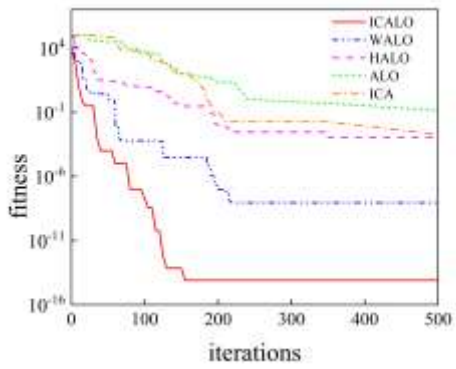
Function	dimension	algorithm	optimal	worst	average	square error
F1	50	ALO	0.13739	3.8784	9.27×10^{-1}	0.4624
		WALO	1.09×10^{-12}	1.31×10^{-9}	2.63×10^{-10}	6.28×10^{-20}
		HALO	3.71×10^{-7}	1.32×10^{-7}	8.59×10^{-7}	4.17×10^{-14}
		ICA	0.00183	1.0422	7.92×10^{-2}	3.68×10^{-2}
		ICALO	4.26×10^{-14}	6.27×10^{-10}	5.61×10^{-13}	6.15×10^{-27}
F2	50	ALO	1.71×10^{-8}	2.56×10^{-6}	7.15×10^{-7}	6.78×10^{-8}
		WALO	9.6951	$6.22\times10^{+7}$	$2.07\times10^{+6}$	1.6251
		HALO	5.11×10^{-7}	9.01×10^{-5}	4.05×10^{-6}	5.11×10^{-15}
		ICA	8.30×10^{-5}	2.00×10^{-4}	1.26×10^{-4}	1.30×10^{-5}
		ICALO	3.64×10^{-8}	4.86×10^{-6}	3.95×10^{-07}	6.15×10^{-13}
F3	50	ALO	77.705	206.58	$1.39\times10^{+2}$	$9.13\times10^{+4}$
		WALO	7.79×10^{-12}	5.59×10^{-10}	1.34×10^{-10}	3.46×10^{-12}
		HALO	4.30×10^{-7}	132.5	50.9	2.05×10^{-9}
		ICA	32.841	125.65	71.5	0.841
		ICALO	6.20×10^{-14}	5.64×10^{-12}	3.62×10^{-12}	6.29×10^{-13}
F4	50	ALO	2.56	14.4	7.81	29.13
		WALO	3.48×10^{-7}	7.47×10^{-6}	2.81×10^{-6}	3.46×10^{-12}

		HALO	8.86×10^{-7}	0.00025	1.92×10^{-04}	2.05×10^{-9}
		ICA	2.4873	13.554	3.84	16.646
		ICALO	3.61×10^{-12}	4.64×10^{-10}	2.61×10^{-10}	2.20×10^{-14}
F5	50	ALO	0.17755	0.90404	0.51009	3.40×10^{-2}
		WALO	1.19×10^{-11}	2.38×10-^{9}	4.45×10^{-10}	2.30×10^{-19}
		HALO	7.39×10^{-7}	0.15853	1.84×10^{-2}	1.87×10^{-13}
		ICA	0.072163	0.32808	1.60×10^{-1}	3.62×10^{-3}
		ICALO	3.94×10^{-9}	9.15×10^{-6}	6.18×10^{-7}	2.31×10^{-14}
F6	50	ALO	48.389	369.76	$2.30\times10^{+2}$	$5.19\times10^{+3}$
		WALO	1.11×10^{-12}	2.24×10^{-10}	4.05×10^{-11}	1.98×10^{-21}
		HALO	5.27×10^{-8}	1.84×10^{-7}	1.01×10^{-7}	7.14×10^{-16}
		ICA	7.7411	78.271	23.7	$3.25\times10^{+2}$
		ICALO	4.41×10^{-15}	9.51×10^{-12}	1.30×10^{-13}	6.18×10^{-19}
F7	50	ALO	1.7838	22.558	11.0	28.5
		WALO	3.41×10^{-13}	4.16×10^{-11}	1.54×10^{-6}	1.66×10^{-22}
		HALO	1.02×10^{-8}	4.27×10^{-8}	2.06×10^{-8}	5.48×10^{-17}
		ICA	1.4741	32.027	12.356	56.195
		ICALO	9.61×10^{-20}	6.44×10^{-13}	6.18×10^{-16}	9.64×10^{-30}
F8	50	ALO	849.45	$1.48\times10^{+5}$	$3.18\times10^{+4}$	$1.99\times10^{+9}$
		WALO	4.30×10^{-10}	2.65×10^{-7}	6.43×10^{-8}	6.00×10^{-15}
		HALO	49.93	55.908	51.62	153.265
		ICA	251.12	$5.26\times10^{+5}$	$3.72\times10^{+4}$	$1.03\times10^{+10}$
		ICALO	6.18×10^{-14}	5.84×10^{-10}	4.65×10^{-13}	8.61×10^{-20}
F9	50	ALO	$1.51\times10^{+6}$	$1.92\times10^{+6}$	$1.73\times10^{+6}$	$3.77\times10^{+13}$
		WALO	2.24×10^{-7}	8.97×10^{-5}	2.28×10^{-5}	5.04×10^{-10}
		HALO	1.58×10^{-2}	5.37×10^{-2}	3.16×10^{-2}	6.67×10^{-5}
		ICA	$4.46\times10^{+6}$	$3.14\times10^{+7}$	$1.52\times10^{+7}$	$3.44\times10^{+7}$
		ICALO	6.18×10^{-13}	8.18×10^{-9}	6.54×10^{-10}	1.98×10^{-17}
F10	50	ALO	11.217	22.529	$2.07\times10^{+1}$	5.229
		WALO	8.27×10^{-11}	2.61×10^{-7}	7.21×10^{-10}	3.15×10^{-19}
		HALO	1.281	18.132	9.81	20.61
		ICA	12.832	19.557	16.665	36.11
		ICALO	1.64×10^{-13}	7.26×10^{-10}	2.19×10^{-11}	2.24×10^{-20}

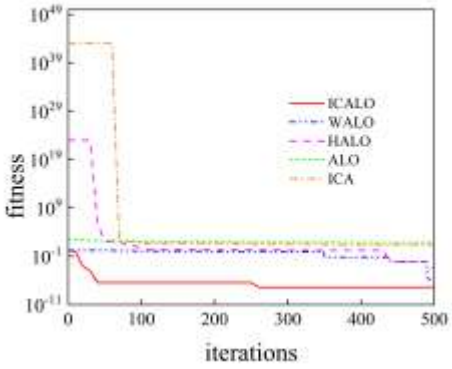
Table 3 100-dimensional optimization performance comparison of standard functions

Function	dimension	algorithm	optimal	worst	average	square error
F1	100	ALO	0.13739	3.8784	0.927	2.4624
		WALO	1.09×10^{-12}	1.31×10^{-9}	2.63×10^{-10}	6.28×10^{-20}
		HALO	3.71×10^{-7}	1.32×10^{-6}	8.59×10^{-7}	4.17×10^{-14}
		ICA	0.00183	1.0422	7.92×10^{-2}	3.68×10^{-2}
		ICALO	6.58×10^{-15}	1.29×10^{-10}	7.26×10^{-11}	2.29×10^{-25}
F2	100	ALO	64.456	$4.54\times10^{+29}$	$1.51\times10^{+28}$	$6.88\times10^{+57}$
		WALO	1.60×10^{-6}	3.03×10^{-5}	9.01×10^{-2}	4.18×10^{-11}
		HALO	0.00019	0.00074	2.90×10^{-4}	1.03×10^{-8}

