

RESEARCH

Open Access



Improved fruit fly algorithm on structural optimization

Yancang Li^{1*}  and Muxuan Han²

Abstract

To improve the efficiency of the structural optimization design in truss calculation, an improved fruit fly optimization algorithm was proposed for truss structure optimization. The fruit fly optimization algorithm was a novel swarm intelligence algorithm. In the standard fruit fly optimization algorithm, it is difficult to solve the high-dimensional nonlinear optimization problem and easy to fall into the local optimum. To overcome the shortcomings of the basic fruit fly optimization algorithm, the immune algorithm self–non-self antigen recognition mechanism and the immune system learn–memory–forgetting knowledge processing mechanism were employed. The improved algorithm was introduced to the structural optimization. Optimization results and comparison with other algorithms show that the stability of improved fruit fly optimization algorithm is apparently improved and the efficiency is obviously remarkable. This study provides a more effective solution to structural optimization problems.

Keywords: Truss structure optimization, Fruit fly algorithm, Improvement, Immune response

1 Introduction

With the rapid development of computer technology, the efficiency of structural optimization is greatly improved, and structural designers can have more time and energy to consider how to get better structural design scheme. In 1974, Schimit and Farshi proposed to combine finite element theory with mathematical induction theory to solve the optimal weight problem of engineering structure. Structural optimization has stepped into a new era [1]. After that, the intelligent optimization algorithm is also widely applied to the structural optimization, and the modern structural optimization method is gradually applied to the engineering practice. Now, after years of research and development, structural optimization has changed from the original optimization of structure size to the present optimization of topology and further optimization of material distribution. From a single objective optimization problem, multiple objectives are optimized simultaneously. Azamirad and Arezoo [2] proposed

an improved software package for stamping die structure that can greatly reduce the weight. Ide [3] designed the lightweight structure with the structural optimization method, and successfully realized the lightweight gear box design by using the design method of reducing contact constraint stress. Kaveh [4] proposed water evaporation optimization algorithm (WEO), which is a population-based intelligent optimization algorithm inspired by physics and used for continuous structural optimization. In order to apply different optimization solvers to various finite-based structural topology optimization problems Rojas-Laband [5], developed a widely representative example library of mechanism design problems with minimum compliance, minimum volume and different sizes. Sivapuram [6] discussed various calculus methods and numerical methods commonly used to solve structural topology optimization problems.

Inspired by the foraging process of fruit flies, scholar Pan [7] proposed a more efficient swarm intelligence optimization algorithm in 2012: fruit fly optimization algorithm (FOA). The algorithm has the advantages of clear principle, fewer parameters and simple operation. However, there are also some shortcomings, such as weak ability to solve complex, high-dimensional and

*Correspondence: liyancang@hebeu.edu.cn

¹ College of Water Conservancy and Hydroelectric Power, Hebei University of Engineering, Handan, Heibei Province, China
Full list of author information is available at the end of the article

nonlinear optimization problems and easy to fall into local extremum. In view of the above typical problems, many scholars have proposed many improvements. As for the generation of candidate solutions [8], added an escape parameter that can be negative to the taste concentration judgment value, so that the candidate solution can take a negative value. A novel double strategy evolutionary fruit fly optimization algorithm (DSE-FOA) was proposed by Fang [9]. DSEFOA dynamically divided the fruit fly population into spermatogonium subgroups and ordinary subgroups, and adopted different strategies to update the evolution of drosophila at different levels of evolution, which improved the optimization ability of the whole population. In terms of search radius, Sang et al. [10] introduced adaptive parameters to adjust the search radius of drosophila so as to better balance the global search capability and local search capability. Cao [11] proposed to replace the traditional search mechanism with sector search mechanism, and the new fruit fly algorithm was generated to effectively improve the stability of the algorithm. In terms of flight strategy [12–14], added the operation of group collaboration and random perturbation to solve the problem of premature development of the algorithm. Sheng [15] proposed that the length of fruit fly search for Uber should be dynamically changed according to the change rate of concentration difference, which can effectively balance the global optimization ability and the local optimization ability. In terms of population diversity [16], divided the drosophila population into multiple subpopulations of the same size. Han [17] et al. proposed a dynamic twin group co-evolutionary FOA to improve the search accuracy. Xin [18] used the gaussian sampling method to update the fruit fly. This method can increase the chance of jumping out of the local extremum in the early stage of the algorithm and conduct more accurate search in the later stage.

Aiming at improving the performance of the standard fruit fly optimization algorithm, the hybrid algorithm was designed to combine with fusion immune response. Immune algorithm was fully utilized to improve the deficiency of fruit fly algorithm that is prone to fall into local extreme value in the later stage. In other words, when the number of evolutionary stasis steps t is greater than the threshold value of evolutionary stasis steps T , immune operation is performed to overcome the defect of basic fruit fly algorithm. The improved algorithm was proved to have better robustness and intelligence through standard functions and tests for solving 0–1 knapsack problems [19–21]. Finally, it was applied to the optimization of truss structure [22] and

compared with other algorithms to verify the feasibility of the improvements.

2 Basic fruit fly algorithm

FOA (fruit fly optimization algorithm) is a new heuristic algorithm that simulates the foraging activities of fruit flies in nature to seek the optimal solution of the objective function. The foraging iteration diagram of fruit flies is shown in Fig. 1.

The basic steps are as follows [23–26]:

Step 1: initialize parameters. Set *Sizepop* and *Maxgen* of the population size, and initialize the population position:

$$(X_{axis}, Y_{axis}). \tag{1}$$

Step 2: fruit fly searches in the olfactory system, which can make the search direction and the search step randomly. Random value (RV) is to be the search distance, and the position of the population is updated simultaneously:

$$\begin{cases} X_i = X_{axis} + RV \\ Y_i = Y_{axis} + RV \end{cases} \tag{2}$$

Step 3: since the exact location of the food is unknown, it is necessary to calculate the distance ($Dist_i$) between the fruit flies and the origin of the coordinate and then calculate the taste concentration parameter (S_i):

$$Dist_i = \sqrt{X_i^2 + Y_i^2} \tag{3}$$

$$S_i = \frac{1}{Dist_i} \tag{4}$$

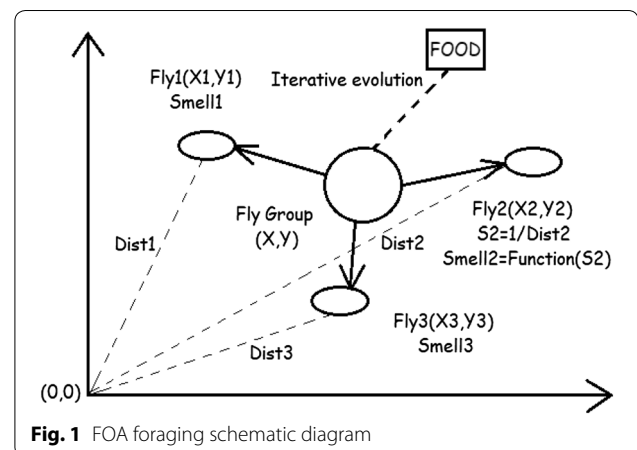


Fig. 1 FOA foraging schematic diagram

Step 4: substitute the fruit fly flavor concentration determination value (S_i) into the taste concentration decision function, the fitness function, then we will obtain the individual taste concentration of the fruit flies $Smell_i$.

$$Smell_i = \text{Fitness}(S_i). \quad (5)$$

Step 5: identify the individual with the highest flavor concentration in the drosophila population.

$$[\text{bestSmell}, \text{bestIndex}] = \min(\text{Smell}). \quad (6)$$

Step 6: retain the best flavor concentration value and coordinate, and other individuals in the population fly to this position:

$$\text{SmellBest} = \text{bestSmell}. \quad (7)$$

$$\begin{cases} X_{\text{-axis}} = X(\text{bestIndex}) \\ Y_{\text{-axis}} = Y(\text{bestIndex}) \end{cases}. \quad (8)$$

Step 7: termination condition, judge whether the concentration of the best position is better than that of the previous generation, and reach the maximum number of iterations; otherwise, skip step 2 to enter the iterative optimization.

