Available online at www.CivileJournal.org

Civil Engineering Journal

Vol. 6, No. 8, August, 2020

Improved Wolf Pack Algorithm for Optimum Design of Truss Structures

Li Yan-Cang [a], Xu Pei-Dong [b*]

[a] *Professor, College of Civil Engineering, Hebei University of engineering, Handan 056038, China.*

[b] *Graduate student of College of Civil Engineering, Hebei University of Engineering, Handan 056038, China.*

Received 08 May 2020; Accepted 18 July 2020

Abstract

In order to find a more effective method in structural optimization, an improved wolf pack optimization algorithm was proposed. In the traditional wolf pack algorithm, the problem of falling into local optimum and low precision often occurs. Therefore, the adaptive step size search and Levy's flight strategy theory were employed to overcome the premature flaw of the basic wolf pack algorithm. Firstly, the reasonable change of the adaptive step size improved the fineness of the search and effectively accelerated the convergence speed. Secondly, the search strategy of Levy's flight was adopted to expand the search scope and improved the global search ability of the algorithm. At last, to verify the performance of improved wolf pack algorithm, it was tested through simulation experiments and actual cases, and compared with other algorithms. Experiments show that the improved wolf pack algorithm has better global optimization ability. This study provides a more effective solution to structural optimization problems.

Keywords: Wolf Pack Algorithm; Improvement; Adaptive; Levy Flight; Structural Optimization.

1. Introduction

An important derivative of the long-term development of structural design theory is structural design optimization [1]. Design optimization is to make the initial structure into a structure that is more satisfactory to the researcher under certain constraints set by the researcher, according to a certain goal the researcher wants, such as the goal of minimizing the quality of the structure [2].

Social group living is a common natural phenomenon in which social groups are able to adapt to the principles of natural selection and survive intra-species competition. They increase collective interests by reducing their energy consumption [3-5]. Swarm intelligence is an algorithm that simulates the evolution or foraging behavior of natural organisms [6]. In recent decades, inspired by the phenomenon of animal groups, many optimized computational methods have been developed to solve complex problems. Therefore, the use of swarm intelligence algorithm to optimize the design of the structure is a novel and efficient method. Classic swarm intelligence algorithms are genetic algorithm [7], particle swarm optimization algorithm [8] and ant colony optimization algorithm [9]. New swarm intelligence algorithms are artificial fish swarm algorithm [10], artificial bee colony algorithm [11], butterfly algorithm [12], fruit fly algorithm [13], pollen algorithm [14] and chicken swarm algorithm [15]. Birds, fish, ants, bees and others exhibit powerful swarm intelligence through constant adaptation and cooperation, which give us solutions

* Corresponding author: xupeidong555@163.com

http://dx.doi.org/10.28991/cej-2020-03091557

to many complex new ideas and problems. These intelligent optimization algorithms have solved many complex and difficult problems, greatly increasing people's ability to deal with optimization problems, and they have effectively promoted the development of computational intelligence. However, in terms of calculation accuracy, further research is needed.

The Wolf Pack Algorithm (WPA, Wolf Pack Algorithm) was a new swarm intelligence algorithm proposed by Wu Shenghu (2013) [16]. The algorithm has better performance in optimization when solving optimization problems, but it also has some shortcomings, such as slow convergence speed, low convergence accuracy and low robustness, etc. [17]. Wu et al. (2018) aimed at the problem of slow convergence in the later period, introduced interactive walking motion, and proposed a wolf swarm search algorithm with leadership strategy [18]. Wu and Fengming (2014) proposed an uncultivated wolf pack algorithm (UWPA Uncultivated Wolf Pack Algorithm) to solve the problem of high-dimensional function optimization [19]. Teng et al. (2018) used tent chaotic sequences to start individual positions, and proposed a wolf swarm optimization algorithm combining particle swarm [20]. Chen et al. (2018) introduced a differential evolution strategy and proposed an improved wolf pack algorithm (IWPA, Improved Wolf Pack Algorithm) based on differential evolution [21]. Zhang et al. (2017) introduced the idea of controlling adaptive parameters and chaos, and proposed an adaptively adjusted chaotic gray wolf optimization algorithm [22]. Kaveh and Zakian (2017) proposed an improved gray wolf algorithm by adding a few tunable parameters to provide proper adaptability for the algorithm and to optimize the structures using fewer structural analyses, while obtaining finer solution. These improved algorithms had improved the accuracy and convergence accuracy of the algorithm to some extent, but they were still some shortcomings [23].

Based on this, in order to overcome the shortcomings of the wolf pack algorithm, based on the wolf pack algorithm, this paper proposes a search strategy based on Levi's flight strategy for the behavior of detecting the wolf, and puts forward an adaptive step size for the movement during the summoning and siege behavior. The improved method is such that the step size of each wolf movement is determined by the current position of the wolf and the current head wolf position. After testing, the proposed adaptive step size and Levy's flight strategy improved wolf pack algorithm (LWPA, Levy Flight and Adaptive Step Size Strategy Improved Wolf Pack Algorithm) greatly accelerates the convergence speed and improve the convergence accuracy, Finally, the improved wolf pack algorithm is used to optimize the design of the truss structure and compare with other algorithms. Experiments show that LWPA has better optimization performance

2. Levy Flight Strategy and Adaptive Step Size Wolf Pack Algorithm

2.1. Description of Intelligent Behavior and Rules

2.1.1. Initialization of Wolves

Let the size of the wolf pack be N, the dimension of the search space be D, and the position of the i artificial wolf can be expressed as Equations 1 and 2:

$$X_i = (x^1{}_i, \cdots x_i{}^D) \tag{1}$$

$$x_i^d = x_{\min} + rand \times (x_{\max} - x_{\min}) \tag{2}$$

Where x_{max} and x_{min} are the maximum and minimum ranges of the search space respectively. $rand \in (0,1)$ a random number.

2.1.2. Wolf Generation Rules

In the initial solution space, the artificial wolf with the best objective function value is selected as the head wolf, and the position of the artificial wolf is updated after each iteration. If there are multiple optimal artificial wolves at this time, one was randomly selected to become the head wolf. The head wolf does not perform the following intelligent behavior, and directly enters the iteration until it is replaced by another stronger artificial wolf.

2.1.3. Walking Behavior Based on Levy Flight

The optimal S_sum artificial wolves, excluding the head wolf, are selected as detective wolves. S_sum random selects the integer between $[n/\alpha + 1, n/\alpha]$, and α is the detective wolf scale factor. However, in practice, it was found that in the process of wandering, the detective wolf would only blindly follow the head wolf and approach the prey breath concentration at the head wolf position, without caring about whether there was a better prey breath concentration around him. In the later stage of the algorithm, it would lead to loss of diversity in the population, easy to fall into local convergence and premature convergence. Aiming at this defect, this paper uses levy flight to conduct global search for detective wolves in the group. Levy flight belongs to random walk, which is a good search strategy and can expand the search scope [24]. Through levy flight, the new generation of detective wolf is calculated as follows Equation 3:

$$x_{id}(t+1) = x_{id}(t) - \oplus Levy(\delta) \tag{3}$$

Where, $x_{id}(t)$ represents the position of wolf i in d dimension of t iteration, $\oplus$ is the point-to-point multiplication, c is the random number of wolf i position, determined by Equation 4, $Levy(\delta)$ represents the random search path, determined by Equation 5.

$$c = rand(size(i\_position)) \tag{4}$$

$$Levy(\delta) \sim 0.01 \frac{u}{|v|^{\frac{1}{\delta}}} (X_i - X_{ibest}) \tag{5}$$

In this equation, the value range of δ is $1 < \delta < 3$, in this paper δ is 1.5, X_{best} represents the historical optimal wolf location, and the normal distribution of u and v is shown in Equation 6:

$$u \sim N(0, \sigma_u^2), v \sim N(0, \sigma_v^2) \tag{6}$$

σ_u and σ_v is shown in Equation 7:

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta) \sin\left(\frac{pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \beta 2^{(\beta-1)/2}} \right\}^{1/\beta}, \sigma_v = 1 \tag{7}$$

At this time, the function value of prey breath concentration perceived by the detective wolf is Y_{ip}. The direction of the function value of prey breath concentration, which is the largest and larger than the current function value's Y_i, is further selected to update the status of the detective wolf, and the above wandering behavior is repeated until the function value of a detective wolf i, $Y_i > Y_{lead}$ or the number of walks reaches the maximum number of walks T_{1max}.