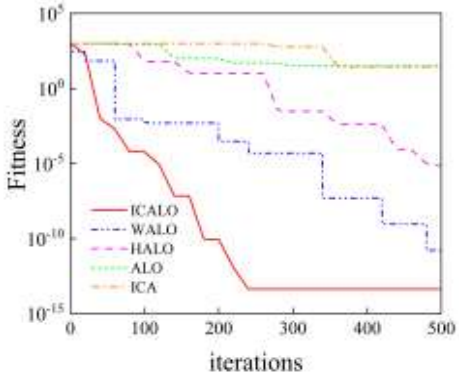
		ICA	24.125	44.733	36.62	63.615
		ICALO	4.09×10^{-8}	6.84×10^{-5}	5.96×10^{-7}	6.84×10^{-15}
F3	100	ALO	252.13	525.33	348.0637	$4.64\times10^{+3}$
		WALO	1.99×10^{-11}	1.57×10^{-9}	3.49×10^{-10}	1.92×10^{-19}
		HALO	1.20×10^{-6}	369.05	$1.55\times10^{+2}$	$2.17\times10^{+4}$
		ICA	131.47	264.94	187.9973	$1.35\times10^{+3}$
		ICALO	$0.00\times10^{+00}$	3.51×10^{-13}	1.89×10^{-15}	4.18×10^{-19}
F4	100	ALO	10.035	16.018	$1.27\times10^{+1}$	23.0
		WALO	8.33×10^{-7}	5.63×10^{-4}	2.79×10^{-5}	1.67×10^{-12}
		HALO	8.88×10^{-6}	0.00031	2.30×10^{-4}	5.01×10^{-9}
		ICA	6.6864	10.179	8.59	23.15
		ICALO	6.94×10^{-9}	6.32×10^{-7}	9.72×10^{-7}	6.58×10^{-16}
F5	100	ALO	5.245 5	31.378	17.733 93	52.62
		WALO	7.82×10^{-12}	1.49×10^{-9}	3.32×10^{-10}	1.54×10^{-19}
		HALO	1.69×10^{-6}	0.275 6	1.79×10^{-2}	4.63×10^{-3}
		ICA	4.314 4	12.694	7.713 64	20.061
		ICALO	9.84×10^{-14}	6.18×10^{-10}	4.94×10^{-13}	9.58×10^{-24}
F6	100	ALO	307.5	1 128.2	701.763	$4.04\times10^{+4}$
		WALO	2.22×10^{-13}	4.03×10^{-10}	1.23×10^{-10}	1.51×10^{-20}
		HALO	4.48×10^{-7}	1.27×10^{-6}	8.25×10^{-7}	4.49×10^{-14}
		ICA	51.612	492.81	158.068 8	$1.85\times10^{+4}$
		ICALO	3.04×10^{-15}	5.18×10^{-11}	9.48×10^{-12}	9.14×10^{-24}
F7	100	ALO	24.856	94.647	47.35	$3.02\times10^{+2}$
		WALO	7.92×10^{-9}	1.24×10^{-6}	3.40×10^{-8}	1.26×10^{-21}
		HALO	5.22×10^{-8}	1.24×10^{-7}	8.33×10^{-8}	3.92×10^{-16}
		ICA	35.394	103.38	64.35	$1.89\times10^{+3}$
		ICALO	6.95×10^{-15}	6.94×10^{-12}	9.59×10^{-13}	5.21×10^{-28}
F8	100	ALO	$3.99\times10^{+6}$	$6.87\times10^{+7}$	$2.25\times10^{+7}$	$2.72\times10^{+14}$
		WALO	5.96×10^{-9}	4.72×10^{-7}	1.18×10^{-7}	1.51×10^{-14}
		HALO	105.95	123.91	$1.13\times10^{+2}$	153.9
		ICA	$1.75\times10^{+6}$	$2.27\times10^{+7}$	$7.36\times10^{+6}$	$2.07\times10^{+13}$
		ICALO	9.58×10^{-13}	5.45×10^{-10}	4.43×10^{-10}	9.84×10^{-21}
F9	100	ALO	$3.35\times10^{+7}$	$1.15\times10^{+8}$	$7.50\times10^{+7}$	$5.17\times10^{+14}$
		WALO	9.59×10^{-7}	9.79×10^{-5}	3.31×10^{-5}	8.78×10^{-10}
		HALO	5.47×10^{-2}	1.54×10^{-1}	9.79×10^{-2}	6.77×10^{-4}
		ICA	4.28×10^{-2}	1.34×10^{-1}	8.95×10^{-2}	4.80×10^{-4}
		ICALO	3.48×10^{-9}	5.78×10^{-8}	1.64×10^{-8}	9.45×10^{-16}
F10	100	ALO	31.65	46.62	41.645	59.462
		WALO	1.65×10^{-11}	4.30×10^{-9}	9.21×10^{-10}	1.12×10^{-18}
		HALO	6.80×10^{-4}	40.651	25.352	130.926
		ICA	27.611	37.62	32.641	47.561
		ICALO	4.29×10^{-14}	9.59×10^{-11}	2.29×10^{-12}	3.23×10^{-24}



