Article no. 40

THE CIVIL ENGINEERING JOURNAL 4-2019

TRUSS STRUCTURE OPTIMIZATION BASED ON IMPROVED WOLF PACK ALGORITHM

Yan-cang LI [1], Zong-jin YANG [2], Cheng-hua Dang[3]*

1. *Hebei University of Engineering, College of Water Conservancy and Hydroelectric Power, No. 199 Guangmin South Street, Handan, China; liyancang@hebeu.edu.cn*
2. *Hebei University of engineering, College of Civil Engineering, No. 199 Guangmin South Street, Handan, China; yangzongjin6781@sina.com*
3. *[*] Corresponsing author, Hebei University of engineering, No. 199 Guangmin South Street, Handan, China; rebeccalee77@163.com*

ABSTRACT

Aiming at the optimization of truss structure, a wolf pack algorithm based on chaos and improved search strategy was proposed. The mathematical model of truss optimization was constructed, and the classical truss structure was optimized. The results were compared with those of other optimization algorithms. When selecting and updating the initial position of wolves, chaos idea was used to distribute the initial value evenly in the solution space; phase factor was introduced to optimize the formula of wolf detection; information interaction between wolves is increased and the number of runs is reduced. The numerical results show that the improved wolf pack algorithm has the characteristics of fewer parameters, simple programming, easy implementation, fast convergence speed, and can quickly find the optimal solution. It is suitable for the optimization design of the section size of space truss structures.

KEYWORDS

Wolf pack algorithm, Improved wolf pack algorithm, Truss optimization, Chaos thought

INTRODUCTION

Structural optimization design is an important research direction in structural engineering. Engineers can achieve energy saving and material reduction through reasonable optimization[1]. Truss structures are widely used in engineering applications, such as transmission towers, long-span bridges, large space roofs and other structural systems. Therefore, in engineering design, it is an important research direction how to optimize the design of the structure to minimize the structural consumables on the premise of satisfying the safety, economy and reliability of the structure. In recent years, bionic intelligent algorithms have been emerging in the field of engineering structure design, such as genetic algorithm, ant colony algorithm, particle swarm optimization, cuckoo algorithm and so on[2-5]. These algorithms have effectively promoted the development of structural optimization in the research of truss structure optimization. However, there is still room for research on the convergence speed and stability of the algorithm. Wolf pack algorithm was first proposed by Yang in 2007 as a swarm intelligence optimization algorithm to simulate the characteristics of wolf hunting. After that, many scholars had studied and improved the wolf swarm algorithm[6-9]. The algorithm has good global convergence and is especially suitable for solving high-dimensional and multi-peak complex functions. Because of its strong robustness, wolf pack algorithm has been applied in many fields, such as multi-area power generation control, job shop scheduling, disease treatment, TSP problem, etc. [10-15]. But the algorithm also has some shortcomings such as easy to fall into local optimum.

On the basis of previous studies, wolf pack algorithm is improved by introducing chaos strategy and reverse learning strategy into initialization and wolf swarm updating, so that prey can be distributed as evenly as possible in solution space. And the search flexibility of wolf-hunting is enhanced by changing the formula of wolf-hunting. The information exchange between the wolves is increased through the method that the wolves understand the overall information in the course of the attack, so as to avoid falling into local extremes. The improved algorithm was applied to the truss simulation experiment and compared with other algorithms, which verifies the effectiveness of the improved wolf pack algorithm.

IMPROVED WOLF PACK ALGORITHM

It is found that wolves are tightly organized in hunting, with a clear division of labour and concerted efforts to catch prey. Wolves first sent a small number of strong wolves to search for odour within a certain range, and the strong-smelling wolves called their companions to attack and surround their prey by howling. Finally, they distributed food through the principle of survival of the fittest. Wolf swarm algorithm is proposed based on the predatory behaviour of wolves mentioned above. It is intended to simulate the optimization of processing function of wolves' predatory behaviour. However, there are also some shortcomings in the basic wolf pack algorithm, such as precocity, incomplete migration mode of detective wolves, and lack of information exchange in the running process, which may lead to the embarrassing situation of local optimization.

Initialization and update selection of wolves

In the basic wolf pack algorithm, the generation criterion is to find the wolf with the best fitness value as the wolf in the search space. The basic wolf pack algorithm does not initialize the location of the wolf swarm, which will lead to the uneven individual wolf swarm. We combines reverse learning and chaos optimization idea to design reverse chaos update strategy[16].