3 Improved fruit fly algorithm with immune response

Since fruit fly has a relatively developed olfactory and visual system, it first carries out a wide range of search through the sense of smell and sends out food odor information to the surrounding drosophila in the process of foraging. When a fly is found to have a higher concentration by comparison, the individual flies will rely on visual function to fly to that location [27, 28]. It is precisely because of such population characteristics that the diversity of the population is reduced, the algorithm that simulates its characteristics, like other bionic intelligent algorithms, has the defect that it is easy to fall into local optimization, which leads to the problem of early maturity [29].

The immune algorithm was inspired by somatic cell theory and network theory [30]. And it can realize the function of self-regulation by generating different antibodies similar to the immune system [31]. This algorithm has strong local search ability. By using this, we can introduce it into the later stage of fruit fly algorithm execution to improve the basic fruit fly algorithm. The new algorithm IAFOA can be used to balance the deficiency of fruit fly algorithm that is prone to fall into local optimal, and improve the search efficiency.

3.1 Immune algorithm

Immune algorithm (IA) is a kind of bionic optimization algorithm. In 1990, Bersini [32] first used immune algorithm to solve problems. By simulating biological immune system identify antigen (objective function), simulation of the principle of the memory in the immune system, combination of antigen and antibody (optimization) solution, and diversity of imitation immune system, IA algorithm can realize the antigen recognition, cell differentiation, and memory of the immune system and self-regulating function [2, 33, 34]. The basic steps of IA are as follows:

Step 1: antigen recognition. Input objective function and constraint conditions as antigen of immune algorithm;

Step 2: generate the initial antibody. Generate initial antibodies randomly within the solution space;

Step 3: calculate the compatibility (fitness evaluation). According to the given fitness evaluation function, the affinities between antibodies and antigens and between antibodies and antibodies were determined. The compatibility between antigen and antibody A_v is defined as follows:

$$A_v = 1/[1 + O_{P_{t_v}}], \quad (9)$$

where $O_{P_{t_v}}$ represents the matching degree of antigen and antibody, and the value of A_v is between 0 and 1. When $O_{P_{t_v}} = 0$, $A_v = 1$, indicating that the antibody matches the antigen very well, that is, the antibody is the optimal solution.

Step 4: update memory units. The antibody with the highest affinity to antigen calculated in step 3 was added into the memory unit and replaced by the original antibody.

Step 5: promote and inhibit node production. Calculate the expected value E_{xi} of antibody i , the low expected value of antibody will be suppressed.

$$E_{xi} = A_i/C_i, \quad (10)$$

where A_i is the affinity between antigen and antibody i , and C_i is the number of antibody i ;

Step 6: generate new antibodies. The father generation produces the next generation antibody through heredity, mutation and crossover.

Step 7: whether termination conditions are met. Yes, stop the algorithm; No, skip to step 3.

3.2 Population diversity improvement of IAFOA

The immune system is the basic defense system to maintain the normal metabolism of living organisms by blocking the

invasion of bacteria [27]. The system recognizes gene types to produce different antibodies, promotes the emergence of new individuals and inhibits the excessive production of individuals through regulatory mechanisms, so as to achieve biological diversity [4].

Suppose the immune system has N antibodies, and each antibody has M genes, as shown in Fig. 2. The information entropy $H_j(N)$ of the j gene is:

$$H_j(N) = \sum_{i=1}^N (-P_{ij} \log P_{ij}). \tag{11}$$

If all the alleles of the antibody are the same at position j , then $H_j(N)$ is equal to 0.

Therefore, the average information entropy $H(N)$ of the system is as follows:

$$H(N) = \sum_{j=1}^M H_j(N)/M. \tag{12}$$

3.3 The implementation of IAFOA

Based on the standard fruit fly algorithm, FOA algorithm was first used to construct the feasible solution set. Then, the number of evolutionary stasis steps t was used as the trigger condition [35]. When $t > T$ (threshold of evolutionary stasis steps), the IA search process was invoked, and the process of immune factors in the immune algorithm (corresponding to individual drosophila) seeking antigens (corresponding to food source in the drosophila algorithm) producing antibodies (optimal solution) was used to expand the search space. The elite retention strategy was carried out in the feasible solution set obtained after IA search. And the obtained optimization solution was used together with the original feasible solution set to update the

concentration of odor information in the search space, so as to guide other fruit fly's path search mechanism.

The threshold value of evolutionary stagnation step T is the index to enter the IA algorithm. The premature introduction of IA algorithm is not conducive to the search ability of IA, or even the convergence of feasible solutions. Through multiple independent experiments, $T=6$ was adopted as the trigger value for entering IA algorithm.

If the trigger condition is satisfied, a fixed immune factor redistribution probability P^* was employed to the optimization of space. And the different individuals were given different adaptive immune probability $P(i)$ according to their fitness values:

$$P(i) = \frac{(\text{Smellbest} - \text{Smell}(i))}{((\text{Smellbest} - \text{Smellworst})P^*)} \tag{13}$$

$$\begin{cases} \text{if}(\text{bestSmell} < \text{Smell}(i)) \\ \text{Smellbest} = \text{bestSmell1} \\ \text{else}(t = t + 1) \\ \text{Smellbest} = \text{bestSmell2} \\ \text{end} \end{cases}, \tag{14}$$

where in order to avoid the algorithm falling into the local optimal solution, the initial probability of immune factor in IA algorithm was given $P^* = 0.25$ to randomly allocate it and increase the diversity of solutions.

The process of IAFOA is as follows:

- Step 1: initialize parameters. Set *Sizepop* and *Maxgen* of population size, initialize population positions X_{axis} and Y_{axis} , and the number of evolutionary stagnation steps $t=0$;
- Step 2: randomly generate fruit fly population according to Eq. (3);
- Step 3: use Eqs. (4)–(7) to operate the population;
- Step 4: record and retain the best flavor concentration value according to Eq. (14), and update the evolutionary iteration step number t ;
- Step 5: judge whether $t < T$ is true, if directly go to step 7; otherwise, according to Eq. (13), the adaptive immunity probability of individual fruit fly was calculated, and the immune operation was carried out according to the immune algorithm. For individuals who did not perform the immune operation, step 7 was taken.
- Step 6: repeat steps 2–4 for iterative search of the new population obtained by immunization;
- Step 7: set $gen = gen + 1$; if $gen < Maxgen$, go to step 2; otherwise terminate the iteration.

IAFOA flowchart is shown in Fig. 3.

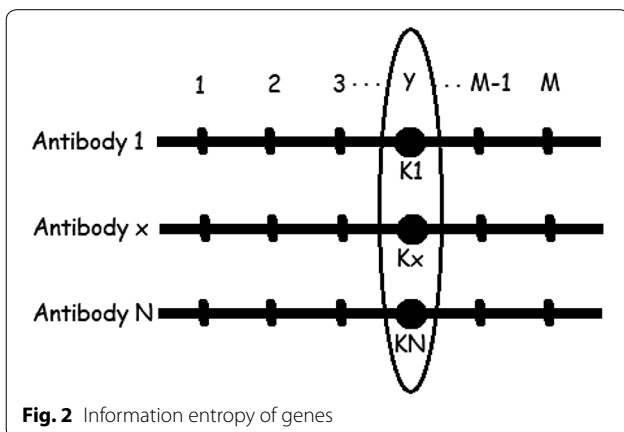


Fig. 2 Information entropy of genes

4 Simulation results

In order to verify the effectiveness of the improved algorithm, four standard test functions [5, 36] were selected for numerical simulation test to test the performance of IAFOA algorithm. In addition, 0–1 knapsack problem was selected for simulation experiment [37], and the results are compared with other algorithms. Initialization parameters: population size $Sizepop=200$, max iteration number $Maxgen=1000$, dimension $D=30$, $P^*=0.25$, $T=6$.