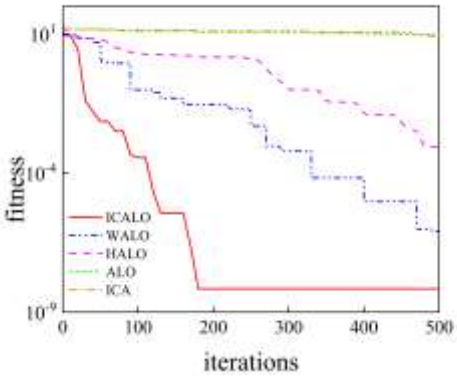
（a）Convergence curve of F1 function



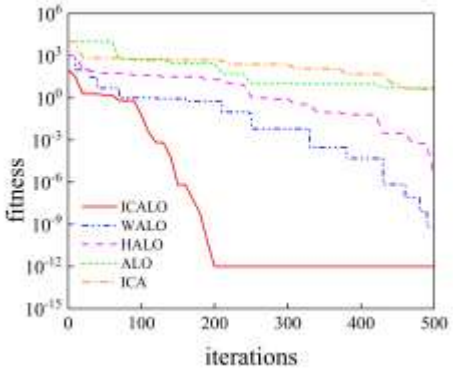
（b） Convergence curve of F2 function



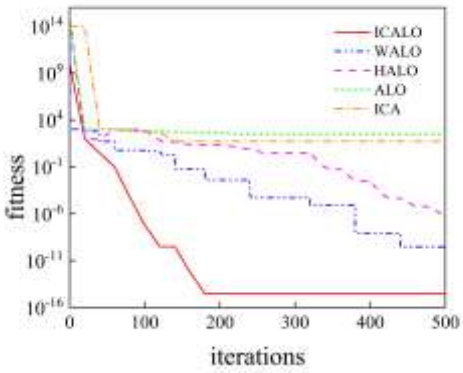
（c）Convergence curve of F3 function



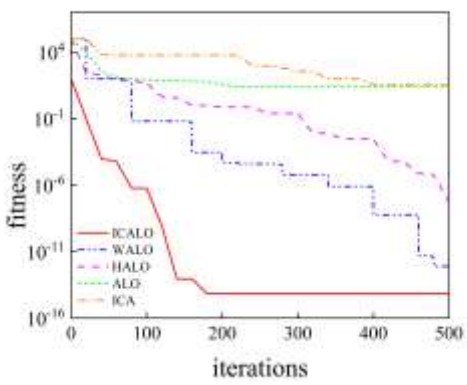
（d） Convergence curve of F4 function



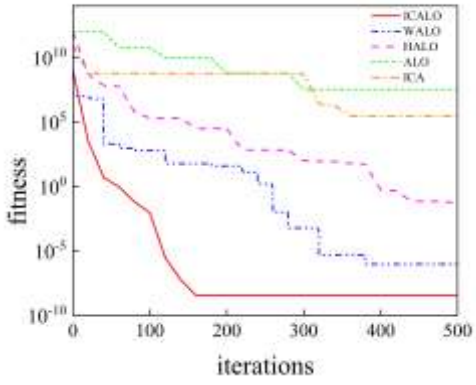
（e）Convergence curve of F5function



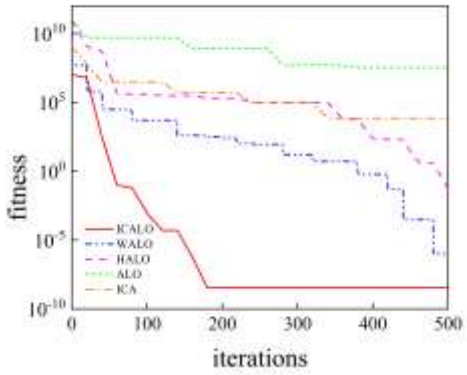
（f）Convergence curve of F6 function



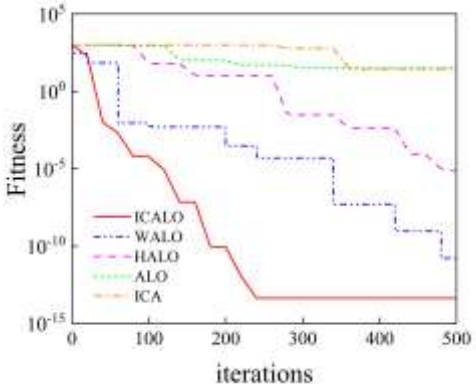
（g）Convergence curve of F7 function



（h）Convergence curve of F8 function



（i）Convergence curve of F9 function



（j）Convergence curve of F10 function

Figure 3 Iterative curve of the algorithm under 10 functions

4.2 0~1 knapsack problem

In the field of combinatorial optimization, the 0~1 knapsack problem is a classic NP problem. Scholars at home and abroad use simulated annealing algorithms [29], particle swarm algorithm [30], genetic algorithm [31] and other algorithms to analyze and study it. This section uses the ICALO algorithm to optimize it and compares it with ALO, WALO and HALO to verify the feasibility of the ICALO algorithm in solving combinatorial problems. The experimental environment is Win10 system, 8GB memory, Matlab2020a software. The initial population is 500, and the maximum number of iterations is 1000. The other parameter settings are the same as in the previous section. Each set of data is run independently 10 times to obtain the optimal solution.

The 0~1 backpack problem refers to selecting suitable items for loading as far as possible under the premise of a certain backpack capacity, so that the value of the items loaded in the backpack is maximized as much as possible. The mathematical model is as follows

Suppose there are D items in total, where the volume of the *i*-th item is corresponding to the price, the backpack capacity is C, xi is used, the variable is set to 0 or 1. If the item can be loaded into the backpack, it is 1, otherwise, it is 0.

$$maxf = RX = \sum_{(i=1)}^{D} R_i x_i \tag{17}$$

$$s.t.WX = \sum_{(i=1)}^{D} w_i x_i \le C \tag{18}$$

Where: $R=(R_1,R_2,...,R_D)$ represents the value vector of the object, $W=(w_1,w_1,...,w_D)$ represents the physical examination vector of the object, $X=(x_1,\mathrm{x}_2,...,\mathrm{x}_D)$ represents the solution vector:

The parameter settings of the algorithm are shown in Table 4

Table 4 Reference table of basic parameters for questions 0~1

parameter	Parameter value
number of the stuffs	*D*=100
Backpack capacity	*C*=1173
Item value	*R* = [199,194,193,191,189,178,174,169,164,164,161,158.157,154,152,152,149,142，131,125,124,124.124,122,119,116,114,113，111,110,109,100, ….,97,94,91,82,82,81,80 ,80,80,79,77,76,74,72,71,70,69,68,65,65,61，56,55,54,53.47,47,46.41,36,34,32,32,30,29,29,26,25,23,22.20,11,10,9,5,4,3,1];
Item weight	*W*=[40,27.5,21,51,16.42,18,52,28,57,34,44,43,52,55,53,42,47,56.57,44,16,2,12，9,40,23,56,3,39,16,54,36,52,5,53,48.23,47,41,49,…,22,42,10,16,53,58,40,1,43,56,40，32,44,35,37,45,52,56,40,2,23,49,50,26,11,35,32,34,58,6,52,26,31,23,4,52,53,19];

Compare ICALO with the basic ALO, WALO and HALO to optimize the 0~1 knapsack problem, and the test results are shown in Table 5.

Table 5 Comparison table of 0~1 knapsack problem results

Algorithm	Optimal	Worst	average	variance
ALO	4 327	3 932	4 193.5	8 865.426
ICALO	5 187	5 020	5 146.6	4 647.062 1
ICA	5 023	4 378	4 878.7	6 786.526
WALO	5 150	4 026	4 735.3	10 318.159

It can be obtained from Table 5 and the convergence curve that although the optimal value of WALO is higher, the difference and variance obtained are worse than that of ICALO, indicating that WALO has poor algorithm stability in the 0~1 knapsack problem, and combined with the convergence curve It can also be seen that in the 169th and 432nd generations, it takes a long time to fall into the game, and it is not easy to jump out of the game. ICA has better results than ALO. It can be seen from the variance that the algorithm is also more stable, but the results obtained by ICA The value is worse than WALO and ALO. Combined with the image, it can be seen that the convergence is slower. It can only converge until 412 iterations. However, ICALO has already converged in the 87th generation. The optimal value is 5187, which is better than other algorithms.

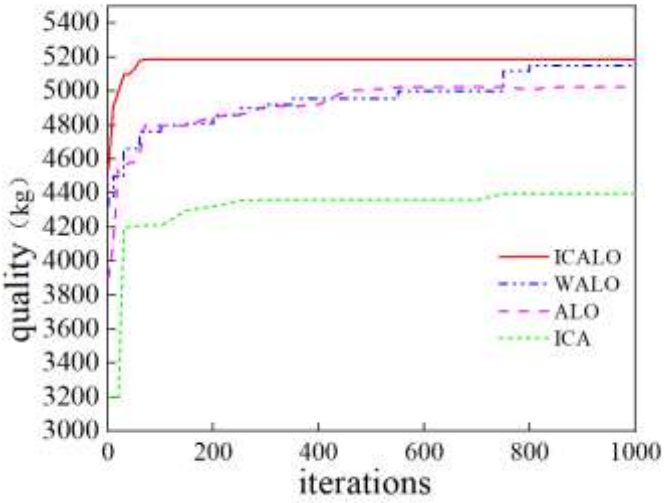


Figure 4 Convergence curve of 0~1 backpack

In summary, ICALO is more stable in the 0~1 knapsack problem, can get better values, and has a better ability to jump out of the game.

5 Truss structure optimization

5.1 Optimization of truss size

A typical spatial truss structure model is selected to verify the feasibility of ICALO in solving truss optimization problems with constraints. The truss optimization of ICALO is analyzed and compared with the optimization results of the basic ALO, WALO and ICA algorithms. By using MATLAB2020a software to realize ICALO, the algorithm is calculated 10 times, and the average value is used as the result value of the calculation. The value of each algorithm parameter is the same as in the third section.

1) Optimization application of truss size

Calculation example 1: 600-bar space truss structure is shown in Figure 5. Material elastic modulus E=6.895×10^7 kpa; material density ρ=2768 kg·m^3. Constraints: The maximum displacement in the x and y directions of all nodes shall not exceed 6.35 mm; the maximum allowable stress of each member is ±172.375 MPa.