2.1.4. Running Behavior

In the basic wolf pack algorithm, the wolf position change is determined by step size. For each fixed D dimensional space, the corresponding $[min_d, max_d]$is fixed, so the step size corresponding to each iteration is fixed. If the step is too large, the accuracy of algorithm optimization will be affected. If the step is too small, the convergence speed of the algorithm will be affected. In other words, when the maximum number of iterations is reached, the optimal solution has not been found. Referring to the idea of Guo (2018) [25], the step size of wolf i in each move is determined by the current position of the wolf and the position of the current head wolf. Therefore, adaptive step size is adopted in the attack and siege behavior as follows Equation 8:

$$step = rand \times \|x_i - X_{lead}\|_2, \quad d = 1, 2, \cdots D \tag{8}$$

In Equation 8, *rand* represents random number between [0, 1]. When the wolf is far away from the leader, it approaches the leader with a larger stride, accelerating the convergence speed and avoiding unnecessary search. When close to the leader, approaching the leader with small steps to improve the precision of search.

Different from previous wolf pack algorithms, this paper randomly selects all the wolves except the head wolf to participate in the summoning, not only the artificial wolves near the head wolf. In the course of the fierce wolf's running, when one of the fierce wolves perceives that the concentration of prey breath is higher at its location, it would replace the head wolf, selected the fierce wolf again and called until the concentration of prey breath at its location was lower than that of the head wolf. At the same time, the step size of summoning behavior adopts Equation 8, then the fierce wolf updates the current position according to Equation 9:

$$x_{id}^{*} = x_{id} + rand \times \|x_{id} - x_{lead}\|_2 \times (x_{leadd} - x_{id}) / |x_{leadd} - x_{id}|, \quad d = 1, 2, \cdots D \tag{9}$$

Where: x_{id}^* represents the updated position of the wolf, x_{id} is the current position of the wolf, and x_{leadd} is the position of the head wolf.

2.1.5. Siege Behavior

Meanwhile, the wolves teamed up to hunt their prey. The moving step size is expressed in Equation 8, and the siege behavior of wolves is expressed in Equation 10:

$$x_{id1} = x_{id} + \lambda \times rand \times \|x_{id} - G_d\|_2 \times |G_d - x_{id}|, \quad d = 1, 2, \cdots D \tag{10}$$

Where: G_d is the location of prey in d dimensional space, and λ is a random number evenly distributed between (-1, 1).

2.1.6. Update Mechanism of "Strong Survival" Wolves

The distribution of prey follows the principle of "from strong to weak", causing the weak wolf to starve to death. That is, the R artificial wolves with the worst objective function value are removed in the algorithm, and the R artificial wolves are randomly generated. In the actual hunting process, the number of each hunting is random, which also leads to the elimination of different numbers of weak wolves. Based on this, β takes a random integer between $[n/(2\times\beta), n/\beta]$, and β is the update scale factor.

2.2. Improved Description of the Wolf Pack Algorithm

Step 1 Initialize the number N of artificial wolves in the wolves and its location X_i, the maximum number of iterations K_{max}, the wolf scale factor α, the update scale factor β, the maximum number of walks T_{1max}, and the maximum number of strikes T_{2max}.

Step 2 The head wolf is determined according to the rules of the head wolf.

Step 3 The wolf detective performs the walking behavior according to Levi flight strategy Equations 3 to 7, until the function value of a certain wolf i is $Y_i > Y_{lead}$ or the number of walk reaches the maximum number of walk T_{1max}, go to step4.

Step 4 The fierce wolves perform the raid and attack the prey according to Equation 9. In the process of attack, if the function value of the prey's breath concentration is $Y_i > Y_{lead}$, then $Y_i = Y_{lead}$, the fierce wolf would be converted into head wolf and initiate summoning behavior; if $Y_i < Y_{lead}$, then continue to run until the function value of a wolf is less than the function value of the head wolf or the running number reach the maximum number of strikes T_{2max}, and turn to step5.

Step 5 According to Equation 10, update the position of the artificial wolf participating in the siege and conduct a siege.

Step 6 Implement the update mechanism of wolves.

Step 7 Determine whether the algorithm satisfies the optimization accuracy requirement or the maximum number of iteration K_{max}. If the requirement is met, the head wolf position is output, that is, the optimal solution of the problem sought, otherwise go to step2.

The whole improved wolf pack system flow chart is shown in Figure 1.

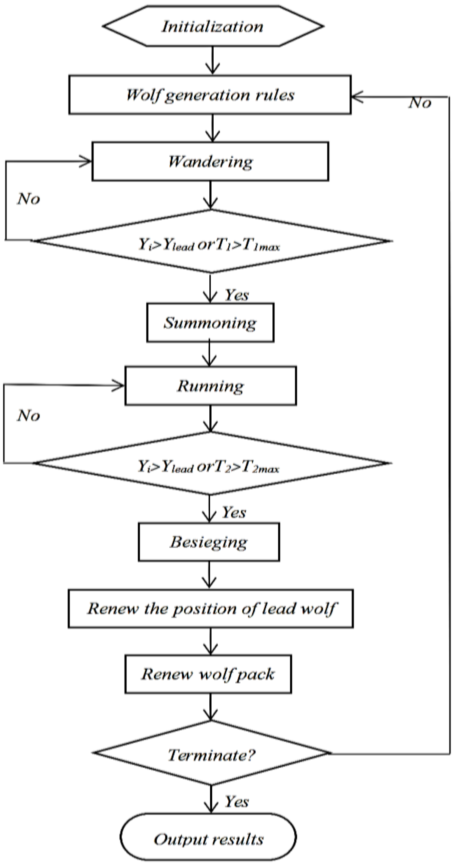


Figure 1. Wolf System Running Flow Chart

3. Experimental Simulation and Analysis

3.1. Basic Test Functions and Parameter Settings

The "*U*" in Table 1 indicates that the function is a unimodal one, "*M*" indicates a multimodal one, "*S*" indicates a separate one, and "*N*" indicates a non-separable one. The unimodal function has global optimal values but no local extreme value in the domain. Multimodal functions are more complex than single-mode functions. It is difficult for general algorithms to find global optimal values with multiple local optimum, is easy to fall into local extremes or oscillations between local values of the leader [26]. Therefore, multimodality is often used to test the global search performance of the algorithm and the ability to avoid premature convergence [27]. If a function with N variables can be expressed as the sum of N univariate functions, then it is separable, otherwise, it is inseparable [28]. Since the relationship between non-separable function variables is complex, it is difficult to find the global optimal value of the inseparable function [29]. In addition, for low-dimensional functions, many algorithms perform outperform high-dimensional complex functions [30]. Parameters involved in the WPA, UWPA and IWPA algorithms are set with reference to [16, 19, 21]. In this paper, N takes 50, α takes 4, K_{max} takes 1000, β takes 4, $T_{1max}(T_{2max})$ takes 10.

3.2. Algorithm Comparison Verification

In order to fully calculate the performance of the algorithm, LWPA, UWPA, WPA and IWPA algorithms were used respectively to perform 100 consecutive optimization calculations on 15 complex functions. The algorithm was evaluated from six indicators in Table 2. When the relationship between different calculation results and the optimal value exceeded e^{-3}, it was considered a failure. The results were shown in Table 2.

Table 1. 15 Functions for Testing Algorithm Performance

Numbering	Function	Expression	Dimension	Feature	Ranges	Theoretical Optimal Solution
1	Eason	$-\cos x_1 \cos x_2 \times \exp(-(x_1-\pi)^2-(x_2-\pi)^2)$	2	UN	[-100,100]	$\min f = -1$
2	Matyas	$f(X)=0.26(x_1^2+x_2^2)-0.48x_1x_2$	2	UN	[-10,10]	$\min f = 0$
3	Booth	$f(X)=(x_1+2x_2-7)^2+(2x_1+x_2-5)^2$	2	MS	[-10,10]	$\min f = 0$
4	Bohachevs1	$f(X)=x_1^2+2x_2^2-0.3\cos(3pix_1)-$ $0.4\cos(4pix_2)+0.7$	2	MS	[-100,100]	$\min f = 0$
5	Eggcrate	$f(X)=x_1^2+x_2^2+25(\sin^2 x_1+\sin^2 x_2)$	2	MS	[-pi,pi]	$\min f = 0$
6	Schaffer	$f(X)=0.5+\frac{\left(\sin\sqrt{(x_1^2+x_2^2)}\right)^2-0.5}{\left(1+0.001(x_1^2+x_2^2)\right)^2}$	2	MN	[-100,100]	$\min f = 0$
7	Six Hump Camel Back	$f(X)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	2	MN	[-5,5]	$\min f = -1.0136$
8	Bohachevs3	$f(X)=x_1^2+2x_2^2-0.3\cos(3pix_1+4pix_2)+0.3$	2	MN	[-100,100]	$\min f = 0$
9	Bridge	$f(X)=\frac{\sin\sqrt{(x_1^2+x_2^2)}}{\sqrt{(x_1^2+x_2^2)}}+\exp\left(\frac{\cos 2pix_1+\cos 2pix_2}{2}\right)$ -0.7129	2	MN	[-1.5,1.5]	$\min f = 0$
10	Trid6	$\sum_{i=1}^{D}(x_i-1)^2-\sum_{i=2}^{D}x_ix_{i-1}$	6	UN	[-36,36]	$\min f = -50$
11	Sumsquares	$\sum_{i=1}^{D} ix_i^2$	10	US	[-10,10]	$\min f = 0$
12	Sphere	$\sum_{i=1}^{D} x_i^2$	30	US	[-1.5,1.5]	$\min f = 0$
13	Rastrigin	$\sum_{i=1}^{D}\left[x_i^2-10\cos 2\pi x_i+10\right]$	60	MS	[-10,10]	$\min f = 0$
14	Quadric	$\sum_{i=1}^{D}(\sum_{k=1}^{i}x_k)^2$	120	MS	[-30,30]	$\min f = 0$
15	Ackley	$-20\exp(-0.2\sqrt{(\frac{1}{D}\sum_{i=1}^{D}x_i^2)}$ $-\exp(\frac{1}{D}\sum_{i=1}^{D}\cos 2\pi x_i)+20+e$	200	MN	[-32,32]	$\min f = 0$