(1) Two one-dimensional chaotic mapping equations, Sinusoidal iterator and Gauss map, are used to generate D chaotic sequences of length H with equal probability. $AX=(ax_1,ax_2,\cdots,ax_D)$, $ax_d=(ax_{1d},ax_{2d},\ldots,ax_{Hd})^T$, $d=1,2,\ldots,D$. H is defined as the number of artificial wolves that need to be updated or initialized. The following expressions are one-dimensional mapping expressions of Sinusoidal iterator and Gauss map, respectively.

$$x_{k+1} = b \cdot x_k \sin(\pi x_k)\,, x_k \in (0,1) \tag{1}$$

$$x_{k+1} = \begin{cases} 0, x_k = 0 \\ 1/x_k - [1 - x_k], x_k \in (0,1) \end{cases} \tag{2}$$

(2) According to the following formula, chaotic sequences are mapped into solution space to initialize or update wolves: $X=\{x_1,x_2,\ldots,x_H\}^T$

$$x_{id} = \lambda a x_{id}(1 - x_{id}) \quad \lambda = 3 \tag{3}$$

(3) According to the following formula, the reverse wolf pack of wolves is obtained: $OX=\{ox_1,ox_2,\ldots,ox_H\}^T$

$$ox_{id} = s \cdot (min_d + max_d) - x_{id} \tag{4}$$

Where: min_d is the lower limit of the d-dimensional solution space; max_d is the upper limit of the d-dimensional solution space; s is a random number between 0 and 1.

(4) The objective function values of *X* and *OX* were calculated, and *H* artificial wolf with the largest objective function value was selected as the initialization or update artificial wolf.

Improvement of Walking Behaviour

Choose the best *m* wolves besides the head wolf as the head wolf, search in the predefined direction, and retain the better prey. Once the better prey than the current head wolf is found, the head wolf with the prey becomes the head wolf. This process is called the head wolf wandering behaviour. The purpose of roaming behaviour is to explore the solution space comprehensively and search for new candidate solutions. However, the basic roaming behaviour lacks guidance. Once the number of search directions is determined, the roaming direction will be determined, and the solution space cannot be searched comprehensively, which easily leads to the algorithm falling into local optimization[14]. For this reason, the new formula is obtained by introducing the phase factor as follows:

$$x_{id}^{p} = x_{id} + (1/z_{max}) \cdot \sin(2\pi \cdot p/h + \theta) \cdot step_{a}^{d} \qquad (5)$$

Where z is the number of iterations, z=1,2,…,z_{max}; $\theta \in (0, p/h)$; *h*=6. The design of $(1/z_{max})$ embodies the asymptotic idea of "from coarse to fine" of wolf searching, and the introduction of phase factor improves the flexibility, randomness and ergodicity of wolf searching. In addition, the introduction of wolf detection update rules and the design of "from coarse to fine" wandering mechanism. The basic idea is as follows: if the function value of *H* direction is less than the objective function value of wolf *i*, there will probably be a maximum value around wolf *i*. At this time, shorten the search distance (walking step) and search the surrounding *h* direction again. If the function value of *H* direction is still less than the objective function value of wolf *i*, the search distance will continue to be shortened. After several times of shortening the search distance, the function values of *H* directions around the wolf are still smaller than the objective function values of the wolf *i*. It can be considered that the wolf *i* falls into the local optimal position. At this time, the wolf *i* needs to restart its initial position and then perform the walking behaviour. The design idea further embodies the asymptotic idea of wolf swarm algorithm from coarse to fine.

Improvement of Running Behaviour

The wolf howls to inform the surrounding fierce wolves to quickly approach the wolf and search for high-quality prey. If the wolf finds better prey than the wolf, the wolf howls again instead of the wolf until the wolf stops at a certain distance from the prey. This process is called rush behaviour. In the *k*+1 iteration of the wolf *i*, the position in the d-dimensional variable space is:

$$x_{id}^{k+1} = x_{id}^{k} + step_{b}^{d} \cdot (g_{d}^{k} - x_{id}^{k}) \div |g_{d}^{k} - x_{id}^{k}| \qquad (6)$$

where, g_{d}^{k} is the position of the K generation group of wolves in the d-dimensional space. In the course of the attack, the lack of necessary information exchange between the wolves and the lack of timely understanding of the "companion" information limit the search ability in the attack. Interaction among individuals in intelligent algorithms helps to enhance the optimization ability and the ability to jump out of local extremum. In order to increase the interaction between wolves, this paper uses the following search methods, as shown in Formula (7).

$$x_{id} = x_{id} + \upsilon_{id}(x_{best,d} - x_{id}) + \tau_{id}(x_{jd} - x_{kd}) \qquad (7)$$

Where, υ_{id} is the random number in [0,1]; τ_{id} is the random number in [-1,1]; *k≠i≠j*; $x_{best,d}$ is the dimension coordinate of the current optimal solution. The first half of Formula (7) enhances the local search ability of the algorithm, and the second half of it enhances the global search ability of the algorithm. It balances the exploiting ability and exploring ability of the roaming behavior well. It not only embodies the leadership ability of the wolf, but also maintains the exchange of information between wolves and improves the diversity of the population.

In swarm algorithm, no matter what stage of search, the communication between groups is an important link. Therefore, after each round (6) is executed, the one-time (7) interactive search process is carried out to select the most odorous prey which is larger than the current position odour concentration Y_i, and to move forward a step, update the position X of the wolf. Wolves with the highest odour concentration in wolves are currently selected as the first wolf.