The objective function is as follows

$$W(A)=\sum_{(i=1)}^{n}\rho A_i l_i \tag{19}$$

Where A_i is the cross-sectional area l_i of the i-th unit, the length of the i-th row, and ρ is the material density.

Considering the strength and stability of the truss, the stress constraint and displacement constraint conditions are as follows:

$$\sigma_{ik}\le\overline{\sigma_i}(i=1,2,...,N;K=1,2,...,P) \tag{20}$$

$$\mu_{jik}\le\overline{\mu_{jl}}(j=1,2,...,m;l=1,2,3;K=1,2,...,P) \tag{21}$$

Where σ_{ik} is the stress of the i-th element under the k-th load condition, μ_{jik} is the displacement of the

node j in the direction l under the k-th load condition, and $\overline{\sigma_i}$ and μ_{jl} are the allowable stress value of the i-th element and the node j in the k-th load condition. The allowable value of displacement in the l direction, P is the total number of load cases, and m is the total number of nodes of the structure.

Therefore, it can be classified as the following mathematical model:

$$\min . f(\mathrm{A}) \tag{22}$$

$$s.t. g_j(\mathrm{A}, \mu, \sigma) \le 0 \quad (\mathrm{j} = 1, 2, ..., \mathrm{J}) \tag{23}$$

$$A_i^l \le A_i \le A_i^u \tag{24}$$

Among them, A_i^l and A_i^u distribution are the maximum and minimum values of the cross-section.

Refer to Table 5 for the substructure of the members. The initial setting of the ICALO algorithm: 200 for the initial population; 500 for the number of iterations. Because the data analysis time of the 600-bar space truss structure is relatively long, and the load condition of 10KN at each non-fixed node has an impact on the structure, here is only a comparison with related algorithms and literature.

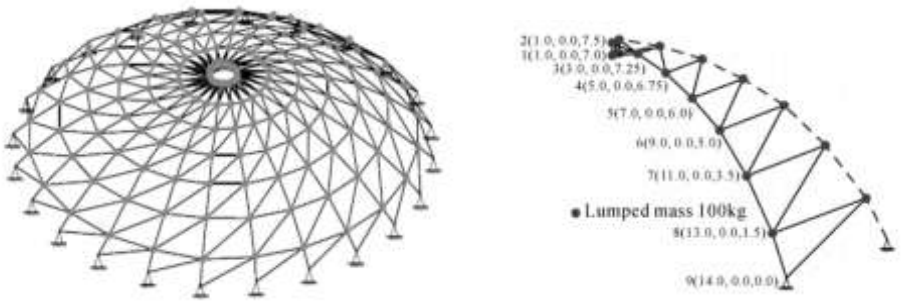


Figure 5 600-pole space truss structure

Table 6 10 random optimization results

Triangle variable number	Cross-sectional area /mm^2				
	ICALO	ALO	HALO	ICA	WALO
1	72.155	102.523	91.523	101.355	88.258
2	84.161	115.15	86.073	145.548	132.967
3	113.526	241.351	175.395	164.258	120.258
4	194.46	175.26	145.249	167.290	191.999
5	304.238	346.645	345.528	326.903	170.516
6	326.415	321.782	331.357	335.483	353.548
7	364.516	464.625	364.412	364.516	364.516
8	368.151	465.234	364.412	464.516	397.355
9	319.15	425.156	372.382	325.869	313.999

The optimized structure of the algorithm in this paper is 6347.65 kg; the optimized result of ALO is 7215.12 kg; the optimized result of HALO is 6698.14 kg. The masses of ICA and WALO are 6725.43 kg and 6,451.78 kg, respectively.

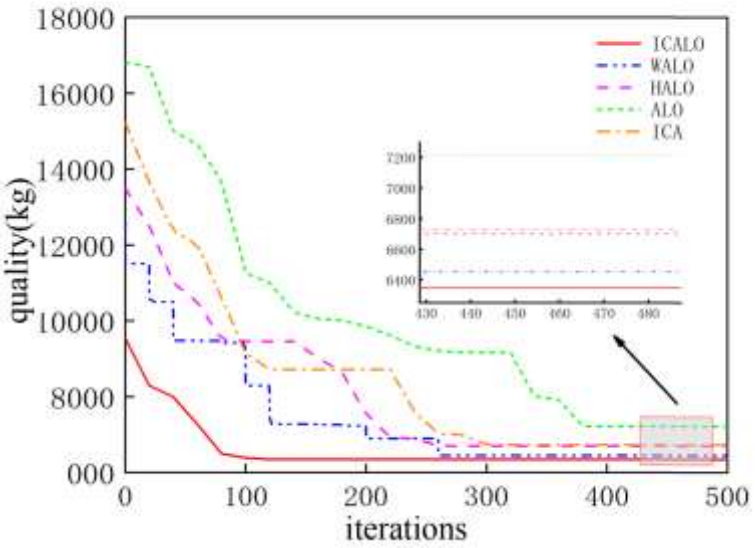


Figure 6 Convergence curve of 600-bar space truss

Figure 6 Convergence curve of 600-bar space truss. The iterative convergence process is shown in Figure 6. It can be seen that the iteration frequency before the 27th generation is low, and the iteration amplitude is large. After the 32 generations, the iteration frequency is accelerated, the iteration amplitude is reduced, and the iteration gradually stabilizes. The 134 generation converges near the optimal solution. The analysis of the above figure can reflect the improved Antlion The algorithm has the advantages of fast convergence speed and high convergence accuracy in solving truss problems. Moreover, the addition of immune cloning algorithm and Cauchy mutation can reduce the time waste of ants in the local optimization area, and can greatly improve the convergence speed of the algorithm. The active level of ants and the diversity of the population make it possible to get out of the game, so the optimization degree is greatly improved compared with the basic Antlion algorithm.

5.2 Optimization of dynamic characteristics of truss structure

Calculation example 1: The 18-bar plane truss structure is shown in Figure 7. The 18-bar plane truss contains 11 nodes, and each non-fixed node of the structure has a load of 10 kg. This truss has 13 design variables, of which 4 node position variables are continuous. The required accuracy error is 10^{-2} cm and 10^{-2} m, of which the upper and lower limits of 18 sections are 1 cm^2 and 10 cm^2. The elastic modulus of the rod material is E=210 Gpa, and the material density is 7800 kg/m^3.

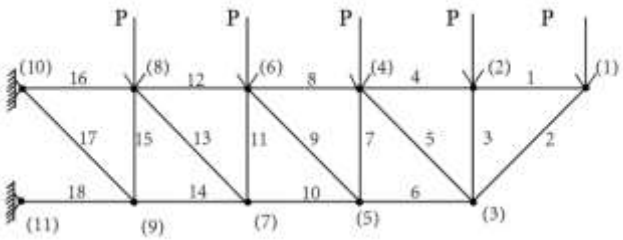


Figure 7 18-pole plane truss structure

The objective function is as follows

$$W(A)=\sum_{(i=1)}^{n}\rho A_i l_i+QM \qquad (Q=0 \text{ or } 1) \tag{25}$$

In this paper, the design variables increase the node position variables, and the coupling effect between the dimensions of each member is processed, and the members are constrained as follows:

$$.t.1\begin{cases} g_i^{\sigma}(A_i,\overline{X})=[\sigma_i]-\sigma_i\geq 0(i=1,2,...,N) \\ g_{jl}^{N}(A_i,\overline{X})=[\mu_{jl}]-\mu_{jl}\geq 0(j=1,2,...,m\ \ l=1,2,...,m) \end{cases} \tag{26}$$

Where $g_i^{\sigma}(A_i,\overline{X})$ is the stress constraint and $g_{jl}^{N}(A_i,\overline{X})$ is the displacement constraint.