4.1 Standard function test

The first standard function: Rosenbrock function:

$$f_1(x) = \sum_{i=1}^{d-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]. \quad (15)$$

The second standard function: Ackley function:

$$f_2(x) = -a \exp \left(-b \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2} \right) - \exp \left(\frac{1}{d} \sum_{i=1}^d \cos(cx_i) \right) + a + \exp(1). \quad (16)$$

The third standard function: Cross-in-tray function:

$$f_3(x) = -0.0001 \left(\left| \sin(x_1) \sin(x_2) \exp \left(\left| 100 - \frac{\sqrt{x_1^2 x_2^2}}{\pi} \right| \right) + 1 \right| \right)^{0.1}. \quad (17)$$

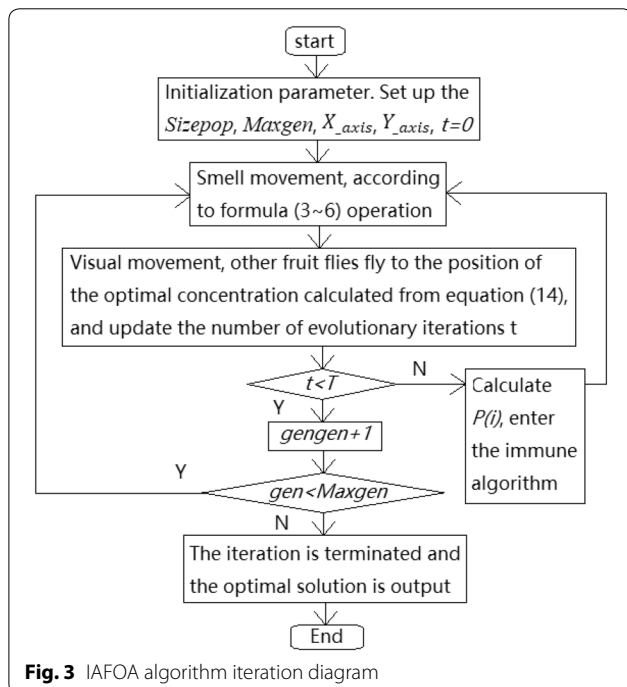


Fig. 3 IAFOA algorithm iteration diagram

The fourth standard function: Levy function:

$$f_4(x) = \sin^2(\pi \omega_1) + \sum_{i=1}^{d-1} (\omega_i - 1)^2 \left[1 + 10 \sin^2(\pi \omega_i + 1) \right] + (\omega_d + 1)^2 \left[1 + \sin^2(2\pi \omega_d) \right],$$

where $\omega_i = 1 + \frac{x_i - 1}{4}$,
for all $i = 1, 2, \dots, d$. (18)

Figures 4, 5, 6, and 7 show the graphs of the four functions, where Rosenbrock function is a single peak function, which is mainly used to test the convergence performance of the improved algorithm in the process of operation. Both Ackley function and Cross-in-tray function are complex multi-peak functions, which tend to make the algorithm fall into local optimization, so that the real optimal value cannot be obtained, which is used to test the ability of the improved algorithm to deal with falling into “premature”. The Levy function has a complex spatial property and is used to test the computational accuracy, convergence stability and time complexity of

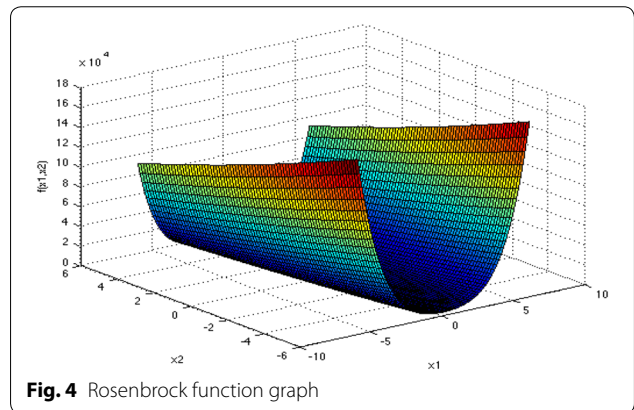


Fig. 4 Rosenbrock function graph

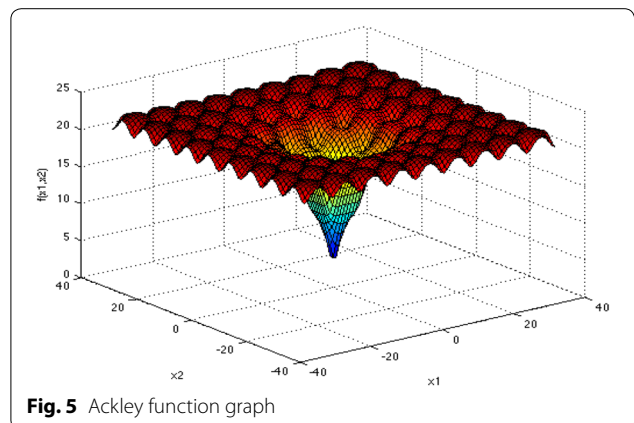


Fig. 5 Ackley function graph

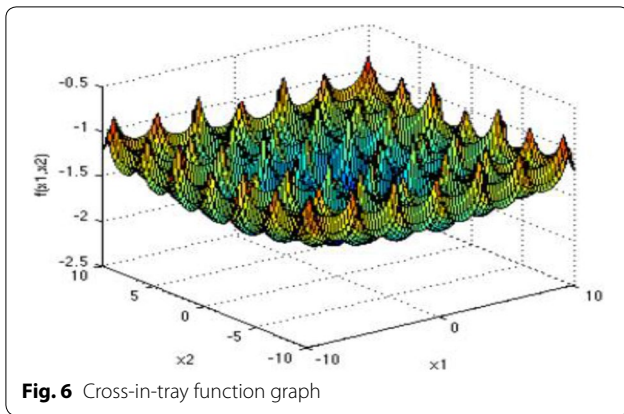


Fig. 6 Cross-in-tray function graph

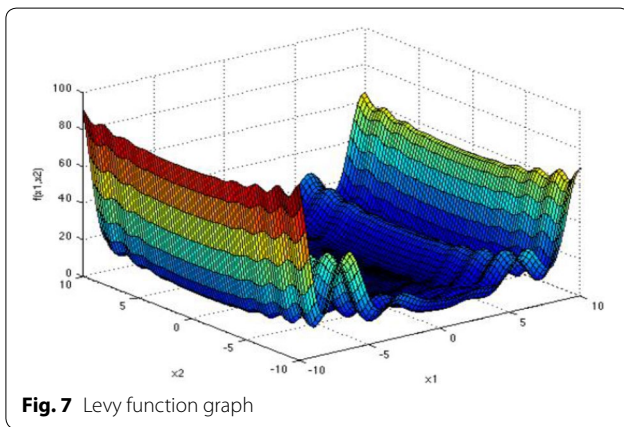


Fig. 7 Levy function graph

the improved algorithm. The selected functions and their expressions are shown in Table 1.