Table 2. Four Algorithms are Applied to the Comparison of 15 kind of Functions

Numbering	Function	Method	Optimal Value	Worst Value	Average Value	Expectation	Success Rate %	Time Consuming/S
1	Eason	LWPA	-1	-1	-1	1e-7	100	1.38406
		UWPA	-1	-0.999985	-0.999999	0	100	2.75053
		WPA	-0.90099	-0.90099	-0.2606	0.1601	12	33.1102
		IWPA	-0.99994	-0.952511	-0.985008	1.59e-4	34	3.9878
2	Matyas	LWPA	1.43e-8	1.59e-5	4.16e-6	2.03e-11	100	1.49761
		UWPA	8.22e-07	0.02727	0.00433	4.24e-05	86	2.91316
		WPA	-127.83	-6897.11	-2474.76	1426.25	21	35.1129
		IWPA	-41.57	-2.6805	-19.5752	105.045	46	7.3495
3	Booth	LWPA	1.96e-9	5.10e-6	5.76e-6	1.50e-10	100	1.55476
		UWPA	3.41e-8	6.06e-8	1.24e-8	1.81e-8	100	1.9993
		WPA	2	2.00101	2.00001	1.0808	32	29.5487
		IWPA	1.44e-9	0.11456	0.01852	6.64e-4	47	3.4539
4	Bohachevs1	LWPA	1.09e-7	2.37e-6	8.20e-7	3.34e-13	100	3.0969
		UWPA	6.79e-3	0.59589	0.16607	0.02667	98	4.40591
		WPA	7.69e-10	171.208	14.0621	915.964	11	34.0251
		IWPA	2.59e-13	7.58e-3	1.67e-4	6.63e-7	91	3.4207
5	Eggcrate	LWPA	2.50e-8	2.94e-6	9.21e-7	6.54e-13	100	0.98280
		UWPA	2.34e-5	6.81e-4	1.75e-4	2.15e-8	100	7.99349
		WPA	6.07e-13	9.21e-7	6.68e-8	1.69e-14	100	43.9853
		IWPA	9.56e-17	5.24e-6	1.59e-7	4.13e-13	100	3.6798
6	Schaffer	LWPA	2.95e-8	0.00098	0.00082	1.03e-5	100	0.88572
		UWPA	4.76e-4	0.03727	0.01027	3.47e-5	48	1.50288
		WPA	3.02e-11	0.12699	0.01926	7.72e-3	12	63.9457
		IWPA	1.478e-9	0.03722	0.01310	8.97e-5	24	6.8185
7	Six Hump Camel Back	LWPA	-1.0316	-1.0316	-1.0316	0	100	1.04521
		UWPA	-1.0316	-1.0312	-1.0315	8.14e-9	100	5.35863
		WPA	-1.85e-8	-4.47e-13	-2.27e-9	1.49e-17	8	59.4475
		IWPA	-1.0316	-0.5329	-0.9738	7.50e-3	12	3.5639
8	Bohachevs3	LWPA	1.28e-7	8.74e-4	3.72e-4	4.50e-7	100	2.5020
		UWPA	2.17e-5	0.26921	0.05028	2.72e-3	13	3.40373
		WPA	6.14e-7	175.978	9.33313	616.702	14	33.8626
		IWPA	2.66e-7	0.2308	4.16e-2	2.89e-2	26	3.5567
9	Bridge	LWPA	-3.0054	-3.0054	-3.0054	0	100	1.1380
		UWPA	-3.00538	-3.0053	-3.0053	1.36e-9	80	15.3302
		WPA	-3.0054	-2.70512	-2.9399	0.01522	6	61.2830
		IWPA	-3.0053	-2.62407	-2.97342	4.48e-3	8	5.6074
10	Trid6	LWPA	-50	-48.495	-49.7808	0.20695	95	7.1294
		UWPA	-127.83	-6897.11	-2474.76	1426.25	95	35.1129
		WPA	-33.5	-33.5	-33.5	1.14e-16	0	7.5918
		IWPA	-41.57	-2.6805	-19.5752	105.045	0	7.3495
11	Sumsquares	LWPA	1.08e-6	4.71e-6	2.49e-6	1.69e-12	100	2.6682
		UWPA	1.02e-5	3.41e-4	9.99e-5	3.88e-9	100	1.6034
		WPA	3.00538	6897.11	2326.73	1.88e+6	0	20.9048
		IWPA	3.34e-6	7.96e-2	2.91e-3	8.99e-5	70	11.5815
12	Sphere	LWPA	2.27e-7	4.56e-7	3.66e-7	5.30e-15	100	1.87461
		UWPA	0.00102	0.00571	0.0023	6.09e-7	80	1.55440
		WPA	3.00538	6897.11	1790.03	2.94e+4	0	67.0066
		IWPA	6.42e-4	0.28189	0.01368	1.17e-3	1	26.6643
13	Rastrigin	LWPA	2.11e-10	3.68e-8	1.43e-8	1.16e-8	100	8.56372
		UWPA	52.7886	157.196	100.765	240.213	0	6.9872
		WPA	2082.85	3105.82	2497.08	523.516	0	207.87
		IWPA	2.29e+2	4.85e+2	3.43e+2	4.23e+3	0	56.7631
14	Quadric	LWPA	6.85e-16	1.50e-5	1.61e-5	1.42e-11	100	29.8425
		UWPA	3474.1	500129	128464	5.36e+9	0	27.5142
		WPA	7.90e+10	5.20e+10	1.24e+10	1.19e+10	0	377.928
		IWPA	7.61e+7	3.03e+8	1.60e+8	1.17e+15	0	140.585
15	Ackley	LWPA	1.155	2.81	2.25	0.5136	100	37.7076
		UWPA	2.21122	3.90741	3.07391	0.11495	0	66.8509
		WPA	20.5291	21.4804	21.1953	0.1876	0	367.249
		IWPA	19.38	20.1071	19.8043	2.41e-2	0	232.951