The dynamic constraints are as follows:

$$s.t.2\begin{cases} \omega_k\geq\omega_{k\max}^{*} \\ \omega_k\leq\omega_{k\min}^{*} \end{cases} \tag{27}$$

Because the optimization of the truss structure is more complicated, it is only compared with ALO, ICA, and WALO. The initial settings of the ICALO algorithm: 200 individuals in the initial population; 500 iterations. The rest of the algorithm parameter settings are the same as above, the 10 random optimization results are shown in Table 7, and the final optimization graph is shown in Figure 9.

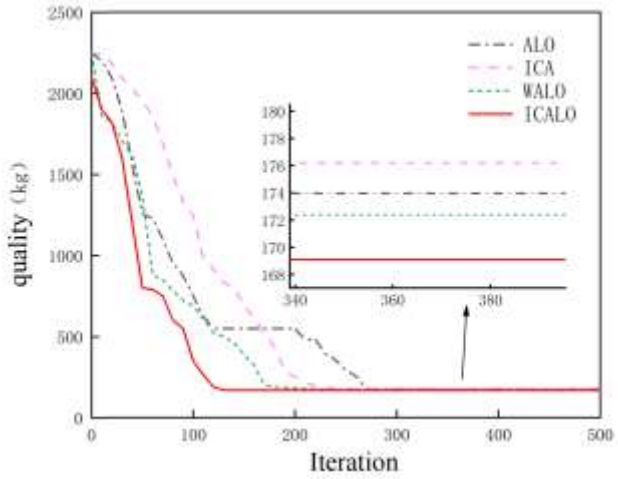


Figure 8 Convergence curve of 18-bar plane truss

Table 7 Comparison of 10 random shape optimization results

Variable number	ICALO	ALO	ICA	WALO
A_1	3.92	6.21	6.32	2.87
A_2	2.65	3.21	2.12	2.47
A_3	1.01	2.64	3.21	1.21
A_4	3.84	4.54	5.65	3.64
A_5	2.14	1.21	2.32	2.05
A_6	3.25	3.02	2.65	2.97
A_7	2.10	1.05	0.98	1.14
A_8	3.47	3.89	3.68	3.54
A_9	2.14	3.10	3.00	1.12
A_{10}	4.21	3.55	2.99	3.84
A_{11}	1.87	1.22	1.35	2.66
A_{12}	3.74	3.88	3.68	3.87
A_{13}	2.14	2.54	2.66	2.04

A_{14}	3.54	3.21	3.54	2.98
A_{15}	2.00	2.55	2.31	3.21
A_{16}	3.64	3.87	4.10	3.65
A_{17}	2.78	3.54	2.95	2.68
A_{18}	4.62	6.75	5.96	3.99
Truss quality （kg）	169.51	173.94	176.24	172.46

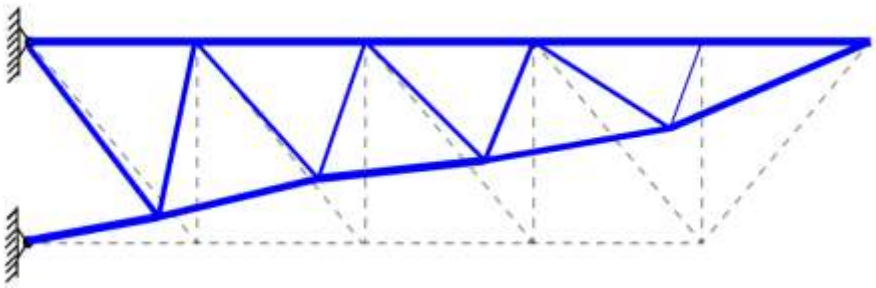


Figure 9. Plane 18-bar truss structure after ICALO layout optimization

It can be seen from Table 7 that the results obtained by the ICALO algorithm are better than those of ALO, WALO, and ICA, and the average value and the optimal value are significantly improved. From Table 11, it can be seen that the optimal solution phase of the ICALO algorithm Compared with the ALO, ICA, and WALO algorithms, are 4.43 kg, 6.73 kg, and 3.95 kg lighter, respectively.

It can be seen in Figure 8 that compared to ALO, ICALO converges faster, and ICALO reached the optimal value near the 114th generation, while ALO and WALO tended to the optimal value in the 283 and 178 generations, respectively. The accuracy of ICALO is also obvious. Better than other algorithms. The structure diagram of the optimized space truss is shown in Figure 9.

Example 2: The 145-bar plane truss structure is shown in Figure 10. There are fixed constraints on the periphery of the structure, and each non-fixed node has a load of 100 kg. The truss has 145 design variables, of which 56 node position variables are continuous. The accuracy error is required to be within 10^{-2} m. The upper and lower limits of the rod section are 1 cm^2 and 200 cm^2.

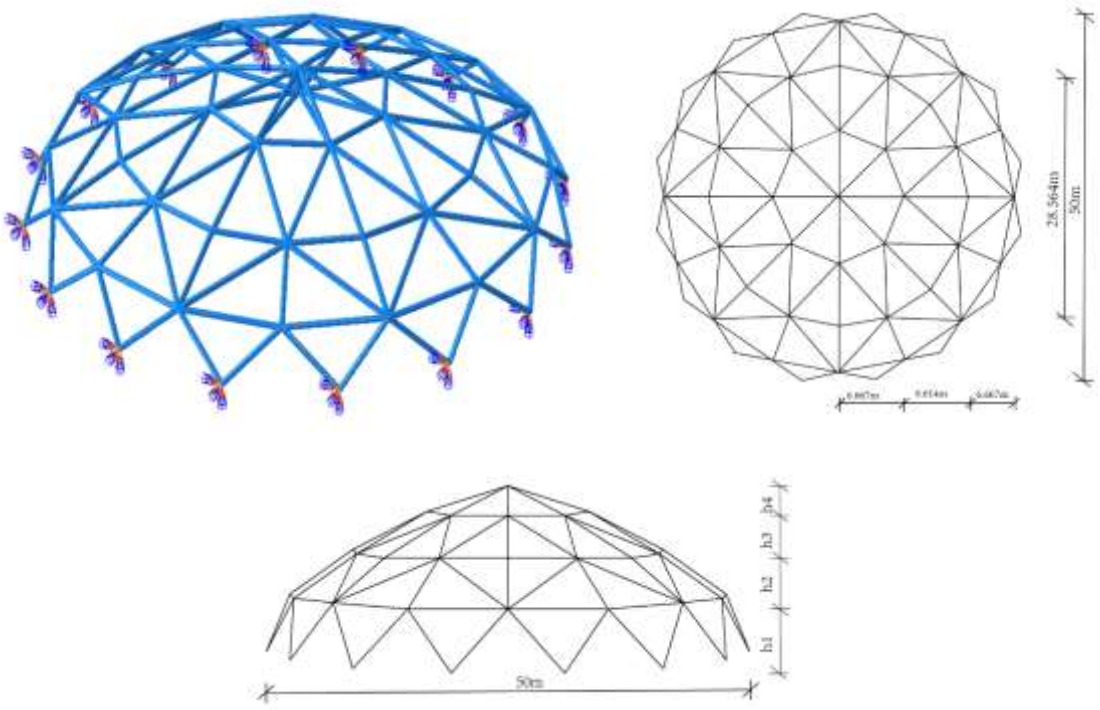


Figure 10 145-bar space truss structure diagram

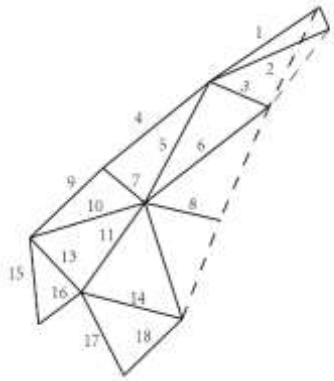


Figure 11 Standard module and truss grouping

This optimization considers three design variables

1) Consider the truss cross-sectional area and height variables

The height constraints of each layer are as follows:

$$1.56m \le h_4 \le 3.98m, 3.65m \le h_3 \le 6.12m, 2.35m \le h_2 \le 9.14m, 1.23m \le h_1 \le 4.32m$$

2) Consider truss cross-sectional area and layout variables

(The layout variable refers to the design of the distance between adjacent nodes at the same height. The number of rods in each layer is less than 48. Since the change of the node position of each layer will cause the total load to change, the load of each layer is set to a total amount. Evenly distributed to each node.)