Four different standard functions were used to test the convergence performance, operation efficiency, handling of “local optimal” and “premature” problems, and the optimal value and average value in the optimization test. The standard deviation reflects the robustness of the algorithm. The running time reflects the convergence speed and accuracy of the algorithm. It can be seen from the calculation results in Table 2 that the fusion immune response hybrid fruit fly algorithm performs better than the standard fruit fly algorithm in different standard functions. The improved algorithm is feasible and effective.

In order to more intuitively see the effectiveness of improved algorithm performance, Fig. 8 shows immune algorithm (IA), particle swarm optimization (PSO), standard fly optimization algorithm. FOA) and fruit fly optimization algorithm based on immune algorithm (IAFOA) under the above four standard functions. The average value of multiple runs was used as the final result to avoid accidental errors.

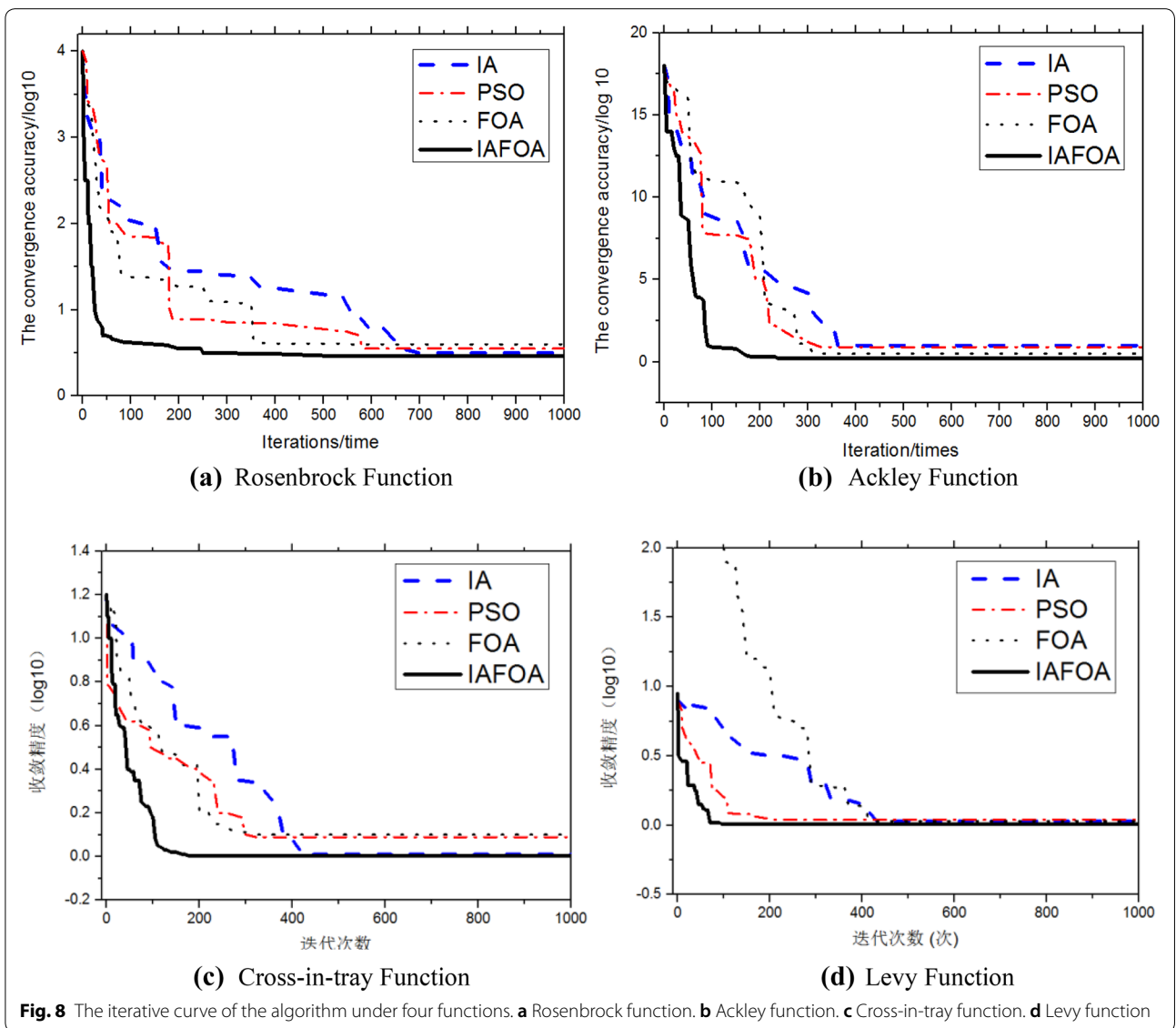
It can be seen from the iterative curve of Fig. 5 that the algorithm can search the local optimal solution, in addition it reflects high convergence speed and convergence accuracy, especially at the initial stage of iterative algorithm. Take (a) in Fig. 5 as an example, when the algorithm runs about 78 times, it can jump out of the local optimal solution. When the number of iterations is about

Table 1 Comparison of optimization performance for benchmark functions

Function	Dimension	Algorithm	Optimal value	Average value	Standard deviation	Running time/s
f_1	30	IA	37.1543	61.3698	33.1659	5.5329
		PSO	16.5998	34.6729	29.5267	3.9532
		FOA	15.2864	20.9341	19.5582	1.0299
		IAFOA	14.0958	17.9574	6.46931	1.3621
f_2	30	IA	2.2763	3.5297	3.061e−03	3.5146
		PSO	2.0051	2.1193	2.796e−03	4.2287
		FOA	1.6849	1.9678	3.909e−04	1.9652
		IAFOA	0	1.0052	6.9768e−03	0.9537
f_3	30	IA	2.18e−02	4.485e−02	3.0702e−01	4.7798
		PSO	1.96e−02	2.073e−01	2.785e−01	3.5669
		FOA	1.854e−01	1.6941e−01	1.1185e−01	1.2463
		IAFOA	0	0	0	0.6805
f_4	30	IA	1.952e−04	2.327e−04	6.001e−01	3.534
		PSO	1.439e−02	2.557e−02	5.026e−01	4.371
		FOA	2.005e−02	1.363e−01	3.778e−02	0.988
		IAFOA	3.958e−03	1.564e−05	2.563e−02	1.015

Table 2 Result comparison table of TSP problem

Algorithm	Oliver30		Att48		Eil51	
	Best	Avg	Best	Avg	Best	Avg
IA	459.9	485.1	394,025	38,732	564.8	579.4
PSO	527.3	572.3	499,758	52,277	570.9	610.2
FOA	415.2	485.6	35,947	44,620	453.6	557.9
IAFOA	415.0	475.9	35,098	43,082	450.9	544.1



255 times, the optimization result is generally stable. It is obviously that the improved algorithm proposed here is superior to the IA, PSO, and standard FOA.

4.2 Solve the TSP problem

Traveling salesman problem (TSP) is a classical combinatorial optimization problem, which is a typical NP hard problem. It has important research value in the field of logistics distribution and vehicle route planning.

The TSP problem can be described as: we know the coordinate position of N cities or the distance between two cities. A salesman needs to go to N cities to sell goods. To start from one city, salesman must pass through other cities only once, and then return to the starting city. Finding out the shortest path for salesman. The mathematical model is as follows.

If $S = (P_1, P_2, \dots, P_N)$ can satisfy the minimum value of $f(S)$, then access order S is the optimal solution of TSP problem under the condition of N cities, that is, the optimal path [38].

$$f(S) = \sum_{n=1}^{N-1} (d(P_i, P_{i+1})) + d(P_N, P_1), \quad (19)$$

where P_i is the city code, $i \in (1, 2, \dots, N)$; $d(P_i, P_j)$ is the distance between city P_i and city P_j . If only the city coordinates are known, then:

$$d(P_i, P_j) = \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}, \quad (20)$$

where $(X_i - X_j)$ is the coordinate position of P_i ; $d(P_i, P_j) = d(P_j, P_i)$.