Table 3. Effect of T_{max} on the algorithm

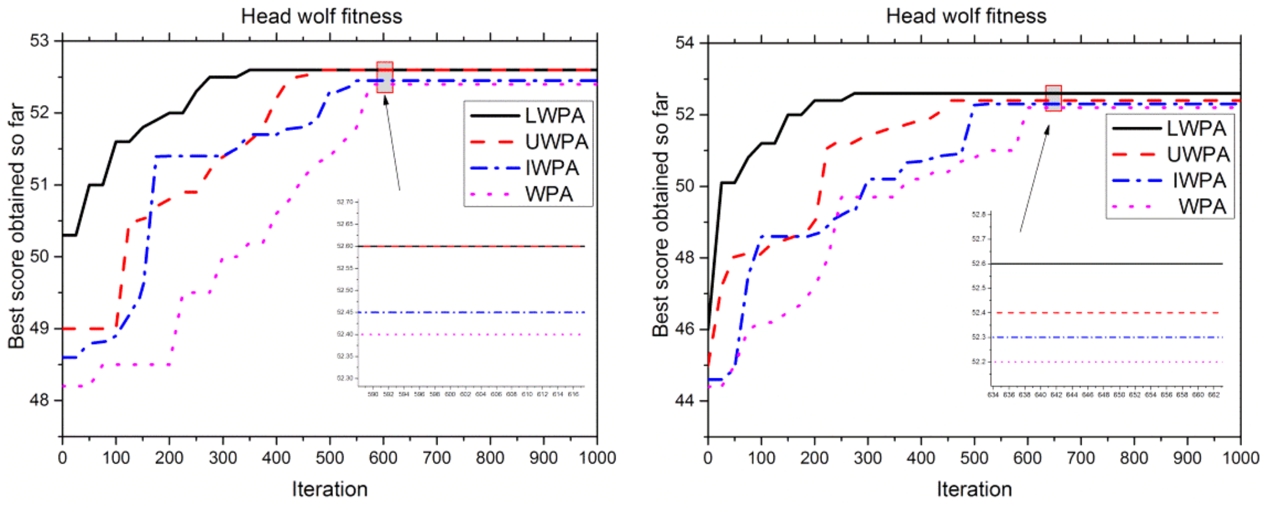
Function	**Standard deviation**					
	6	**8**	**10**	**12**	**14**	**16**
Eason	2.40E-08	8.40E-09	9.60E-09	1.40E-09	2.60E-08	2.10E-08
Booth	3.20E-11	4.20E-11	7.50E-11	4.10E-11	3.20E-11	2.60E-11
Bridge	5.60E-15	5.60E-15	5.60E-15	5.60E-15	5.60E-15	5.60E-15
Sumsquares	3.20E-07	3.40E-07	5.50E-07	3.80E-07	3.60E-07	3.00E-07
Sphere	7.10E-14	4.50E-15	7.80E-16	1.90E-15	8.70E-14	6.90E-14
Quadric	4.10E-16	2.40E-19	4.20E-19	8.30E-18	8.50E-17	5.40E-16
Ackley	5.30E-17	5.30E-17	5.30E-17	5.30E-17	5.30E-17	5.30E-17

Table 4. Effect of β on the algorithm

Function	Standard deviation					
	2	3	4	5	6	7
Eason	4.00E-08	8.70E-08	1.20E-09	8.60E-08	6.40E-08	5.90E-08
Booth	6.10E-11	9.50E-11	1.90E-12	8.50E-11	7.50E-11	6.30E-11
Bridge	5.60E-15	5.60E-15	5.60E-15	5.60E-15	5.60E-15	5.60E-15
Sumsquares	3.20E-07	4.40E-07	1.20E-08	5.40E-07	3.90E-07	3.40E-07
Sphere	1.60E-14	5.20E-15	2.90E-16	4.30E-15	3.90E-15	2.30E-14
Quadric	2.80E-17	1.40E-18	2.80E-19	8.30E-16	6.90E-15	5.30E-15
Ackley	5.30E-17	5.30E-17	5.30E-17	5.30E-17	5.30E-17	5.30E-17

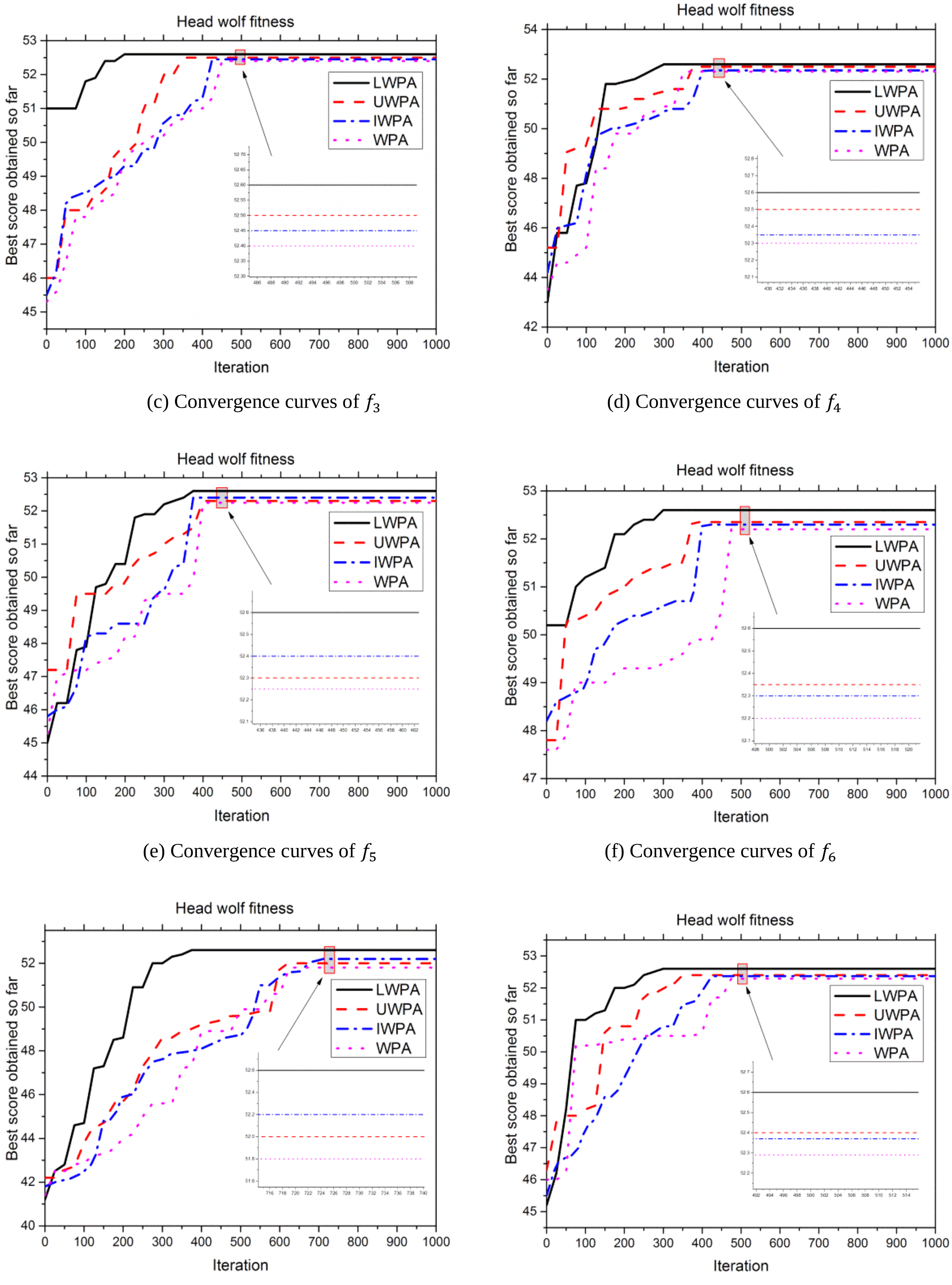
3.3. Analysis of Main Parameters of LWPA

Although LWPA had some advantages, it involved many parameters, and the main parameters had different effects on the performance of the algorithm. T_{1max} (T_{2max}) is the maximum number of times in the process of wolf migration (running), and β is the proportional coefficient of wolf renewal. According to the characteristics of the 15 functions, they are divided into seven categories. And the sizes of T_{max} and β are changed respectively to conduct 50 optimization calculations for these 7 functions. The effects of T_{max} and β on the algorithm performance is shown in Tables 3 and 4.