The specific rules of interactive attack are as follows:

(1) The head wolf initiates the summoning behaviour and calls the surrounding fierce wolves to approach the head wolf position quickly. According to formula (6), the wolf carries out a rush once to get the position of the new prey after the rush.

(2) Random selection of fierce wolves *k*, *j*, according to formula (7) for an interaction, get new prey after interaction, so far, fierce wolf *i* rushed to search for a total of two prey;

(3) By comparing the odour concentration of the two new prey and the original location of the fierce wolf, fierce wolf *i* made a step towards the direction with the strongest smell of the three.

(4) A comparison was made between the fierce wolf with the maximum concentration of prey odour and the lead of the wolf. If $Y_i > Y_{lead}$, fierce wolf *i* replaced the lead wolf as the new lead wolf and initiated the summoning behaviour again. Otherwise, the head wolf remained in the same position and entered the wolf siege behaviour.

By increasing fierce wolf information interaction between, better let the fierce wolf on the impact on the way to understand the global information, avoid falling into local extremum, reduce the number of attacks, simplify the complexity of the algorithm. Through the analysis, the basic behaviour fired fine search, but easy to fall into local optimization, algorithm premature convergence, and constantly tumbling makes the algorithm complexity, robustness is not stable, and the running time is too long, unfavourable to better solve the problem of real-time. It appear that that primary attacking behaviour is not finis, but after the interaction has been added, the algorithm has global direction, the local optimal is better, and the global convergence speed is accelerate.

The standard wolf pack algorithm is an algorithm proposed to solve the continuity problem. Considering that the truss structure is optimized as a discrete problem, a binary wolf pack algorithm is introduced[17]. The position X_i of artificial wolf *i* is represented as binary encoding (x_{i1}, x_{i2}... , x_{ij},... , x_{im})(i = 1,2... , N. j = 1,2... , m); N is the total number of artificial wolves; M is the encoding length. x_{ij} is the value of the number j for position X_i and can only be 0 or 1. The distance between p and q of two artificial wolves is defined as the binary encoded Manhattan distance between them, as shown below:

$$L(p,q) = \sum_{j=1}^{m} \left| x_{pj} - x_{qj} \right|, p,q \in \{1,2,\cdots,N\} \tag{8}$$

$\Gamma(X_i, M, r)$is defined as the motion operator, which represents the moving position of the artificial wolf. M is the set of inverted coding bits and is not empty, which can be understood as the range of activity of the artificial wolf. r represents the number of coding bits to invert, which can be understood as the walking step of the artificial wolf. For example: X_i ={0,0,1,0,0 1}, M = 3, r = 1, so, $\Gamma(X_i, M, r) = \{1,0,1,1,0,0\}$. The modified Wolf pack algorithm formula is replaced by the corresponding binary formula, which can be used in truss structure optimization.

Improve the basic flow chart and flow chart of the Wolf pack algorithm

The basic flow of the improved Wolf pack algorithm is as follows (taking solving the maximum value as an example) :

*Step*1: initialize. Initialise that number N of artificial wolf, the maximum number of iterations k_{max}, the maximum number of walk T_{max}, the distance determination factor ω, the step factor S, updating the scale factor β, and initializing the spatial position X of the wolf by the inverse double chaos updating strategy;

*Step*2: walk behaviour. The artificial wolf with the largest value of the objective function was selected as the head wolf, and the other artificial wolves were regarded as the detective wolves and walked away according to equation (5) (During the exploration of wolf i wandering, if the function values of h directions around exploration of wolf i are all smaller than the target function values of exploration of wolf i, $step_a$, the wandering step length, is halved. If the exploration of wolf i is still unable to move forward after repeated halving of the wandering step length, it is considered that exploration of wolf i is trapped in local optimum, and the spatial location of exploration of wolf should be updated with the reverse double-chaos updating strategy), until the target function value Y_i of detective wolf i, which is greater than the target function value Y_{lead} or the wandering number of detective wolf, reaches the maximum wandering number T_{max}, and is transferred to *step*3

*Step*3: summon behaviour. The first wolf calls upon the surrounding fierce wolves to quickly approach the position of the first wolf. According to formula (6), the fierce wolf runs once to obtain the position of the new prey after the attack. The fierce wolves k,j were randomly selected, and an interaction was conducted according to equation (7) to obtain the new prey V_i after interaction. Thus, fierce wolf i ran and found two prey in total. Comparing the odour concentration of the two new prey and the original location of the fierce wolf, fierce wolf i made a step towards the direction with the strongest smell of the three. Compare the fierce wolf i, which has the maximum concentration of prey odour, with the concentration of head wolf. If $Y_i > Y_{lead}$, fierce wolf i becomes the new head wolf instead of head wolf and starts the summoning behaviour again.