3) Consider the cross-sectional area of the truss, layout variables and layout variables (height and layout constraints are the same as above)

When the truss is optimized, it needs to be symmetrically divided into 8 groups according to the left and right

symmetry, each group of 18 rods, as shown in Figure 10, of which 12 node positions are variable, and the displacement can be in the three directions of x, y, and z. Move, the accuracy is within 1 cm. The first constraint frequency of the structure is $6.92\ HZ \le \omega_1 \le 13.84$, and the second natural frequency is $\omega_2 \le 28.65\ HZ$. To study the quality of the structure at different frequencies, the results of the structure frequency $13.84 HZ \le \omega_1 + \omega_2 \le 35.57\ HZ$ and $35.57 HZ \le \omega_1 + \omega_2 \le 57.84\ HZ$ are compared. The optimization results and deformation diagrams of the truss are as follows:

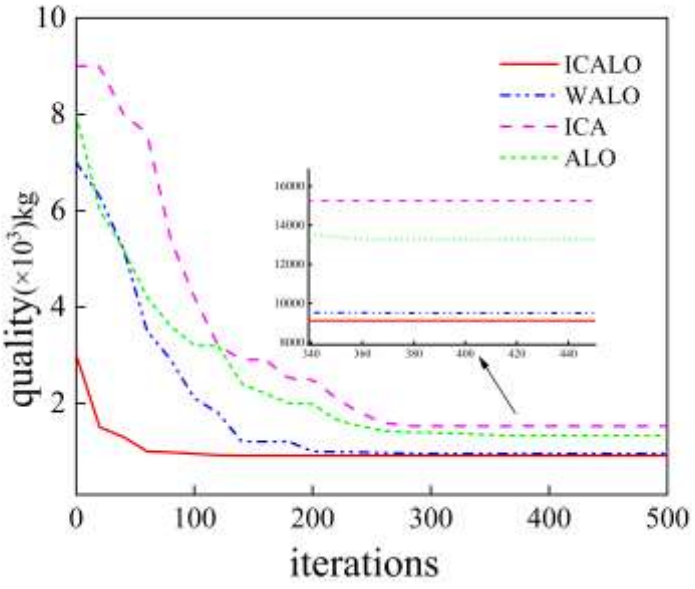


Figure 12 Convergence curve of highly optimized 145-bar space truss

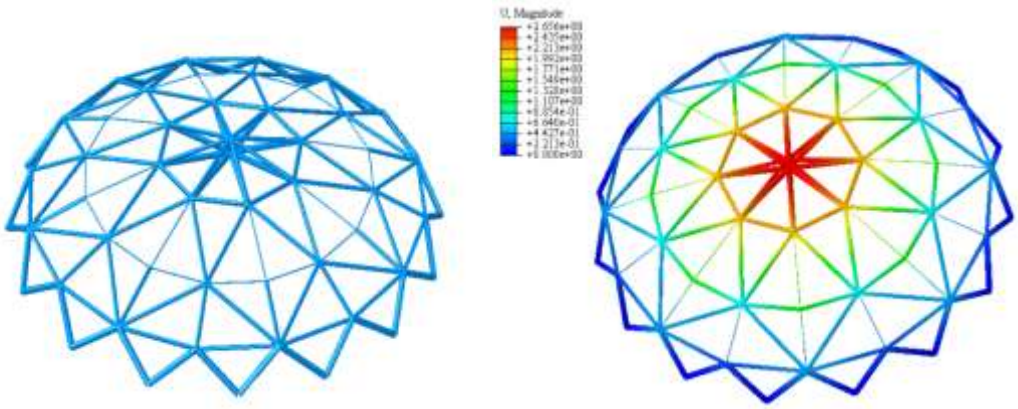


Figure 13. ICALO's highly optimized space truss structure and displacement diagram

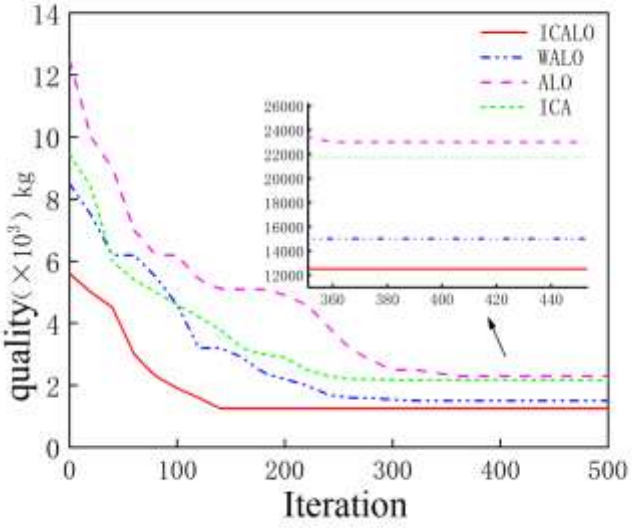


Figure 14 Convergence curve of 145-bar space truss after layout optimization

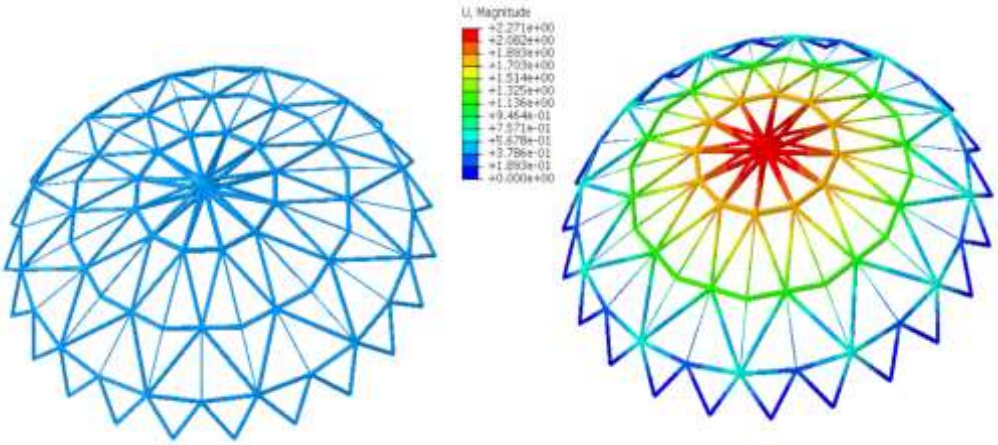


Figure 15. Space truss structure and displacement diagram after ICALO layout optimization