(a) Convergence curves of f_1

(b) Convergence curves of f_2



(c) Convergence curves of f_3

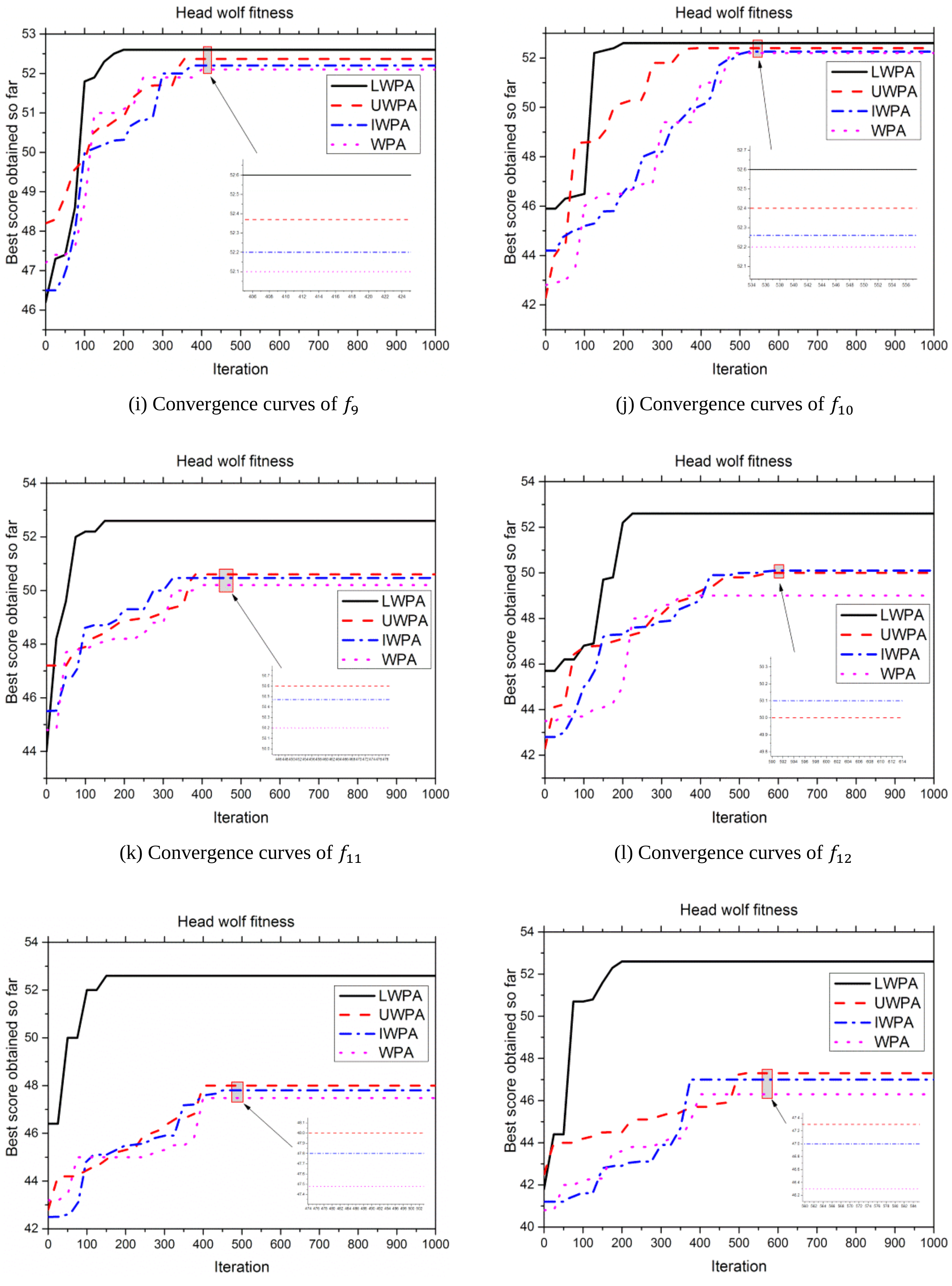
(d) Convergence curves of f_4

(e) Convergence curves of f_5

(f) Convergence curves of f_6

(g) Convergence curves of f_7

(h) Convergence curves of f_8



(i) Convergence curves of f_9

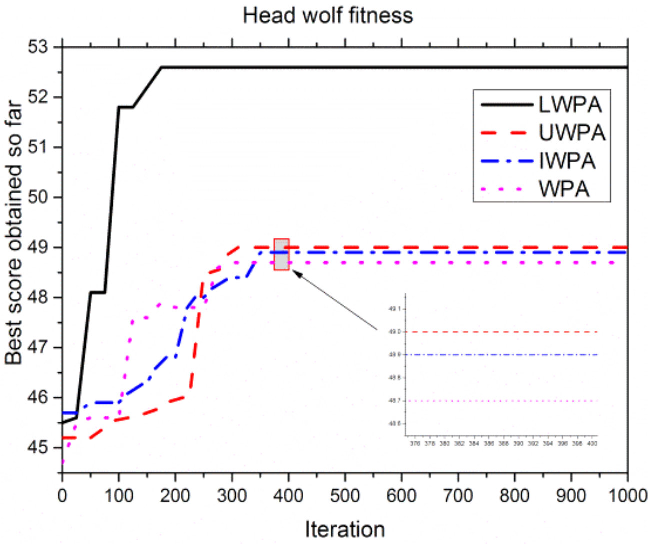
(j) Convergence curves of f_{10}

(k) Convergence curves of f_{11}

(l) Convergence curves of f_{12}

(m) Convergence curves of f_{13}

(n) Convergence curves of f_{14}



(o) Convergence curves of f_{15}

Figure 2. Convergence curves of 15 test functions

3.4. Results Analysis

a) It can be seen from the comparison results of various algorithms in Table 2. Eason and Matyas are unimodal, low-dimensional and non-separable functions, LWPA and UWPA have been successfully optimized and have good performance, they both were close to the optimal. Even the precision of standard deviation of UWPA was better than LWPA, while the precision of IWPA and WPA was poor. In terms of time, UWPA consumed twice as much time as LWPA.

b) Booth, Bohachevs1 and Eggcrate are multimodal, low-dimensional separable functions, the convergence accuracy of LWPA was significantly higher than that of the other three algorithms, reaching more than 1e-6. In terms of time consumption, LWPA and UWPA took the shortest time, followed by IWPA, and WPA took the longest.

c) Schaffer, Six Hump Camel Back, Bohachevs3 and Bridge are multimodal, low-dimensional non-separable functions, WPA and IWPA were immersed in local optimization, leading to the failure of optimization, LWPA and UWPA were successfully optimized and LWPA had better optimization performance. Meanwhile, LWPA had the shortest time, which was significantly less than the other three algorithms.

d) For Trid6, a unimodal, high-dimensional non-separable function, LWPA and UWPA were successfully optimized and perform better. In terms of time consumption, except for WPA, which took a long time, the other three algorithms had the same time consumption.

e) Sumsquares and Sphere are unimodal, high-dimensional separable functions, LWPA and UWPA optimization was successful, and the optimization accuracy of LWPA was significantly better than the other three algorithms, reaching 1e-7 or more. However, in terms of time consumption, UWPA was slightly better than LWPA.

f) Rastrigin and Quadric are multimodal, high-dimensional and separable functions, as the number of dimensions increased, only LWPA optimization was successful and the accuracy was high. Both LWPA and UWPA took less time.

g) Ackley, a multimodal, high-dimensional and non-separable function, only LWPA optimization was successful, and the time was also the shortest.

In conclusion, both in terms of accuracy and time consumption, LWPA was more accurate and effective in handling function problems than other three algorithms, especially for multimodal, high-dimensional complex functions, the effect was better. The effect of UWPA was second, IWPA and WPA were worse.

It can be seen from Table 3 that as T_{max} increased, the standard deviation showed a tendency to decrease first and then increase. If the value of T_{max} is too small, the search efficiency of the wolves would be reduced, and it would take a long time to find the optimal solution. If the value of T_{max} is too large to ignore the optimal solution, the required accuracy cannot be achieved. In summary, T_{max} takes 10.

As can be seen from Table 4, as T increased, the standard deviation tended to decrease first and then increase. If β is too small, the updated number of wolves is too large, which makes it difficult for the wolves to gather, resulting in the reduced optimization effect of the algorithm. However, if β is too large, the updated number of wolves is too small, resulting in a sharp decline in the diversity of wolves, and it is easy to fall into local optimization. To sum up, β takes 4.

To further illustrated the superiority of the LWPA algorithm, Figure 2 showed the convergence curves of the LWPA, UWPA, IWPA, and WPA in each test function. As can be seen from the figures, for unimodal, low-dimensional, non-separable and complex functions, when the algorithm iterated to 300 times, the LWPA algorithm had found the optimal value and tended to be stable, while the other three algorithms also tended to be stable but had poor accuracy when iterating to 400 times. For simple low-dimensional functions, UWPA worked best in the early search, but in the latter part of the algorithm, the search efficiency was not high, it took to a long time and easy to fall into a local optimal. For complex functions, the convergence accuracy of LWPA was significantly higher than the other three algorithms. When the number of dimensions increased to 30, 60, 120, or even 200 dimensions, the optimization effect of UWPA, IWPA, and WPA was obviously poor, resulting in a long search time in the early stage and a local optimum in the later stage. Therefore, the LWPA has been improved very well for the problem that it was liable to fall into the local optimum in the later stage. From the above comparison, it can be seen that compared with the other three algorithms, the optimization performance of the LWPA has improved significantly in terms of convergence speed and convergence accuracy, which further illustrated the correctness of the improvement direction.

4. LWPA for Optimization of Truss Structures

4.1. Truss Structural Design Optimization

4.1.1. Optimization Model

The problem of truss optimization model with cross-sectional area as design variable can be described as Equations 11 and 12:

$$\min F = W(x) \, , \tag{11}$$

$$s.t. \, g_i(x) \leq 0, i = 1, 2, \Lambda, p \tag{12}$$

Where: $g_i(x)$ is the constraint function; p is the number of constraints.