*Step*4: act of siege. The head of the wolf position as the moving position of the prey, the fierce wolves involved in the siege by the formula of the prey siege;

$$x_{id}^{k+1} = x_{id}^{k} + \lambda \cdot step_b^d \cdot \left| g_d^k - x_{id}^k \right| \qquad (9)$$

Where, λ is the random number with uniform distribution between [-1,1]; $Step_c$ is the attack step size of wolf i during siege.

*Step*5: wolf pack update. The objective function of the optimal wolf generated in this iteration is compared with the objective function value of the head wolf in the last iteration. If it is larger, the position of the head Wolf is updated. Otherwise, the first wolf is recorded with no update times t. Determine the number of artificial wolves with smaller values of the elimination objective function R, $R \in (N/2 \cdot \beta, N/\beta)$ based on the updated proportional factor. If t is less than the limit value t_{max}, then the wolf pack is updated according to equation (10) and combined with reverse learning. On the contrary, the wolves will be updated according to the reverse double-chaos strategy.

$$x_{id} = g_d \cdot [2 - \cos(\psi)] \quad \psi \in (-0.1, 0.1) \qquad (10)$$

*Step*6: judge termination. Judge whether the target function value of head wolf meets the calculation accuracy requirement, or whether the algorithm reaches the maximum iteration number k_{max}. If so, output the location of head wolf and the target function value, otherwise turn to *Step*2.

The flow chart of the improved wolf pack algorithm is shown in Figure 1.