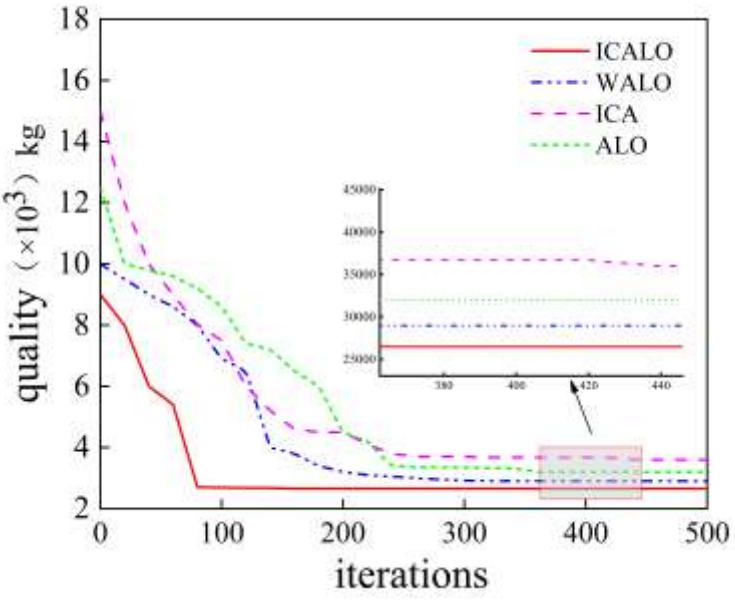


Figure 16 Convergence curve of low-frequency 145-bar space truss

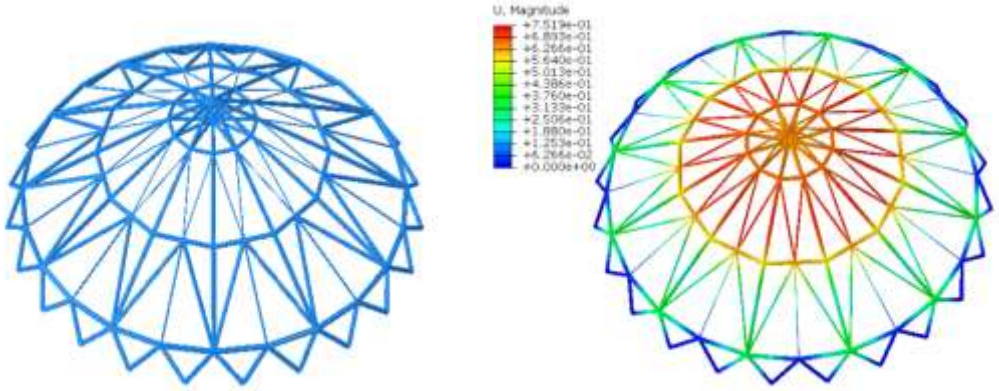


Figure 17 The truss structure and displacement diagram after low-frequency height and optimized layout

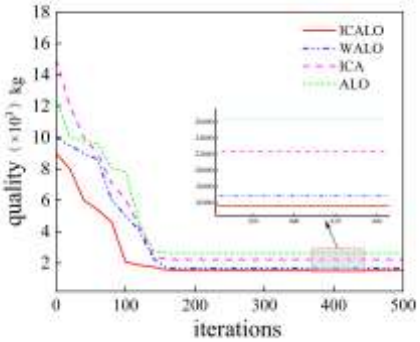


Fig.18 Convergence curve of high-frequency 145-bar space truss

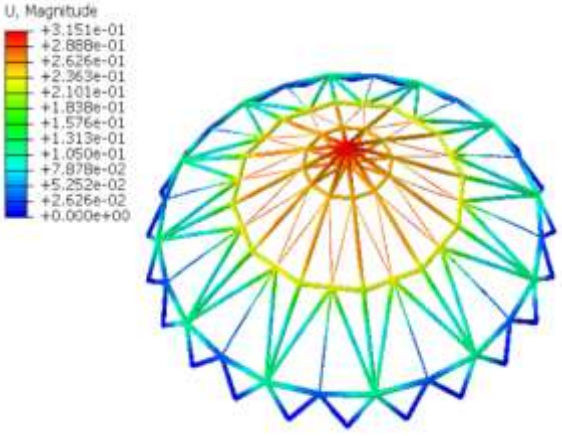


Fig.19 Deformation diagram of high-frequency truss after optimization

。Figures 12, 14, 16, and 18 are the optimization convergence curves of the truss structure under different

design variables with different algorithms. It can be seen from Table 8 that compared to WALO, ALO, ICA, ICALO has a smaller mass and is similar to the basic ALO. Ratio, the quality can be reduced by up to 80%, and it can be seen from the convergence curve that ICALO converges faster and can better jump out of the game.

Figures 13, 15, 17, and 19 are the final deformation diagrams of the truss optimization. It can be seen that the displacement of the truss is within 3cm after the load is applied. From Table 9, it can be seen that the frequency of ICALO is within the constrained range. The frequency of the overall truss exceeds the constraint range, indicating that the improved strategy in this paper can better adapt to the mechanical characteristics of the truss. And from the displacement diagram of the truss, it can be seen that the displacement of each layer of the members after the layout optimization is reduced by 15.04% compared with the height optimization, and the force of the members is better than that of the truss that only considers the cross section and height; from Figure 17, 19 It can be seen that after considering the frequency constraints, the displacement of the structure with a higher frequency than the low-frequency truss is optimized after optimization, the displacement is reduced by 58.04%, and the mass is increased by 30.61%. The displacement of each layer of the truss is better than that of the low-frequency truss. However, since the number of optimized single rods has been changed from 18 to 23, and the total weight of the truss has been changed from 9120kg to 12426kg, the mass difference between before and after optimization is too large. If the stability requirements of the truss are low, for economic reasons, The optimization is only optimized from variables such as height and cross-sectional area.

Table 8 Comparison of results of 10 times of shape optimization /kg

Scheme		Algorithm	Optimal	Worst	Average
1		ICALO	9120	10658	93218
		WALO	9504	12847	98115
		ICA	15243	20681	17944
		ALO	13265	16311	14218
2		ICALO	12426	14147	12988
		WALO	15521	17784	16358
		ICA	21657	25921	24581
		ALO	23482	25691	23648
3	$13.84HZ \le \omega_1+\omega_2 \le 35.57\ HZ$	ICALO	15628	18916	15968
		WALO	16814	17518	17214
		ICA	22351	26585	23814
		ALO	26289	28681	26948
	$35.57HZ \le \omega_1+\omega_2 \le 57.84\ HZ$	ICALO	22531	24518	22987
		WALO	29047	33629	32628
		ICA	36121	48817	39481
		ALO	32022	39281	36176

Table 9 Frequency of each order of optimization results

plan	Frequency order	ICALO	ALO
1	1	11.946	14.412
	2	28.649	28.653
	3	28.649	28.653
	4	28.679	28.697

2		1	13.156	16.614
		2	28.649	28.652
		3	28.649	28.652
		4	28.967	29.125
3	$13.84HZ \le \omega_1+\omega_2 \le 35.57\ HZ$	1	13.514	15.381
		2	22.056	26.654
		3	22.056	26.655
		4	22.075	29.745
	$35.57HZ \le \omega_1+\omega_2 \le 57.84\ HZ$	1	13.840	15.618
		2	28.649	28.652
		3	28.649	28.653
		4	28.501	28.662

6、Conclusion

(1) In this paper, the Antlion algorithm is easy to fall into the local optimum, and the later convergence speed is slow to solve the problem of low accuracy. An Antlion algorithm based on Kent chaos and Gaussian mutation is proposed. The improvement mechanism is to initialize the position of the ant by using Kent chaos mapping. Introduce the previous generation of elite ant lions as an ant lion position update method, improve the global search ability, change the adaptive weight and Gaussian mutation to reduce the influence of the poor ant lion on the ants in the later stage, and analyze it through 10 standard test functions and 0~1 knapsack problem The improved algorithm converges faster and is easier to get out of the game.

(2) In this paper, the cross-sectional area, height, and layout of the truss are used as design variables to constrain the stress, displacement and frequency of the truss structure. The ICALO algorithm is applied to the optimization of the truss structure, and the shape of the truss structure is determined by the optimal solution. After optimization, the truss structure shape and force-deformation diagram are obtained, which shows the feasibility of applying the improved Antlion algorithm to actual projects. However, how to use the algorithm to optimize the truss material and the time complexity of the algorithm itself is the direction that needs specific research and exploration in the next step.