4.1.2. Objective Function

$$W(A) = \sum_{i=1}^{n} \gamma A_i L_i \tag{13}$$

In Equation 13: $W(A)$ is the weight of the structure, A_i is the cross-sectional area of the i member, L_i is the length of the i member, γ is the material density, n is the number of design variables.

4.1.3. Constraint Condition

Each bar must meet the requirements for strength, stiffness, stability, and cross-sectional dimensions, and the constraints are as follows Equations 14 to 16：

$$\frac{\sigma_i}{[\sigma]} - 1 \leq 0 \, , \tag{14}$$

$$\frac{\mu_j}{\mu_{\max}} - 1 \leq 0 \, , \tag{15}$$

$$\mathrm{A}_{\min} \leq \mathrm{A}_i \leq \mathrm{A}_{\max} \, . \tag{16}$$

In the Equation: σ_i is the axial normal stress of the i rod; $[\sigma]$ is the allowable stress of the material; μ_j is the displacement of node j ; $\mu_{\max}$ is the allowable displacement of node j ; $\mathrm{A}_{\min}$、 $\mathrm{A}_{\max}$ are the upper and lower limits of the section of the member respectively.

5. Validation: Structural Design Experiments

5.1. A 25-bar Spatial Truss

The topology and nodal numbers of a 25-bar spatial truss structure are shown in Figure 3. The material density is considered as 2768 kg/m^3 and the modulus of elasticity is taken as 68950 Mpa. Twenty-five members of this truss are divided into eight groups of cross-sectional areas as design variables, and they are shown in Table 5. Results of optimization are reported in Table 6 admitting superiority of LWPA relative to other algorithms, because it reaches to fine solution after a number of iterations less than those of the others. Convergence curves of best runs for LWPA and other three algorithms are drawn in Figure 4.

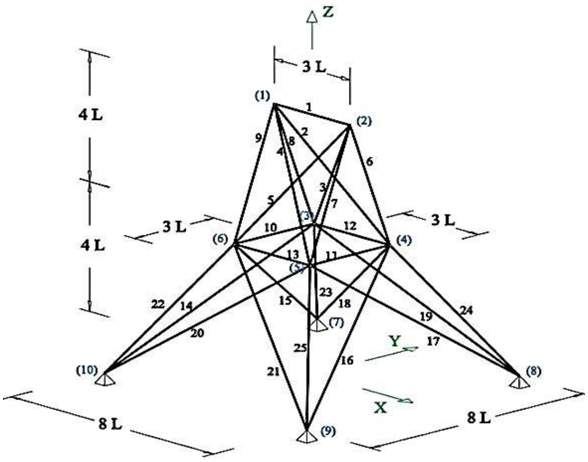


Figure 3. The 25-bar Spatial Truss Structure

Table 5. Load Cases of the 25-bar Spatial Truss Structure

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

Table 6. Comparison of Optimal Designs for the 25-bar Spatial Truss Structure

Element group	Optimal cross-sectional area（cm^2）			
	(GA)	(FA)	(WPA)	(LWPA)
A_1	0.0643	0.0643	0.0643	0.0643
$A_2 - A_5$	14.243	14.318	14.49	12.229
$A_6 - A_9$	17.925	18.862	18.952	19.482
$A_{10} - A_{11}$	0.0643	0.0643	0.0643	0.0643
$A_{12} - A_{13}$	0.0643	0.0643	0.0643	0.0643
$A_{14} - A_{17}$	4.4807	4.8872	4.3382	4.3641
$A_{18} - A_{21}$	10.318	9.9621	9.7581	10.949
$A_{22} - A_{25}$	17.551	17.042	17.78	17.161
Weight (kg)	265.92	249.76	246.67	243.11

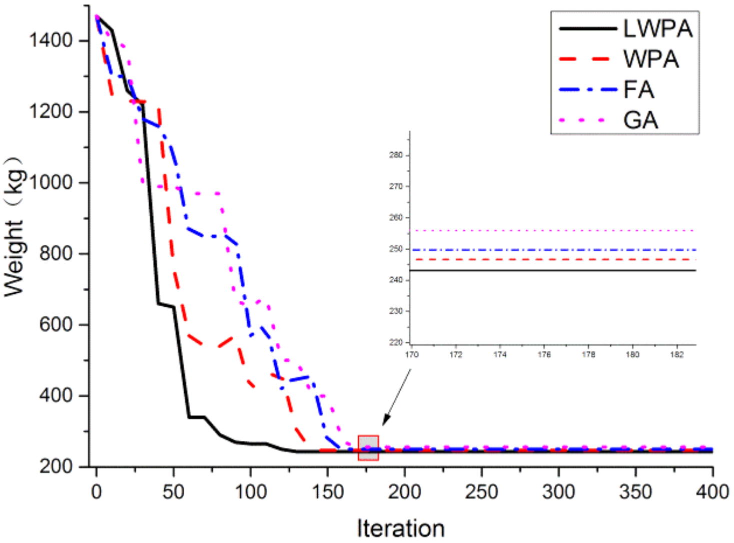


Figure 4. Convergence curve of 25 bar truss structure

5.2. A 72-bar Spatial Truss

A 72-bar spatial truss structure is shown in Figure 5. The cross-sectional areas members are employed as design variables and are parted into 16 groups. The material density is considered as 2768 kg/m^3 and the modulus of elasticity is taken as 68950 Mpa, and the maximum displacement of each bar in all directions can't exceed 6.35 mm, and maximum allowable stress is [-172.375, 172.375], and the optimized results are shown in Table 7. Form Table 7, the total weight of the optimized structure is 321.58kg, and it is less than the GA for (348.71-321.58)/321.58=8.43%; and it is less than the FA for (334.58-321.58)/321.58=4.04%; and it is less than the standard WPA for (329.13-321.58)/321.58=2.34%. Meanwhile, it can be seen from Figure 6 that the improved wolf pack algorithm can search for the global optimal solution, which has higher convergence speed and convergence accuracy than GA, FA, and WPA. Especially in the early stage of iterative calculation, the effect is very obvious. Obviously, the optimization effect obtained by the improved wolf pack algorithm is better than GA, FA and WPA.

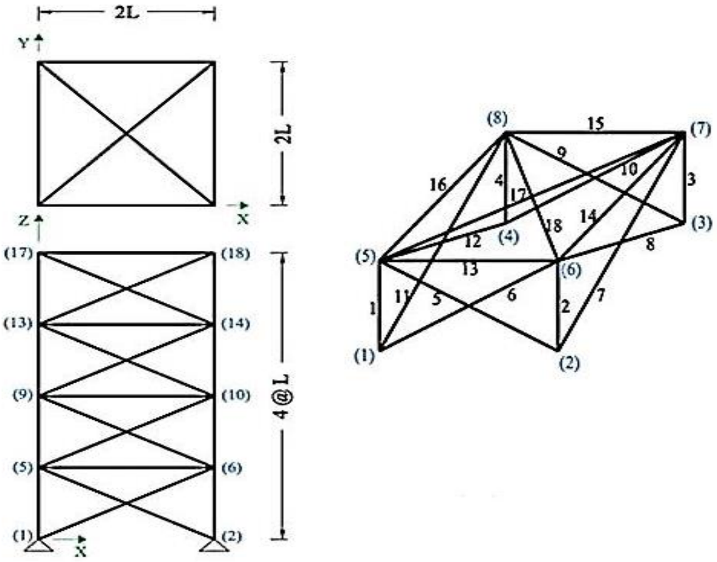


Figure 5. The 72-bar Spatial Truss Structure

Table 7. Comparison of Optimal Results for the 72-bar Spatial Truss Structure

Element group	Optimal cross-sectional area（cm^2）			
	(GA)	(FA)	(WPA)	(LWPA)
$A_1 - A_4$	2.9872	3.3411	3.3437	3.5017
$A_5 - A_{12}$	7.8591	7.7587	7.8688	7.934
$A_{13} - A_{16}$	0.643	0.643	0.643	0.643
$A_{17} - A_{18}$	0.643	0.643	0.643	0.643
$A_{19} - A_{22}$	8.765	9.0202	8.1626	8.0215
$A_{23} - A_{30}$	8.1461	8.2477	7.9612	7.9796
$A_{31} - A_{34}$	0.643	0.643	0.6432	0.643
$A_{35} - A_{36}$	0.643	0.643	0.643	0.643
$A_{37} - A_{40}$	13.45	12.045	12.266	12.817
$A_{41} - A_{48}$	8.0732	8.0401	8.1845	8.1129
$A_{49} - A_{52}$	0.6454	0.645	0.6451	0.645
$A_{53} - A_{54}$	0.6451	0.645	0.6451	0.645
$A_{55} - A_{58}$	16.685	17.382	17.963	17.336
$A_{59} - A_{66}$	8.1592	8.0561	8.1292	8.101
$A_{67} - A_{70}$	0.643	0.643	0.643	0.643
$A_{71} - A_{72}$	0.643	0.643	0.643	0.643
Weight（kg）	348.71	334.58	329.13	321.58