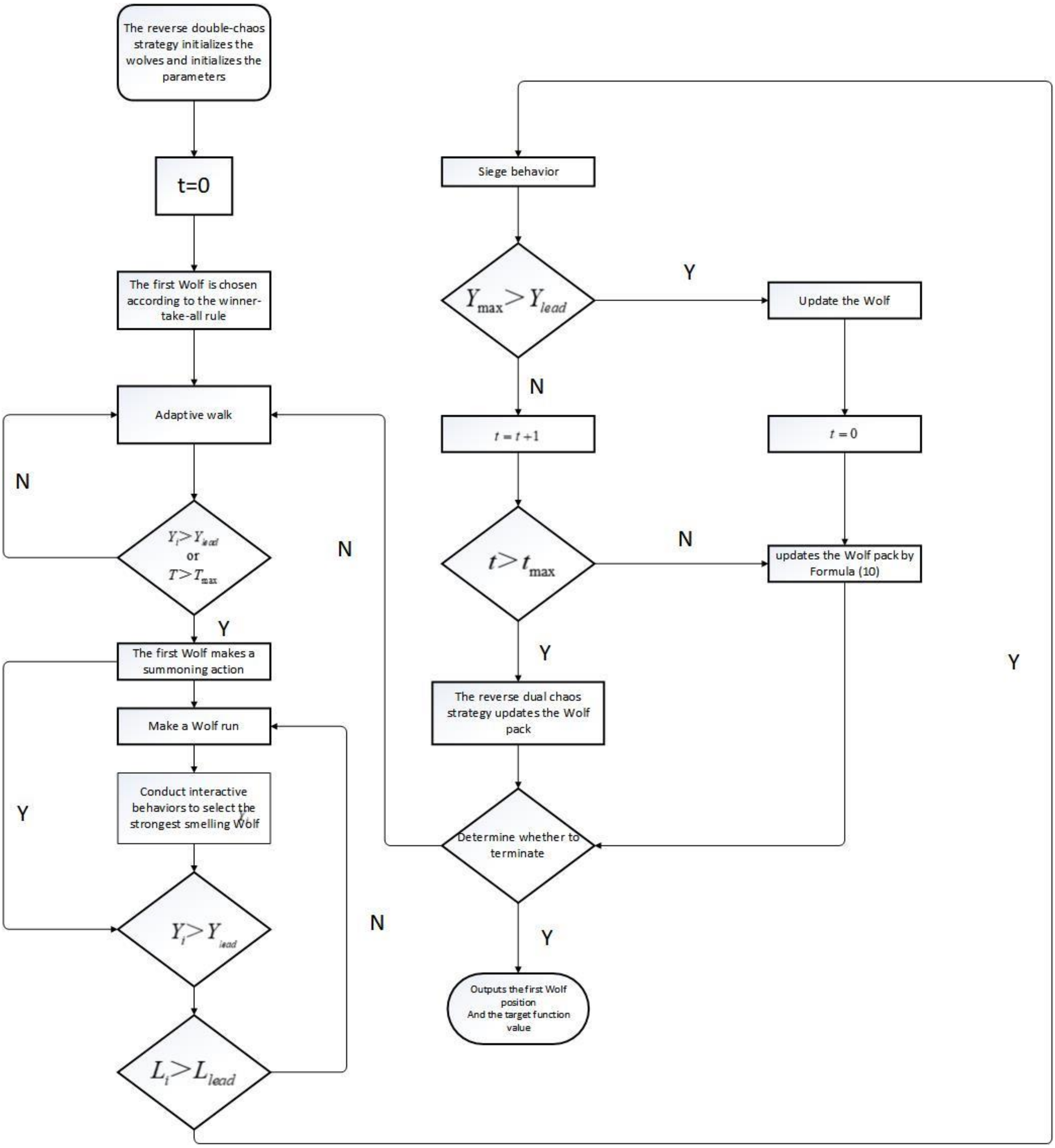

Fig.1 – Flow chart of improved WPA

EXAMPLE ANALYSIS

In industrial production, the cross-sectional area of truss structure members is generally standardized, that is, the cross-sectional area is selected from a given set of discrete real Numbers. Due to different conditions and different requirements, truss structure optimization can have multiple optimization objectives, such as minimum total mass of the structure, minimum displacement of designated nodes, maximum natural frequency, etc. [18]. In this paper, the optimal cross sectional area of the truss and the minimum mass of the truss and the minimum joint displacement are found under the stress constraint condition of the truss. Taking the n-bar truss structure system as the research object, the basic parameters of the system are known (Including elastic modulus, material density, maximum allowable stress, maximum allowable displacement, etc) . Under the given load conditions, the optimal section area of the n - bar truss is found to minimize the mass.

10-bar plane truss

Figure 2 shows the 10-bar spatial truss structure model with the known material density *ρ* =2768 *kg*/*m*³, the elastic modulus *E*=68950 *MPa*, the stress constraint is [-172.4, 172.4]*MPa*, *L*=9144 mm, the downward load *p*=444.5 *KN* concentrated force at nos. 2 and 4, and the downward displacement constraint of movable nodes is 50.8 *mm*. The truss has 6 joints and 10 design variables and is made of aluminium. The control parameters of the algorithm are set as: the maximum number of iterations is 500; Search space dimension 10. The node loads are shown in Table 1.

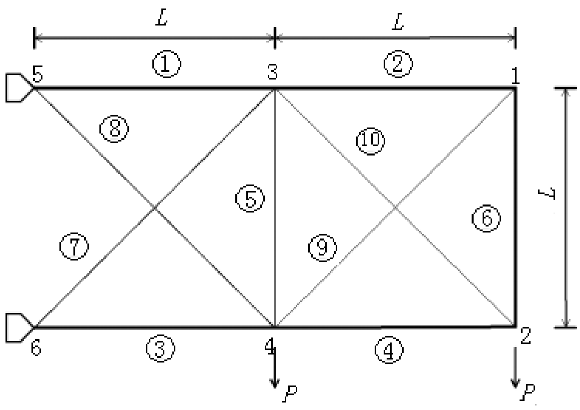

Fig.2 – Schematic diagram of the 10-bar space truss structure

Tab.1 - Design area of cross section of ten bar trusses

number	1	2	3	4	5
area	0.645	3.225	6.450	12.90	19.35
number	6	7	8	9	10
area	25.80	32.25	38.70	41.93	45.15
number	11	12	13	14	15
area	48.38	51.60	54.83	58.05	61.28
number	16	17	18	19	20
area	64.50	70.95	77.40	83.85	90.30
number	21	22	23	24	25
area	96.75	103.2	109.7	116.1	122.6
number	26	27	28	29	30
area	129.0	135.5	141.9	148.4	154.8
number	31	32	33	34	35
area	161.3	167.7	173.1	183.2	190.5
number	36	37	38	39	40
area	195.8	202.0	207.6	211.2	216.8

The optimized results are shown in Table 2.

Tab. 2 - The results of truss optimization of ten pole

Serial number	Section area of member（cm^2）		
	IACO [19]	IPSO[20]	IWPA
1	210.76	196.92	198.25
2	0.64	0.64	0.66
3	147.97	149.79	146.85
4	98.47	97.91	97.03
5	0.64	0.64	0.62
6	3.39	3.55	3.01
7	128.91	135.58	135.69
8	49.70	48.07	48.37
9	0.65	0.64	0.63
10	138.41	139.18	136.94
The total weight of the structure（*kg*）	2299.69	2295.50	2283.68

From the above Table 2, we can conclude that, in the same conditions of constraints, the algorithm with the improved wolves of the 10 bar truss structure has been optimized design, the optimized the structure of the total mass of 2283.68 *kg*, compared with the improved particle swarm algorithm quality reduced 0.70%, the quality reduced 0.52% compared with the improved genetic algorithm, the optimization results better improved. The optimal iteration curves of the three algorithms are shown in Figure 3.