In order to verify the performance of the IFOA in solving the TSP problem, this paper selected Oliver30, Att48 and Eil51 standard examples from the international TSP database (The Library of TSP, TSPLIB) for parameter testing. The result is shown in Table 2.

The above analysis results show that IAFOA can not only effectively solve the TSP problem, but also has a higher robustness than other algorithms.

5 Optimization of truss structure by IAFOA

5.1 Optimization model of truss structure

1. The optimization model

The optimization model problem of truss with sectional area as the design variable is described as follows [6]:

$$\min F = W(x) \quad (21)$$

$$\text{s.t. } g_i(x) \leq 0, \quad i = 1, 2, \dots, m, \quad (22)$$

where $g_i(x)$ is the constraint function; m is the number of constraints.

2. The objective function

$$W(A) = \sum_{i=1}^n \rho A_i L_i, \quad (23)$$

where $W(A)$ is the weight of the structure; A_i and L_i are, respectively, the cross-sectional area and length of the i th root. ρ is the density of the material; n is the number of design variables.

3. The constraint

Each member of the structure shall meet the requirements of strength, stiffness, stability and section size:

$$\frac{\sigma_i}{[\sigma]} - 1 \leq 0, \quad (24)$$

$$\frac{\mu_j}{\mu_{\max}} - 1 \leq 0, \quad (25)$$

$$A_{\min} \leq A_i \leq A_{\max}, \quad (26)$$

where σ_i is the normal stress of the i th root element. $[\sigma]$ is the allowable stress of the material. μ_j represents the displacement of node j ; allowable displacement of μ_{\max} node j ; A_{\min} and A_{\max} are the upper and lower limits of the member section, respectively.

Example 1 The 25-bar truss structure model [39] is established as shown in Fig. 9 and Table 3. Basic parameters of truss structure: rod length $L = 0.635$ m, elastic modulus $E = 6.895 \times 10^4$ MPa, material density $\rho = 2.678 \times 10^3$ kg/m³, allowable stress range: $[-275.8275.8]$, maximum vertical displacement $y = 8.889$ mm of nodes 1 and 2.

The results are shown in Table 4 and Fig. 10. Under the same constraint conditions, IAFOA was used to optimize the 25-bar truss structure, and the total mass of the optimized structure was 206.591 kg. Compared to IA, mass decreased $(246.436 - 206.591) / 206.591 = 1.93\%$. Compared with PSO, the quality decreased $(216.339 - 206.591) / 206.591 = 4.72\%$; compared with FOA, the optimization results were optimized $(214.702 - 206.591) / 206.591 = 3.93\%$. After about 68 iterations, IAFOA can find the global optimal solution. IA searched for the global optimal solution for about 130 times, PSO for about 150 times and FOA for about 165 times. For the quality problem of global optimal solution, IAFOA

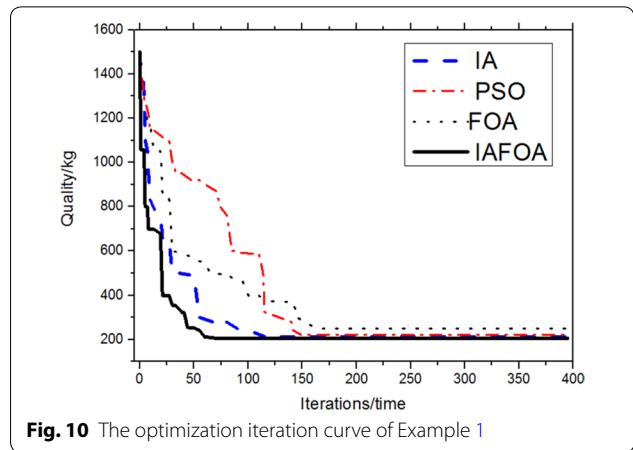
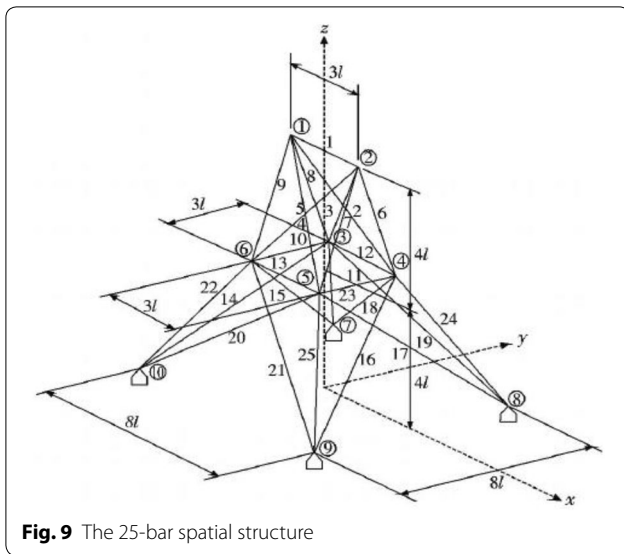
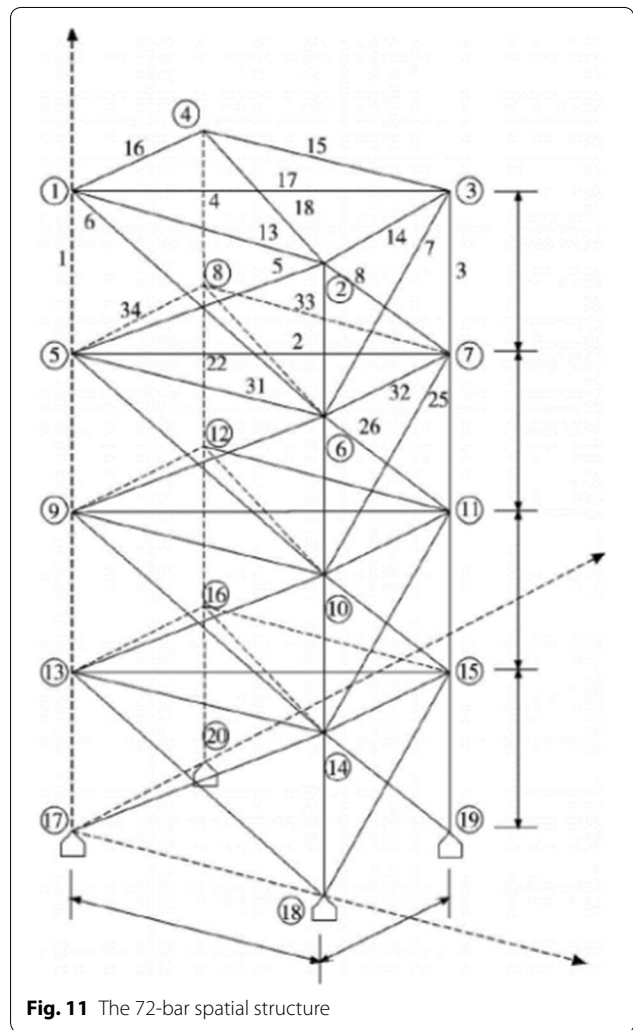


Table 3 Load conditions of 25-bar truss structure

Node number	F_x	F_y	F_z
1	4.448	44.482	-22.241
2	0	44.482	-22.241
3	22.241	0	0
6	22.241	0	0