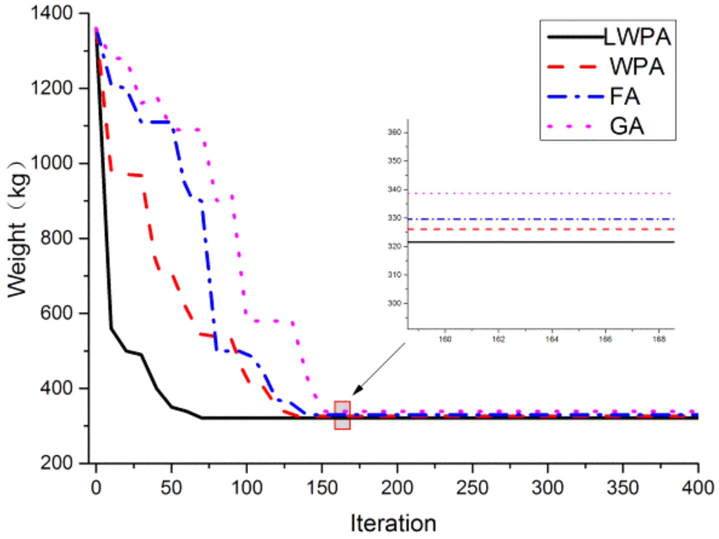


Figure 6. Convergence curve of 72 bar truss structure

5.3. A 200-bar Planar Truss

The 200-bar planar truss structure is shown in Figure 7. The 200 structural members of this planar truss are categorized as 29 groups using symmetry. It can be seen from the Table 8 that the structure weight of LWPA is less than that of GA for (12956.8-11573.4)/11573.4=11.95%; it is less than that of FA for (12171.2-11573.4)/11573.4= 5.16%; it is less than that of the standard for (11951.6-11573.4)/11573.4=3.26%; Meanwhile, it can be seen from the iterative relationship curve in Figure 8 that LWPA can search for the global optimal solution, which has higher convergence speed and accuracy than GA, FA and WPA, especially in the initial effect of the iterative algorithm. When the number of iterations reaches about 140 times, it basically converges to the global optimal solution. Obviously, the improved algorithm in this paper is superior to the algorithms in the literature in terms of optimization effect.

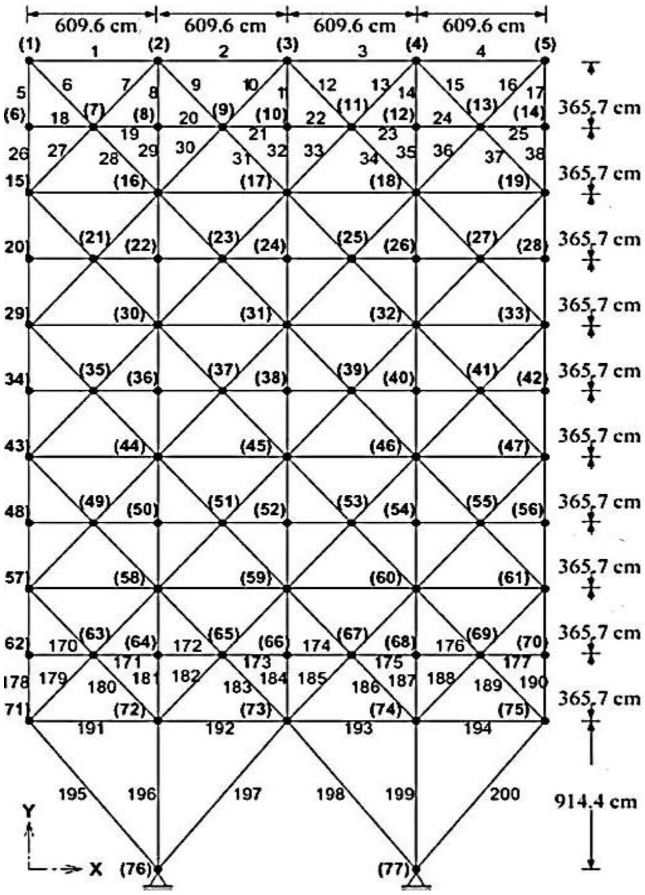


Figure 7. The 200-bar Spatial Truss Structure

Table 8. Comparison of Optimal Results for the 200-bar Spatial Truss Structure

Element group	**Optimal cross-sectional area（cm^2）**			
	(GA)	**(FA)**	**(WPA)**	**(LWPA)**
A_1	2.238	0.665	0.645	0.677
A_2	6.947	6.107	6.102	6.956
A_3	0.645	0.664	0.645	1.126
A_4	0.645	0.665	0.645	0.68
A_5	13.82	19.54	12.21	12.64
A_6	2.239	1.461	1.74	2.171
A_7	0.645	0.645	0.645	0.677
A_8	23	27.27	31.19	19.08
A_9	2.339	0.664	3.281	0.677
A_{10}	31	33.92	25.69	29.68
A_{11}	2.839	2.073	2.984	2.582
A_{12}	38.4	32.21	34.38	33.82
A_{14}	2.239	3.479	4.674	0.677
A_{15}	42.4	38.59	39.88	49.23
A_{16}	6.155	4.56	4.986	2.946
A_{17}	2.239	0.665	0.645	0.677
A_{18}	55	48.42	64.16	48.69
A_{19}	0.645	0.645	0.645	3.758
A_{20}	60	54.8	58.48	54.61
A_{21}	6.155	3.676	5.356	5.426
A_{22}	11.38	9.433	1.635	0.677
A_{23}	85.81	65.54	71.81	67.47
A_{24}	2.239	0.665	1.497	3.36
A_{25}	85.81	72.71	78.75	73.57
A_{26}	13.82	8.183	7.261	7.067
A_{27}	31	46.34	35.62	44.34
A_{28}	60	78.49	63.08	66.46
A_{29}	110.8	85.75	91.83	86.98
Weight（kg）	12956.8	12171.2	11951.6	11573.4

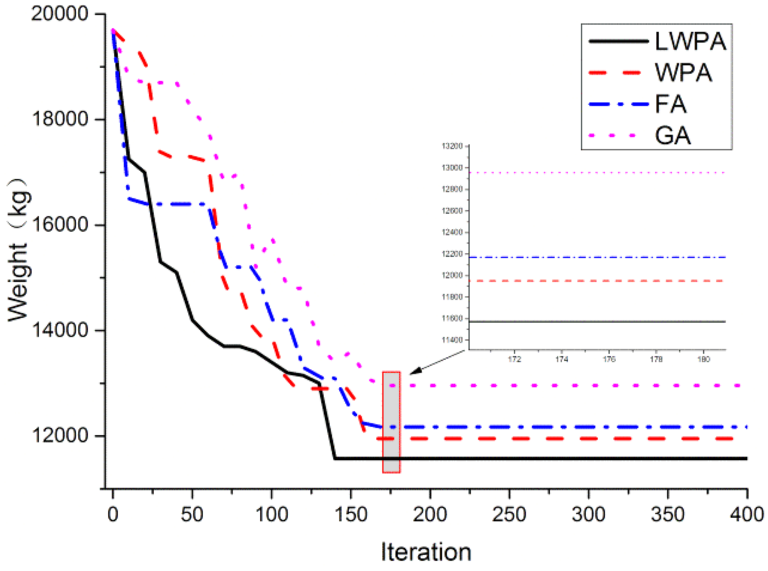


Figure 8. Convergence curve of 200 bar truss structure

6. Conclusion

Structural design optimization is a problem we have to discuss in the course of social development, and how to intelligently use intelligent algorithms to solve structural optimization is the key. In this paper, a new improved WPA algorithm is proposed for optimal design of truss structures. To enhance WPA, a few step and strategy are introduced to provide the algorithm with suitable flexibility for challenging optimization problems like those incorporated in structural optimization. Fifteen test functions and three truss structure cases are employed to show the performance of the presented LWPA compared to WPA and other well-known algorithms. Optimization results and comparison with other algorithms show that the stability of LWPA optimization algorithm is apparently improved and the efficiency is obviously remarkable. In general, the improved wolf pack algorithm is not only suitable for high-dimensional and complex functions, but also effectively solves the problems of several common truss structural design optimization, providing a good ideas for more complex structural optimization problems. Future research will focus on how to combine the improved wolf pack algorithm with other intelligent algorithms to solve engineering optimization problems.