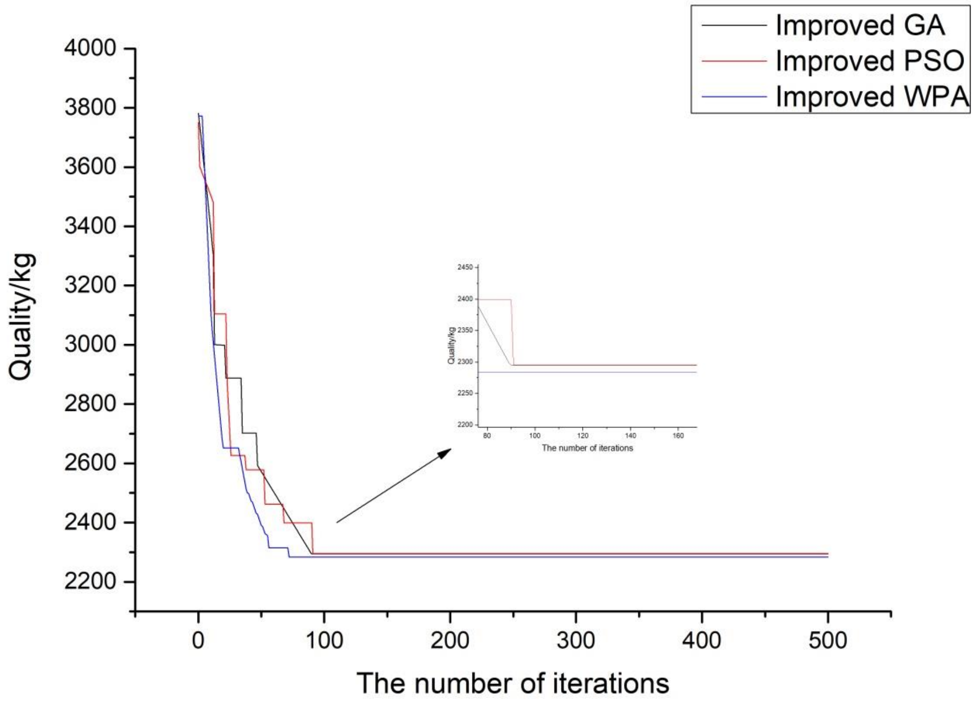

Fig.3 – Three algorithms optimization iteration curve

As can be seen from Figure 3, the improved wolf pack algorithm can search for the global optimal solution, which has higher convergence accuracy and speed than the improved genetic algorithm and the improved particle swarm optimization algorithm, with obvious effect. This is due to the use of chaos during the initialization so that the wolves as much as possible to traverse the solution space in all states, to avoid early into early maturity. By introducing the phase factor, the hunting wolf walk formula is optimized and the hunting flexibility is improved. The interactive behaviour is added to make the algorithm have global guidance, better jump out of the local optimal, accelerate the global convergence speed, and improve the global optimization ability.

72 bar space truss

Figure 4 shows the 72 bar space truss structure, which takes into account two load conditions, as shown in Table 3. The 72 bars in the structure are divided into 16 groups according to the stress of the bars, and the specific groups of the bars are shown in Table 4. Bar all use the same materials, the material density of $\rho = 2678\,kg/m^3$, the elastic modulus $E = 68950MPa$, bar in every direction of each connection point of the maximal displacement change interval for $\pm 6.35mm$, limit allowable stress range is [-172.375,172.375], the optimized results as shown in Table 5.

Tab.3 - Load cases of the 72-bar spatial truss structure

node	Condition 1			condition 2		
	F_x	F_y	F_z	F_x	F_y	F_z
1	22250	22250	-22250	0	0	-22250
2				0	0	-22250
3				0	0	-22250
4				0	0	-22250

Note: the data unit in the table is (kN).

Tab.4 - The grouping of the 72-bar spatial truss structure

Group number	Bar code	Group number	Bar code
A_1	1,2,3,4	A_9	37,38,39,40
A_2	5,6,7,8,9,10,11,12	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