Acknowledgements

This work was supported by the Innovative Funding Project for Graduate Students in Hebei Province. All data generated or analysed during this study are included in this published article [and its supplementary information files].

Authors' contributions

LIY and WY were involved in initiating the study and developing the model. LY drafted the initial manuscript and performed the analyses. LY and WY edited the manuscript. All authors read and approved the final manuscript.

Funding

National Science Fund subsidized project（51478311）

Project supported by the Natural Science Foundation of Hebei Province (E2020402079)

Science and Technology Research Project of Hebei Province Colleges and Universities (ZD2019114)

Availability of data and materials

All data generated or analyzed during this study are included in this published article.

Competing interests

The authors declare that they have no competing interests.

Author details

1 College of Civil Engineering, Hebei University of Engineering, Hebei University of Engineering, Handan, Heibei Province, China. 2 College of Civil Engineering, Hebei University of Engineering, Handan, Heibei Province, China.

(References)

[1] LI X Y, LI X W, LI J. Multi-objective rough set attribute reduction based on chaotic cluster intelligent optimization algorithm[J].Operations Research and Management,2020,29(12):30-37.

[2] YE X L, YAO J S, XIONG X, GONG C C, JIN T Y. Surface air temperature quality control algorithm based on cosine similarity and moving surface fitting[J].Chinese Science and Technology Paper,2020,15(05):577-584+592.

[3] Giuseppe Feo,Jyotsna Sharma,Stephen Cunningham. Integrating Fiber Optic Data in Numerical Reservoir Simulation Using Intelligent Optimization Workflow[J]. Sensors,2020,20(11):506-524.

[4] Zhiteng Ma,Xianfeng Yuan,Sen Han,Deyu Sun,Yan Ma. Improved Chaotic Particle Swarm Optimization Algorithm with More Symmetric Distribution for Numerical Function Optimization[J]. Symmetry,2019,11(7):106-110.

[5] Zhao K X, Huang C Q, Wang Y, al. UAV trajectory planning based on chaotic antlion algorithm[J]. Flight Mechanics, 2018, 36(1): 93-96.

[6] JIAN C F, WANG X F, WANG S. Research on Fault Diagnosis of Analog Circuit Based on ALO-SVM[J].Automation Technology and Application,2019,38(12):113-118.

[7] Haydar Kılıç,Uğur Yüzgeç, et al. Improved antlion optimization algorithm via tournament selection and its application to parallel machine scheduling[J]. Computers & Industrial Engineering,2019,132:166-186.

[8] LI Z N, WU W M, LIN Z Y. An image enhancement method using an improved Antlion optimization algorithm[J].Application Research of Computers,2018,35(04):1258-1260+1265

[9] WU W M, Zhang J J, Lin Z Y, ×10t al. Antlion algorithm with dual feedback mechanism[J]. Computer Engineering and Applications, 2017, 53(12): 31-35, 75.

[10] ZHAO X G, LIU D, JING K L. Noisy nonlinear system identification based on fusion of improved Antlion algorithm and T-S fuzzy model[J].Control and Decision,2019,34(04):759-766.

[11] WANG Y D, SHI Q, YOU Z F, Differential quasi-adversarial learning multi-objective Ant-lion algorithm_Wang Yadong[J]. Computer Engineering and Applications, 2020, 56(13): 156-163.

[12]XIAO Z Y, LIU S. Research on Elite Reverse Golden Sine Whale Algorithm and Its Engineering Optimization[J].Chinese Journal of Electronics,2019,47(10):2177-2186.

[13] Haydar Kılıç,Uğur Yüzgeç. Improved antlion optimization algorithm via tournament selection and its application to parallel machine scheduling[J]. Computers & Industrial Engineering,2019,132:166-186.

[14] Alaa Tharwat,Aboul Ella Hassanien. Chaotic antlion algorithm for parameter optimization of support vector machine[J]. Applied Intelligence,2018,48(3):670-686.

[15] ZU Y X, ZHOU J. Cognitive radio resource allocation based on combined chaotic genetic algorithm[J].Acta Physica Sinica,2011,60(07):894-901.

[16] Namhee Ryu,Seungjae Min, et al. Multiobjective optimization with an adaptive weight determination scheme using the concept of hyperplane[J]. International Journal for Numerical Methods in Engineering,2019,118(6):303-319.

[17] Àngel Calsina,Sílvia Cuadrado,Laurent Desvillettes,Gaël Raoul. Asymptotic profile in selection–mutation equations: Gauss versus Cauchy distributions[J]. Journal of Mathematical Analysis and Applications,2016,444(2):1515-1541.

[18] LIU J Q, DU W L, ZHANG F. Function complexity classification method based on particle swarm optimization[J].Control Engineering,2020,27(08):1337-1345.

[19] KANG J T, Zou L, CAO H Y, ZHANG Z. Optimal design of truss structure based on salvia group algorithm[J].Space Structure,2020,26(03):51-58+83.

[20] K.R. Subhashini, J.K. Satapathy, et al. Development of an Enhanced Ant Lion Optimization Algorithm and its Application in Antenna Array Synthesis[J]. Applied Soft Computing,2017,59:153-173.

[21] LIU J S, HUO Y, LI Y. Adaptive Antlion Optimization Algorithm Based on Optimization Strategy[J]. Pattern Recognition and Artificial Intelligence, 2020, 33(2): 121-132. (in Chinese)

[22] Meng Jia. Image encryption with cross colour field algorithm and improved cascade chaos systems[J]. IET Image Processing,2020,14(5):973-981.

[23] HUANG L Q, LIU H, WANG Z Y, WWANG J H. An adaptive image encryption algorithm combining chaotic mapping and DNA computing[J].Small Microcomputer System,2020,41(09):1959-1965.

[24] FANG P F,HUANG L G,LOU M M,JIANG K.Image encryption algorithm based on two-dimensional Logistic chaotic mapping and DNA sequence operations[J].Chinese Science and Technology Papers,2021,16(03):247-252.

[25] Milad Yousefi Valandar,Peyman Ayubi,Milad Jafari Barani. A new transform domain steganography based on modified logistic chaotic map for color images[J]. Journal of Information Security and Applications,2017,34:142-151.

[26] LU Y J, NI Z W, ZHU X H Xu L F, Wu Z J. Improved Discrete Firefly algorithm and Multifractal Attribute Reduction Method based on MapReduce [J]. Pattern recognition and artificial intelligence,2018,31(06):537-547.

[27] Mousumi Basu. Improved Particle Swarm Optimization for Global Optimization of Unimodal and Multimodal Functions[J]. Journal of The Institution of engineers (India): Series B,2016,97(4):525-535.

[28] SHI S, DU D S, WANG S G, LI W W. Multi-peak model of completely non-stationary ground motion based on Gaussian function and its application[J]. Journal of building structures,2020,41(05):198-206.

[29] Nima Moradi,Vahid Kayvanfar,Majid Rafiee. An efficient population-based simulated annealing algorithm for 0-1 knapsack problem[J]. engineering with Computers,2021,96(3)1245-1365.

[30] Wei Zhong Sun , Min Zhang , Jie Sheng Wang , Sha Sha Guo , Min Wang , Wen Kuo Hao. Binary Particle Swarm Optimization Algorithm Based on Z-shaped Probability Transfer Function to Solve 0-1 Knapsack Problem[J]. Binary Particle Swarm Optimization Algorithm Based on Z-shaped Probability Transfer Function to Solve 0-1 Knapsack Problem,2021,48(2):236-246.