7. Funding

This work was supported by Hebei Province Innovation Ability Promotion Plan Project (No.19456102D) and the Project of Scientific Research Program of Colleges and Universities of Hebei Province (No. ZD2019114).

8. Conflicts of Interest

The authors declare no conflict of interest.

9. References

[1] Mayencourt, Paul, and Caitlin Mueller. “Hybrid Analytical and Computational Optimization Methodology for Structural Shaping: Material-Efficient Mass Timber Beams.” Engineering Structures 215 (July 2020): 110532. doi:10.1016/j.engstruct.2020.110532.

[2] Baghdadi, Abtin, Mahmoud Heristchian, and Harald Kloft. “Design of Prefabricated Wall-Floor Building Systems Using Meta-Heuristic Optimization Algorithms.” Automation in Construction 114 (June 2020): 103156. doi:10.1016/j.autcon.2020.103156.

[3] Beheshti, Zahra, and Siti Mariyam Shamsuddin. “Non-Parametric Particle Swarm Optimization for Global Optimization.” Applied Soft Computing 28 (March 2015): 345–359. doi:10.1016/j.asoc.2014.12.015.

[4] Li, Wenjing, Yingzhou Bi, Xiaofeng Zhu, Chang-an Yuan, and Xiang-bo Zhang. “Hybrid Swarm Intelligent Parallel Algorithm Research Based on Multi-Core Clusters.” Microprocessors and Microsystems 47 (November 2016): 151–160. doi:10.1016/j.micpro.2016.05.009.

[5] Ma, Lianbo, Yunlong Zhu, Dingyi Zhang, and Ben Niu. “A Hybrid Approach to Artificial Bee Colony Algorithm.” Neural Computing and Applications 27, no. 2 (March 13, 2015): 387–409. doi:10.1007/s00521-015-1851-x.

[6] Jahangiri, Milad, Mohammad Ali Hadianfard, Mohammad Amir Najafgholipour, Mehdi Jahangiri, and Mohammad Reza Gerami. “Interactive Autodidactic School: A New Metaheuristic Optimization Algorithm for Solving Mathematical and Structural Design Optimization Problems.” Computers & Structures 235 (July 2020): 106268. doi:10.1016/j.compstruc.2020.106268.

[7] John H. Holland, Adaptation in Natural and Artificial Systems, The University of Michigan Press, New York (1975).” Bulletin of Mathematical Biology 38, no. 2 (1976): 211–214. doi:10.1016/s0092-8240(76)80036-5.

[8] VaeziNejad, SeyedMahmood, SeyedMorteza Marandi, and Eysa Salajegheh. “A Hybrid of Artificial Neural Networks and Particle Swarm Optimization Algorithm for Inverse Modeling of Leakage in Earth Dams.” Civil Engineering Journal 5, no. 9 (September 23, 2019): 2041–2057. doi:10.28991/cej-2019-03091392.

[9] Dorigo, M., V. Maniezzo, and A. Colorni. “Ant System: Optimization by a Colony of Cooperating Agents.” IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics) 26, no. 1 (1996): 29–41. doi:10.1109/3477.484436.

[10] Li, Xiao-lei. "An optimizing method based on autonomous animats: fish-swarm algorithm." Systems Engineering-Theory & Practice 22, no. 11 (2002): 32-38.

[11] Karaboga, Dervis. An idea based on honey bee swarm for numerical optimization. Vol. 200. Technical report-tr06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[12] O’Neil, Michael, Franco Woolfe, and Vladimir Rokhlin. “An Algorithm for the Rapid Evaluation of Special Function Transforms.” Applied and Computational Harmonic Analysis 28, no. 2 (March 2010): 203–226. doi:10.1016/j.acha.2009.08.005.

[13] Pan, Wen-Tsao. “A New Fruit Fly Optimization Algorithm: Taking the Financial Distress Model as an Example.” Knowledge-Based Systems 26 (February 2012): 69–74. doi:10.1016/j.knosys.2011.07.001.

[14] Yang, Xin-She. “Flower Pollination Algorithm for Global Optimization.” Lecture Notes in Computer Science (2012): 240–249. doi:10.1007/978-3-642-32894-7_27.

[15] Meng, Xianbing, Yu Liu, Xiaozhi Gao, and Hengzhen Zhang. “A New Bio-Inspired Algorithm: Chicken Swarm Optimization.” Advances in Swarm Intelligence (2014): 86–94. doi:10.1007/978-3-319-11857-4_10.

[16] Wu, H. S., Fengming Zhang, and Lushan Wu. "New swarm intelligence algorithm-wolf pack algorithm." Systems engineering and electronics 35, no. 11 (2013): 2430-2438.

[17] Yang, Chenguang, Xuyan Tu, and Jie Chen. “Algorithm of Marriage in Honey Bees Optimization Based on the Wolf Pack Search.” The 2007 International Conference on Intelligent Pervasive Computing (IPC 2007) (October 2007). doi:10.1109/ipc.2007.104.

[18] Wu, Chenghai, Kaiyu Qin, Penghui He, and Houbiao Li. “An Improved Wolf Colony Search Algorithm Based on Mutual Communication by a Sensor Perception of Wireless Networking.” EURASIP Journal on Wireless Communications and Networking 2018, no. 1 (June 15, 2018). doi:10.1186/s13638-018-1171-9.

[19] Husheng, Wu, and Zhang Fengming. “A Uncultivated Wolf Pack Algorithm for High-Dimensional Functions and Its Application in Parameters Optimization of PID Controller.” 2014 IEEE Congress on Evolutionary Computation (CEC) (July 2014). doi:10.1109/cec.2014.6900432.

[20] Teng, Zhi-jun, Jin-ling Lv, and Li-wen Guo. “An Improved Hybrid Grey Wolf Optimization Algorithm.” Soft Computing 23, no. 15 (June 25, 2018): 6617–6631. doi:10.1007/s00500-018-3310-y.

[21] Chen, Xiayang, Chaojing Tang, Jian Wang, Lei Zhang, and Qingkun Meng. “Improved Wolf Pack Algorithm Based on Differential Evolution Elite Set.” IEICE Transactions on Information and Systems E101.D, no. 7 (July 1, 2018): 1946–1949. doi:10.1587/transinf.2017edl8201.

[22] Zhang, Y., Sun, H., Wei, Z. and Han, B. “Chaos grey wolf optimization algorithm with adaptive adjustment strategy”. Computer Science, Vol. 44, No. 11, (2017):119-123.

[23] Kaveh, A., and P. Zakian. “Improved GWO Algorithm for Optimal Design of Truss Structures.” Engineering with Computers 34, no. 4 (November 30, 2017): 685–707. doi:10.1007/s00366-017-0567-1.

[24] Mo, Y., Nie, H., Liu, Z. and Yang, H. “Grey wolf optimization algorithm based on levy flight”. Microelectronics and Computer, Vol. 36, No. 4, (2019):78-83.

[25] Guo, L. “Improved wolf pack algorithm based on adaptive and variable wandering direction”. Journal of Zhejiang University, Vol. 45, No. 3, (2018):284-293.

[26] Tang, Qin, Yi Shen, Chengyu Hu, Jianyou Zeng, and Wenyin Gong. “Swarm Intelligence: Based Cooperation Optimization of Multi-Modal Functions.” Cognitive Computation 5, no. 1 (April 28, 2012): 48–55. doi:10.1007/s12559-012-9144-5.

[27] Parpinelli, Rafael S., Fábio R. Teodoro, and Heitor S. Lopes. “A Comparison of Swarm Intelligence Algorithms for Structural Engineering Optimization.” International Journal for Numerical Methods in Engineering 91, no. 6 (May 30, 2012): 666–684. doi:10.1002/nme.4295.

[28] Caamaño, Pilar, Francisco Bellas, Jose A. Becerra, and Richard J. Duro. “Evolutionary Algorithm Characterization in Real Parameter Optimization Problems.” Applied Soft Computing 13, no. 4 (April 2013): 1902–1921. doi:10.1016/j.asoc.2013.01.002.

[29] Wu, J. H., Jing Zhang, R. F. Li, and C. H. Liu. "A Multi-Subpopulation PSO Immune Algorithm and Its Application on Function Optimization." Journal of Computer Research and Development 49, no. 9 (2012): 1883-1898.

[30] Wu, Hu-Sheng, and Feng-Ming Zhang. “Wolf Pack Algorithm for Unconstrained Global Optimization.” Mathematical Problems in Engineering 2014 (2014): 1–17. doi:10.1155/2014/465082.