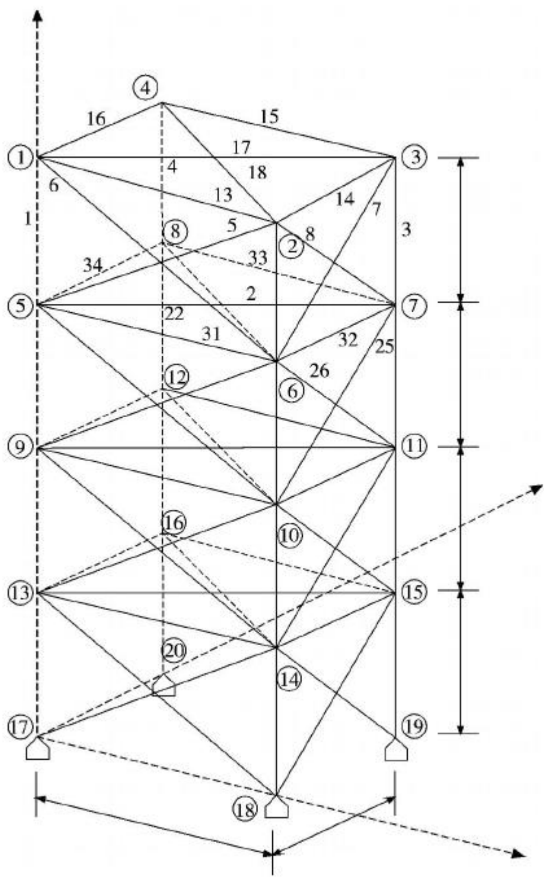

Fig.4 – structure diagram of the 72-bar spatial truss

Tab.5 - Comparison of optimal designs for the 72-bar spatial truss structure

Bar grouping number	IPSO [20]	IGA[21]	IWPA
1	100.74	102.257	101.334
2	360.28	372.965	345.552
3	269.20	220.268	264.254
4	367.69	391.997	367.293
5	342.92	170.513	326.915
6	336.70	353.458	335.484
7	64.516	64.516	64.514
8	64.516	64.516	64.517
9	870.23	713.977	825.875
10	318.26	373.811	332.126
11	64.516	64.516	64.536
12	64.516	64.516	64.514
13	1188.0	1330.019	1224.055
14	325.44	324.878	332.776
15	64.516	64.516	64.518
16	64.516	64.516	64.513
The total weight of the structure（*kg*）	172.44	173.26	172.20

Note: the weight unit of the data in the table is (kg), and the cross-sectional area unit of the bar is (mm^2).

According to Table 5 of the optimization results, the total weight of the 72 bar truss structure was changed to 172.36*kg* after the improvement of the Wolf pack algorithm. IWPA algorithm further reduced the self-weight of the truss structure by 0.14% compared with IPSO algorithm. Compared with IGA, the mass is reduced by 0.62%, so the improved drosophila optimization algorithm has certain advantages in the final optimization results.

DOI 10.14311/CEJ.2019.04.0040

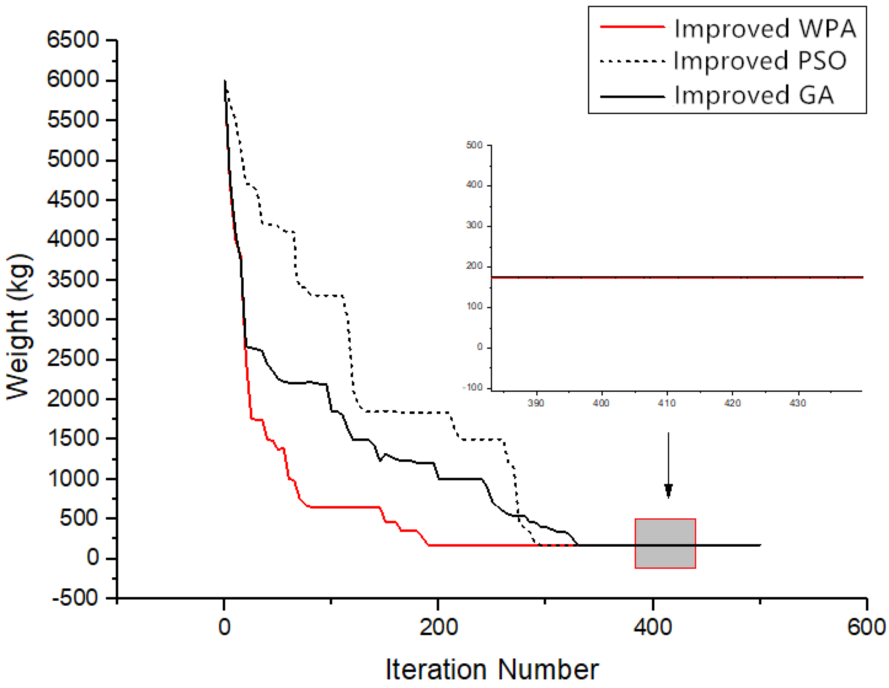

Fig.5 - Iterative curve diagram of improved WPA

It can be seen from the iterative graph 5 that the improved WPA we proposed is superior to the comparison algorithm in both the convergence speed and the final global optimal solution. The global optimal solution can be found after about 200 iterations, and the optimization efficiency is also superior to the comparison algorithm, and the optimization process is relatively stable.

CONLUSION

(1) Structural optimization design is very important for the truss structural engineering. In recent years, with the concerted efforts of many scholars, the artificial intelligent algorithms have been successfully applied in the field of engineering structure design. But, for the efficiency of the algorithms, we still have a long way to go.

(2) Aiming at providing a new method for the optimization of truss structure, an improved wolf pack algorithm based on chaos and improved search strategy was proposed. The improved wolf pack algorithm is applied to the optimization design of truss structure. The example verifies that the improved wolf pack algorithm has good stability, high optimization efficiency and fast convergence speed. The example of truss optimization shows that the improved algorithm can be successfully applied to the section optimization of truss structure.

(3) Only the initialization of the wolf algorithm, the exploring wolf walk formula and the running way are improved, and the values of the wolf algorithm's wolf pack size, the siege algorithm threshold and other parameters need to be further studied. Future work will apply the improved algorithm to other fields, such as radar imaging.

(4) The results on the example cases show that reducing of weight is not so large (less than 1 %). Companies are looking for ways to save costs of course, but design of the truss structures

from many different cross-sections are not the best solution. This should be taken into account in the future work.