Table 4 Comparison of optimization results of 25-bar truss structure

Bar number	Bar section area (unit: mm ²)			
	IA	PSO	FOA	IAFOA
A_1	65.7	64.516	64.9	64.1
A_2-A_5	242.6	228.5	234.5	226.3
A_6-A_9	2287.5	2237.6	2230.1	2240.4
$A_{10}-A_{11}$	65.12	64.516	63.149	63.56
$A_{12}-A_{13}$	1245.8	1227.9	1226.1	1223.5
$A_{14}-A_{17}$	505.1	506.9	501.7	500.9
$A_{18}-A_{21}$	92.1	83.9	89.5	85.7
$A_{22}-A_{25}$	2523.4	2575.7	2568.3	2495.6
Total weight (unit kg)	246.436	216.339	214.702	206.591



is superior to the other three optimization algorithms. It is proved that IAFOA is stable in the face of complex optimization problems in the optimization process, and

Table 5 The classification of 72-bar truss structure

Group number	Bar number	Group number	Bar number
A ₁	1, 2, 3, 4	A ₉	37, 38, 39, 40
A ₂	5, 6, 7, 8, 9, 10, 12	A ₁₀	41, 42, 43, 44, 45, 46, 47, 48
A ₃	13, 14, 15, 16	A ₁₁	49, 50, 51, 52
A ₄	17, 18	A ₁₂	53, 54
A ₅	19, 20, 21, 22	A ₁₃	55, 56, 57, 58
A ₆	23, 24, 25, 26, 27, 28, 29, 30	A ₁₄	59, 60, 61, 62, 63, 64, 65, 66
A ₇	31, 32, 33, 34	A ₁₅	67, 68, 69, 70
A ₈	35, 36	A ₁₆	71, 72

it is not easy to fall into the local optimal solution. Thus, IAFOA is effective.

Example 2 The 72-bar truss structure model [39] is established as shown in Fig. 11. 72 bars are divided into 16 groups, and the grouping condition is shown in Table 5. Material density is $\rho = 2.678 \times 10^3 \text{ kg/m}^3$, elastic modulus is $E = 6.895 \times 10^4 \text{ MPa}$, and the maximum displacement of each bar in all directions cannot exceed 6.35 mm, and maximum allowable stress is $[-172.375, 172.375]$, and the optimized results are shown in Table 6.

The results are shown in Table 7 and Fig. 12. Under the same constraint conditions, IAFOA

was used to optimize the 72-bar truss structure, and the total mass of the optimized structure was 171.98 kg. Compared to IA, mass decreased $(175.03 - 171.98)/171.98 = 1.77\%$. Compared with PSO, the quality decreased $(173.87 - 171.98)/171.98 = 1.10\%$; Compared with FOA, the optimization results were optimized $(176.31 - 171.98)/171.98 = 2.52\%$.

Example 3 The 200-bar truss structure model [39] is established as shown in Fig. 13. 200 bars are divided into 29 groups, and it is relatively high-dimensional optimization problem. The grouping condition is shown in Table 7. Material density is $\rho = 7.86 \times 10^3 \text{ kg/m}^3$, elastic modulus is $E = 2.1 \times 10^{11} \text{ N/m}^2$. The minimum permitted cross-sectional area for the truss members is taken as 0.1 cm^2 , a non-structural mass of 100 kg is attached for all free nodes.

Table 6 Comparison of optimal designs for the 72-bar spatial truss structure

Bar group number	Bar section area (unit: mm ²)			
	IA	PSO	FOA	IAFOA
A ₁	103.75	100.69	102.26	101.31
A ₂	358.36	359.28	372.97	349.15
A ₃	271.02	270.31	220.27	238.97
A ₄	367.88	368.96	392.01	398.29
A ₅	343.59	342.99	170.51	173.49
A ₆	337.23	337.12	353.46	336.89
A ₇	64.518	64.518	64.528	64.518
A ₈	64.518	64.518	64.518	64.518
A ₉	871.37	871.01	713.98	825.37
A ₁₀	318.96	318.55	373.81	332.31
A ₁₁	64.518	64.518	64.518	64.518
A ₁₂	64.518	64.518	64.518	64.518
A ₁₃	1188.59	1187.96	1330.02	1152.32
A ₁₄	324.97	325.72	324.89	332.61
A ₁₅	64.518	64.518	64.518	64.518
A ₁₆	64.518	64.518	64.518	64.518
Weight (kg)	175.03	173.87	176.31	171.98

The results are shown in Table 8 and Fig. 14. Under the same constraint conditions, IAFOA was used to optimize the 200-bar truss structure, and the total mass of the optimized structure was 2156.05 kg. Compared to IA, mass decreased $(2298.73 - 2156.05)/2156.05 = 6.62\%$. Compared with PSO, the quality decreased $(2276.59 - 2156.05)/2156.05 = 5.59\%$; compared with FOA, the optimization results were optimized $(2259.86 - 2156.05)/2156.05 = 4.81\%$.

6 Conclusion

With the development of civil engineering, the structural optimization is more and more important. To find a new method for the truss structure optimization, an improved FOA was proposed. To overcome the shortage of the basic FOA, the IA was introduced into FOA, and a new IAFOA algorithm was proposed. Combining the global search capability of the original standard FOA algorithm and the strong local search capability of IA algorithm itself, the IAFOA achieves

Table 7 The classification of 200-bar truss structure

Group name	Bar number	Group name	Bar number
A ₁	1, 2, 3, 4	A ₁₆	82, 83, 85, 86, 88, 89, 91, 92, 103, 104, 106, 107, 109, 110, 112, 113
A ₂	5, 6, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17	A ₁₇	115, 116, 117, 118
A ₃	19, 20, 21, 22, 23, 24	A ₁₈	119, 122, 125, 128, 131
A ₄	18, 25, 56, 63, 94, 101, 132, 170, 177	A ₁₉	133, 134, 135, 136, 137, 138
A ₅	26, 29, 32, 35, 38	A ₂₀	140, 143, 146, 149, 152
A ₆	6, 7, 9, 10, 12, 13, 15, 16, 27, 28, 30, 31, 33, 34	A ₂₁	120, 121, 123, 124, 126, 127, 129, 130, 141, 142, 144, 145, 147, 148, 150, 151
A ₇	39, 40, 41, 42	A ₂₂	153, 154, 155, 156
A ₈	43, 46, 49, 52, 55	A ₂₃	157, 160, 163, 166, 169
A ₉	57, 58, 59, 60, 61, 62	A ₂₄	171, 172, 173, 174, 175, 176
A ₁₀	64, 67, 70, 73, 76	A ₂₅	178, 181, 184, 187, 190
A ₁₁	44, 45, 47, 48, 50, 51, 53, 54, 65, 66, 68, 69, 71, 72, 74, 75	A ₂₆	158, 159, 161, 162, 164, 165, 167, 168, 179, 180, 182, 183, 185, 186, 188, 189
A ₁₂	77, 78, 79, 80	A ₂₇	191, 192, 193, 194
A ₁₃	81, 84, 87, 90, 93	A ₂₈	195, 197, 198, 200
A ₁₄	95, 96, 97, 98, 99, 100	A ₂₉	196, 199
A ₁₅	102, 105, 108, 111, 114		

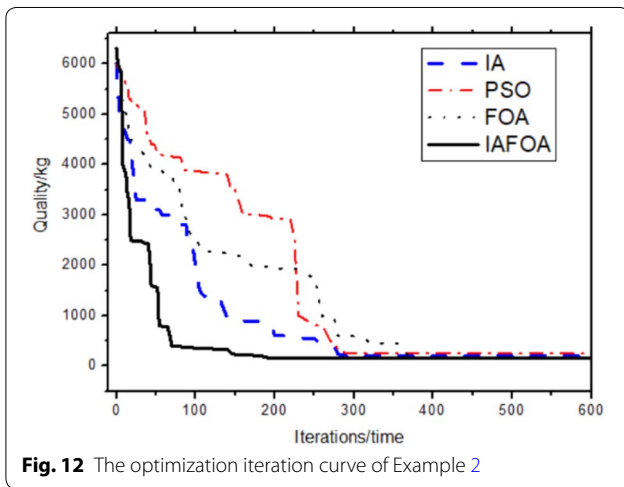


Fig. 12 The optimization iteration curve of Example 2

good results in numerical simulation with high ability of solving optimization problems. Although these improvements can overcome some shortcomings of the algorithm, it lacks the adjustment of parameter stagnation step number wide value T and fixed probability P^* . In the future, more detailed improvements are needed to make IAFOA algorithm converge faster and more accurately. In addition, the example in this paper is relatively simple, and the application of the algorithm in the optimization of complex structures needs to be further studied.