ACKNOWLEDGEMENTS

This work was supported by the Project of Scientific Research Program of Colleges and Universities in Hebei Province (No. ZD 2019114) and the Fund of Hebei Engineering Technology Research Centre for Water Resources High Efficiency Utilization.

REFERENCES

[1] Mathias S. Jørgensen, Uffe F. Larsen, Karsten W. Jacobsen, et al., 2018. Exploration Versus Exploitation in Global Atomistic Structure Optimization. Journal of Physical Chemistry A, vol. 122, no. 5: 1504-1509.

[2] Pirmohammad S, Marzdashti S E., 2018. Crashworthiness Optimization of Combined Straight-tapered Tubes Using Genetic Algorithm and Neural Networks. Thin-Walled Structures, vol. 127: 318-332.

[3] Imran M M, Mazhar F, Ahmad R., 2016. Multivariate Optimization of Fiber Reinforced Laminate Using Ant Colony Optimization Algorithm. Materials Science Forum, vol. 867: 116-120.

[4] El-Shorbagy, M.A, Hassanien A E., 2018. Particle Swarm Optimization from Theory to Applications. International Journal of Rough Sets and Data Analysis, vol. 5, no. 2: 1-24.

[5] Nguyen T T , Vo D N ., 2018. The Application of an Effective Cuckoo Search Algorithm for Optimal Scheduling of Hydrothermal System Considering Transmission Constraints. Neural Computing & Applications, vol. 9: 1-22.

[6] Yang Chenguang, Tu Xuyan, Chen Jie., 2007. Algorithm of Marriage in Honey Bees Optimization Based on the Wolf Pack Search. International Conference of the International Conference on Intelligent Pervasive Computing, 462-467.

[7] Rui Tang, Simon Fong, Xin She Yang, et al., 2012. Wolf Search Algorithm With Ephemeral Memory. Seventh International Conference on Digital Information Management, 165-172.

[8] Simon Fong, Suash Deb, XinShe Yang, et al., 2015 A Heuristic Optimization Method Inspired by Wolf Preying Behavior. Neural Computing & Applications, vol. 26, no. 7: 1725-1738.

[9] Wu Husheng, Zhang Fengming, Wu Lushan., 2013. New Swarm Intelligence Algorithm-wolf Pack Algorithm. Systems Engineering and Electronics., vol. 35, no. 11: 2430-2438.

[10] Komaki G M, Kayvanfar V., 2015. Grey Wolf Optimizer Algorithm for the Two-stage Assembly Flow Shop Scheduling Problem with Release Time. Journal of Computational Science, vol. 8: 109-120.

[11] Mirjalili S, Saremi S, Mirjalili S M, et al., 2016. Multi-objective Grey Wolf Optimizer: A novel Algorithm for Multi-criterion Optimization. Expert Systems with Applications, vol. 47: 106-119.

[12] Sharma Y, Saikia L C., 2015. Automatic Generation Control of a Multi-area ST – Thermal Power System Using Grey Wolf Optimizer Algorithm Based Classical Controllers. International Journal of Electrical Power and Energy Systems, vol. 73: 853-862.

[13] Radu Emil Precup, Radu Codrut David, Emil M Petriu., 2016. Grey Wolf Optimizer Algorithm-Based Tuning of Fuzzy Control Systems With Reduced Parametric Sensitivity. IEEE Transactions on Industrial Electronics, vol. 64, no.1: 527-534.

[14] Shubham G, Kusum D., 2018. Cauchy Grey Wolf Optimiser for Continuous Optimisation Problems. Journal of Experimental and Theoretical Artificial Intelligence, vol. 30, no. 6: 1051-1075.

[15] Shankar Chakraborty, Ankan Mitra., 2018 Parametric Optimization of Abrasive Water-jet Machining Processes Using Grey Wolf Optimizer. Materials and Manufacturing Processes, vol. 33, no.13: 1-12.

[16] Goulianas K, Margaris A, Refanidis I, et al., 2018. Solving Polynomial Systems Using a Fast Adaptive Back Propagation-type Neural Network Algorithm. European Journal of Applied Mathematics, vol.29, no. 2: 301-337.

[17] Wu Husheng, Zhang Fengming, Li Hao, et al., 2015. Discrete Wolf Pack Algorithm for Traveling Salesman Problem. Control and Decision, 30(10): 136-142.

[18] Lu Dai, Fuling Guan, James K.Guest., 2014.Structural Optimization and Model Fabrication of a Double-Ring Deployable Antenna Truss. Acta Astronautica, vol. 94, no. 2: 843-851.

[19] Kaveh A, Azar B F, Hadidi A, et al., 2010.Performance-based Seismic Design of Steel Frames Using Ant Colony Optimization. Steel Construction, vol. 66, no. 4: 566-574.

[20] Teke T, Pekcan O., 2013. A Bat-inspired Algorithm for Structural Optimization. Computers & Structures, vol. 128, no. 128:77-90.

[21] 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 Computing & Applications, vol. 23–24: 1-21.