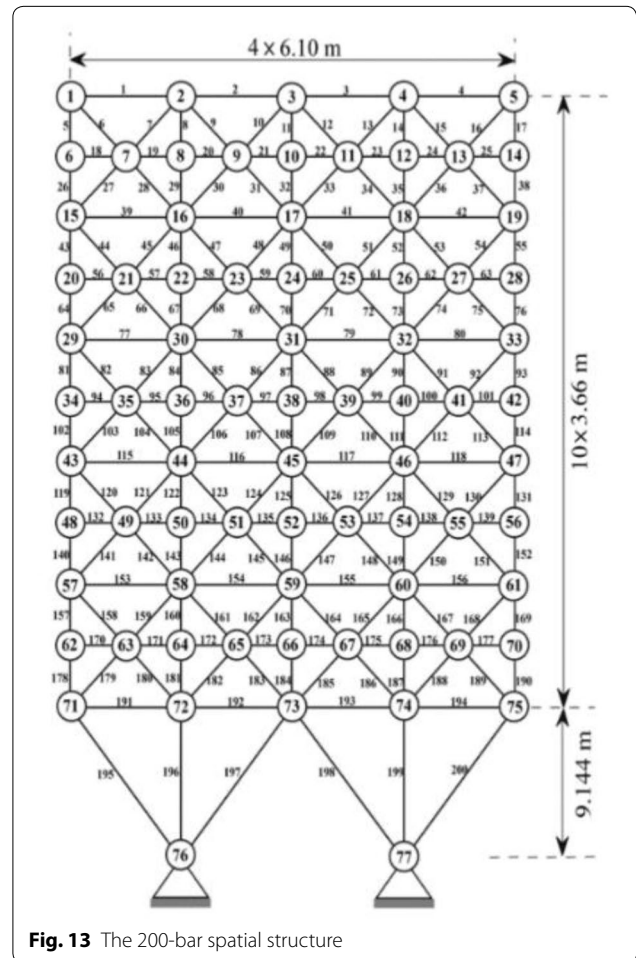


Fig. 13 The 200-bar spatial structure

Table 8 Comparison of optimal designs for the 200-bar spatial truss structure

Bar group number	Bar section area (unit: m ²)			
	IA	PSO	FOA	IAFOA
A ₁	0.2998	0.3078	0.4439	0.2879
A ₂	0.5564	0.4571	1.0438	0.4853
A ₃	0.2867	0.1006	0.3769	0.1002
A ₄	0.1908	0.1006	0.1494	0.1002
A ₅	0.8329	0.5093	0.4835	0.4985
A ₆	0.6598	0.8277	0.8103	0.8038
A ₇	0.1779	0.1051	0.2364	0.1031
A ₈	1.4794	1.4339	1.4554	1.3769
A ₉	0.4462	0.1013	1.0103	0.1000
A ₁₀	1.4566	1.5497	2.1382	1.5541
A ₁₁	1.2295	1.1535	0.8583	1.1507
A ₁₂	0.2794	0.1469	1.2718	0.1332
A ₁₃	1.9279	2.9492	2.0807	3.0109
A ₁₄	0.1179	0.1111	0.2677	0.1005
A ₁₅	3.4435	3.2199	3.2403	3.2584
A ₁₆	1.3953	1.5796	2.0098	1.6205
A ₁₇	0.6288	0.2901	1.5956	0.2079
A ₁₈	4.8937	5.1002	6.2338	5.0203
A ₁₉	0.6093	0.1101	0.5793	0.1328
A ₂₀	5.4397	5.4594	3.0520	4.8531
A ₂₁	1.8439	2.9007	1.8121	2.0138
A ₂₂	1.4273	0.6489	1.2986	0.7201
A ₂₃	8.1702	7.6592	5.8810	7.7219
A ₂₄	0.3271	0.1379	0.2324	0.1819
A ₂₅	10.902	8.0097	7.7536	8.9438
A ₂₆	2.9359	2.7602	2.6871	2,9953
A ₂₇	19.5329	10.4765	10.2694	10.2049
A ₂₈	20.9768	21.3501	21.5704	20.6507
A ₂₉	15.7301	10.5014	8.2901	11.5468
Weight (kg)	2298.73	2276.59	2259.86	2156.05

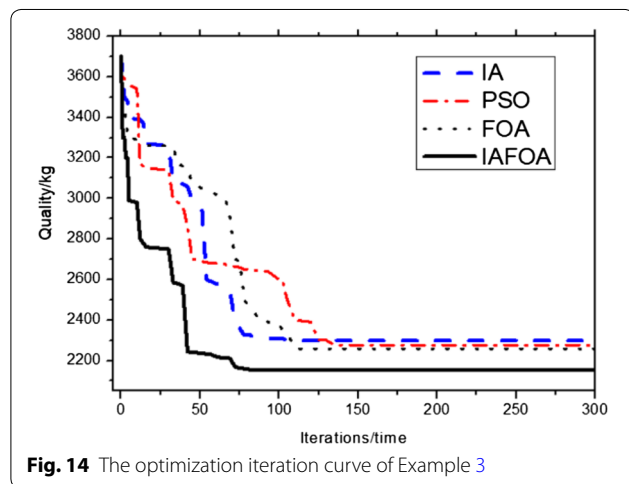


Fig. 14 The optimization iteration curve of Example 3

Acknowledgements

This work was supported by the Project of Scientific Research Program of Colleges and Universities in Hebei Province (No. ZD 2019114) and the Innovative Funding Project for Graduate Students in Hebei Province (No. CXZZSS2020079).

Authors' contributions

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

Funding

This work was supported by the Project of Scientific Research Program of Colleges and Universities in Hebei Province (No. ZD 2019114). Innovative Funding Project for Graduate Students in Hebei Province (No. CXZZSS2020079).

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 Water Conservancy and Hydroelectric Power, Hebei University of Engineering, Handan, Heilbei Province, China. ² College of Civil Engineering, Hebei University of Engineering, Handan, Heilbei Province, China.

Received: 30 November 2019 Accepted: 4 February 2020

Published online: 16 February 2020

References

- Lin SM (2013) Analysis of service satisfaction in web auction logistics service using a combination of Fruit fly optimization algorithm and general regression neural network. *Neural Comput Appl* 22(3–4):783–791
- Azamirad G, Arezoo B (2016) Structural design of stamping die components using bi-directional evolutionary structural optimization method. *Int J Adv Manuf Technol* 87(1–4):969–979. <https://doi.org/10.1007/s00170-016-8344-7>
- Ide T, Kitajima H, Otomori M et al (2016) Structural optimization methods of nonlinear static analysis with contact and its application to design lightweight gear box of automatic transmission of vehicles. *Struct Multidiscip Optim* 53(6):1383–1394
- Kaveh A, Bakhshpoori T (2016) A new metaheuristic for continuous structural optimization: water evaporation optimization. *Struct Multidiscip Optim* 54(1):2343. <https://doi.org/10.1007/s00158-015-1396-8>
- Rojas-Labanda S, Stolpe M (2015) Benchmarking optimization solvers for structural topology optimization. *Struct Multidiscip Optim* 52(3):527–547. <https://doi.org/10.1007/s00158-015-1250-z>
- Sivapuram R, Dunning PD, Kim HA (2016) Simultaneous material and structural optimization by multiscale topology optimization. *Struct Multidiscip Optim* 54(5):1267–1281. <https://doi.org/10.1007/s00158-016-1519-x>
- Pan WT (2012) A new fruit fly optimization algorithm: taking the financial distress model as an example. *Knowl Based Syst* 26:69–74. <https://doi.org/10.1016/j.knosys.2011.07.001>
- Pan WT (2013) Using modified fruit fly optimization algorithm to perform the function test and case studies. *Connect Sci* 25(2/3):151–160
- Fang B, Chen HM (2019) A novel double strategies evolutionary fruit fly optimization algorithm. *J Shandong Univ* 49(3):22–31
- Pan QK, Sang HY, Duan JH (2014) An improved fruit fly optimization algorithm for continuous function optimization problems. *Knowl Based Syst* 62:69–83. <https://doi.org/10.1016/j.knosys.2014.02.021>
- Cao ZG, Li ZW, Yu JF (2019) Fruit-fly optimization algorithm of sector search mechanism. *Comput Eng Des* 40(6):1584–1590
- Li J, Pan Q, Mao K (2014) Solving the steelmaking casting problem using an effective fruit fly optimization algorithm. *Knowl Based Syst* 72:28–36

13. Wang L, Shi Y, Liu S (2015) An improved fruit fly optimization algorithm and its application to joint replenishment problems. *Expert Syst Appl* 42(9):4310–4323. <https://doi.org/10.1016/j.eswa.2015.01.048>
14. Zhang QT, Fang LQ, Zhao YL (2015) Gemini fly optimization algorithm with Levy flight characteristics. *Comput Appl* 35(5):1348–1352
15. Sheng C, Zou H, Zhu FZ (2019) A new self-changing step fruit fly optimization algorithm. *Microelectron Comput* 36(2):62–67
16. Liu ZX, Wang YF, Zhang Y (2014) Multi-population fruit fly optimization algorithm for automated warehouse selection scheduling. *J Wuhan Univ Technol* 36(3):71–77
17. Han JY, Liu CZ, Wang LG (2013) Dynamic twin group coevolution drosophila optimization algorithm. *Pattern Recognit Artif Intell* 26(11):1057–1067
18. Xin CT, Zou H, Sheng C, Ding GK (2019) The optimal entropy threshold image segmentation of the new fruit fly optimization algorithm. *Microelectron Comput* 36(4):52–56
19. Cai ZX, Gong T (2004) Progress of immune algorithm research. *Control Decis* 19(8):841–846
20. Chen D, Li S, Wang J, Feng Y, Liu Y (2019) A multi-objective trajectory planning method based on the improved immune clonal selection algorithm. *Robot Comput Integr Manuf* 59:431–442. <https://doi.org/10.1016/j.rcim.2019.04.016>
21. Zhen Y, Ding YS, Hao KG, Cai X (2019) An adaptive immune algorithm for service-oriented agricultural internet of things. *Neurocomputing*. <https://doi.org/10.1016/j.neucom.2018.06.094>
22. Corus D, Oliveto PS, Yazdani D (2019) When hypermutations and ageing enable artificial immune systems to outperform evolutionary algorithms. *Theor Comput Sci*. <https://doi.org/10.1016/j.tcs.2019.03.002>
23. Hassan Z, Bahman N, Mohammad M (2019) The economic lot scheduling problem in limited-buffer flexible flow shops: mathematical models and a discrete fruit fly algorithm. *Appl Soft Comput J* 6:904–919. <https://doi.org/10.1016/j.asoc.2019.03.054>
24. Hu JH, Chen P, Yang Y, Liu YM (2019) The fruit fly optimization algorithms for patient-centered care based on interval trapezoidal type-2 fuzzy numbers. *Int J Fuzzy Syst* 5:1270–1287. <https://doi.org/10.1007/s40815-019-00624-y>
25. Wang L, Zhang XL (2017) Research progress of fruit fly optimization algorithm. *Control Theory Appl* 34(5):557–563
26. Xu XP, Shi XT, Qian FC (2008) Solving 0–1 knapsack problem based on monkey swarm algorithm. *Appl Comput Syst* 27(5):133–138
27. Gholizadeh S, Poorhoseini H (2015) Optimum design of steel frame structures by a modified Dolphin echolocation algorithm. *Struct Eng Mech* 55(3):535–554. <https://doi.org/10.12989/sem.2015.55.3.535>
28. Zhang JM, Liu HJ (2018) Prediction of parking space based on BP neural network optimized by genetic algorithm. *J Chongqing Univ* 41(3):76–81
29. Xie KH, Zeng XH, Liu CY (2003) Comparative analysis of immune algorithm and other stochastic optimization algorithms. *J Chongqing Univ* 26(11):43–47
30. Li YC, Yan Z (2019) Improved shuffled frog leaping algorithm on system reliability analysis. *Brain Inf*. <https://doi.org/10.1186/s40708-019-0095-4>
31. Bonyadi MR, Michalewicz Z (2017) Impacts of coefficients on movement patterns in the particle swarm optimization algorithm. *IEEE Trans Evol Comput* 21(3):378–390. <https://doi.org/10.1109/tevc.2016.2605668>
32. Brsini H, Varela F (1991) Hints for adaptive problem solving gleaned from immune network. Springer-Verlag, Berlin Heidelberg, pp 343–354
33. Babalik A, İşcan H, Babaoğlu İ, Gündüz M (2017) An improvement in fruit fly optimization algorithm by using sign parameters. *Soft Comput* 2:1–17. <https://doi.org/10.1007/s00500-017-2733-1>
34. Du TS, Ke XT, Liao JG, Shen YJ (2018) DSLC-FOA: improved fruit fly optimization algorithm for application to structural engineering design optimization problems. *Appl Math Model* 55:314–339. <https://doi.org/10.1016/j.apm.2017.08.013>
35. Mohanty B, Hota PK (2015) Comparative performance analysis of fruit fly optimization algorithm for multi-area multi-source automatic generation control under deregulated environment. *Gener Trans Distrib* 9(14):1845–1855. <https://doi.org/10.1049/iet-gtd.2015.0284>
36. Mousavi SM, Tavana M, Alikar N, Zandieh M (2017) A tuned hybrid intelligent fruit fly optimization algorithm for fuzzy rule generation and classification. *Neur Comput Appl*. <https://doi.org/10.1007/s00521-017-3115-4> (In Press)
37. Zargham S, Ward TA, Ramli R, Badruddin IA (2016) Topology optimization: a review for structural designs under vibration problems. *Struct Multidiscip Optim* 53(6):1157–1177. <https://doi.org/10.1007/s00158-015-1370-5>
38. Cai YG, Chen RH, Qi YH (2019) Chaotic fireworks algorithm for solving salesman problem. *Comput Sci* 46(6A):85–88
39. Assimi H, Jamali A, Nariman-Zadeh N (2018) Multi-objective sizing and topology optimization of truss structures using genetic programming based on a new adaptive mutant operator. *Neural Comput Appl* 23(24):1–21. <https://doi.org/10.1007/s00521-018-3401-9>

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen® journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)
