Hindawi
Advances in Civil Engineering
Volume 2019, Article ID 6902428, 16 pages
https://doi.org/10.1155/2019/6902428

Research Article

Truss Structure Optimization Based on Improved Chicken Swarm Optimization Algorithm

Yancang Li,[1] **Shiwen Wang**,[2] **and Muxuan Han**[2]

[1]*College of Water Conservancy and Hydroelectric Power, Hebei University of Engineering, Handan 056038, China*
[2]*College of Civil Engineering, Hebei University of Engineering, Handan 056038, China*

Correspondence should be addressed to Shiwen Wang; tumuwangshiwen@163.com

Received 23 May 2019; Revised 26 July 2019; Accepted 27 August 2019; Published 13 October 2019

Academic Editor: Dimitris Rizos

Copyright © 2019 Yancang Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To improve the efficiency of the structural optimization design in truss calculation, an improved chicken swarm optimization algorithm was proposed for truss structure optimization. The chicken swarm optimization is a novel swarm intelligence algorithm. In the basic chicken swarm optimization algorithm, the concept of combining chaos strategy and reverse learning strategy was introduced in the initialization to ensure the global search ability. And the inertia weighting factor and the learning factor were introduced into the chick position update process, so as to better combine the global and local search. Finally, the overall individual position of the algorithm was optimized by the differential evolution algorithm. The improved algorithm was tested by multipeak function and applied to the truss simulation experiment. The study provided a new method for the truss structure optimization.

1. Introduction

Engineering structure optimization is a problem that has plagued scholars for many years. Mechanical constraints and optimization methods are combined in engineering structure optimization. According to the different engineering requirements, some parameters in the project are involved in the optimization calculation of the engineering structure in the form of variables, thus forming a solution domain of the optimized structure. Then, based on the constraints of the project, the mathematical model is established, and the mathematical solution is used to find the most reasonable design scheme in accordance with the design requirements. In 1974, Schimit and Farshi proposed an approximation problem in structural optimization [1]. After that, with the application of computer technology, modern structure optimization become a real possibility.

The traditional structural optimization method had certain defects. There was no effective optimization design method. At first, it was judged by people directly, such as the limit stress method and the ultimate strain method. These methods were difficult to find the best results, and there was no corresponding theoretical basis. Later, as the time passed, the corresponding mathematical theory support appeared, but it only existed in mathematical theory, and it still cannot describe the actual engineering structure problem. Faced with complex engineering optimization problems, more and more scholars had proposed the theory of bionic intelligent algorithms based on the habit of biological evolution and survival in nature. The optimization of structure was mainly focused on reducing the total quality of the structure, improving the design rationality, and reducing the engineering cost [2–5], for example, improving the application of artificial fish swarm algorithm in truss structure optimization, the truss optimization of artificial bee colony algorithm, and the application of firefly algorithm in truss structure optimization [6–8].

After years of research and development, intelligent optimization algorithms had been widely used in various fields. Fiore et al. proposed an improved differential evolution algorithm to optimize the structure of flat steel trusses. The weight of the steel truss was designed as the minimum objective function, and the square hollow section was used as the calculation variable. The design optimization involved size optimization, shape optimization, and topology

optimization [9]. Karaboga et al. proposed an improved artificial bee colony algorithm (ABC) based on chaos theory, which was applied to the truss structure example to verify the effectiveness of the improved algorithm for truss engineering structure optimization [10]. Kayabekir explored a new intelligent algorithm, the flower pollination algorithm (FPA), and discussed its practical application in civil engineering, mechanical engineering, electronic communication, and chemistry, In the civil engineering field, the FPA algorithm was applied to the 25-bar truss structure to optimize the structure, and the feasibility of the algorithm was verified [11]. Chen et al. proposed a hybrid particle swarm optimization (PSO) algorithm based on the improved Nelder–Mead algorithm (NMA). The improved NMA selected a part of the n-simplex subplane for optimization, using modal strain energy. The index (MSEBI) to locate the damage effectively improved the convergence speed and accuracy of the PSO algorithm and greatly improved the computational efficiency in the field of structural damage detection (SDD) [12]. Andrea Caponio et al. proposed a metaheuristic algorithm based on population distribution theory for the application of practical engineering structure optimization. The proposed algorithm can converge well when faced with nonlinear and indistinguishable optimization problems. The optimization process was stable and provides a new way of thinking [13].

In 2014, Chinese scholar Meng proposed the chicken swarm optimization (CSO) [14], which was a group-based stochastic optimization algorithm. The advantage was that the implementation was simple. The disadvantage was that it was easy to fall into the local optimal solution, converges slowly, and is low in precision. Scholars had done a lot of research on this optimization algorithm. In 2015, Fei et al. proposed an improved chicken swarm optimization (CSO), which analyzed the weighting factors in the reference particle swarm optimization algorithm. The adaptive weight was introduced in the process of updating the individual position of the chicken, which solved the problem that the optimal search solution was easy to skip due to the large search space in the early stage of the algorithm; the late search space was small, the convergence was slow, and the learning part of the individual with the cock was added. This allowed the chick to enjoy a comprehensive learning mechanism [15]. Yu et al. improved the chicken swarm optimization (CSO) in 2016 using a variety of hybrid methods. The reverse direction learning method was used to initialize the population individuals to improve the quality of the population, and the variation idea was introduced in the boundary processing to enhance the diversity of the population, improving the global optimization ability of the algorithm. Finally, the idea of simulated annealing was used to accept the inferior solution with a certain probability, which enhanced the ability of the algorithm to jump out of local optimum [16]. These improved algorithms effectively improved the optimization performance of the algorithm, and the results of its function optimization proved that the improved algorithm was effective. The existing research showed that the chicken swarm optimization (CSO) had been successfully applied to resource scheduling [17], engineering optimization design [14], cluster analysis, and optimization of classifier coefficients [18, 19]. In 2016, Hafez et al. proposed a feature selection system based on chicken swarm optimization (CSO), which was used to select features in a wrapper mode to search for feature space for the best combination of features, thus maximizing classification performance while minimizing the number of selected features [20]. Shayokh et al. used the chicken swarm optimization (CSO) to solve the wireless sensor network (WNS) node location problem [21]. In 2016, Roslina et al. proposed an improved chicken swarm optimization (CSO) for ANFIS performance, which can more accurately solve the ANFIS network training classification problem [22]. In 2017, Awais et al. combined the chicken swarm optimization (CSO) with energy optimization for home users, enabling home users to reduce power costs, power consumption, and peak-to-average ratio [23, 24]. Ahmed et al. improved the chicken swarm optimization (CSO) search ability by applying logistic and tend chaotic mapping to help the chicken swarm optimization (CSO) better explore search space [25]. These successful applications show that the chicken swarm optimization (CSO) has a good development and application prospects. However, in these application studies, the chicken swarm algorithm (CSO) is also imperfect. For example, the algorithm is not initialized, the chicken position update does not prevent the individual from being out of bounds, and the algorithm has no overall individual optimization.

Based on the above research, this paper improved the chicken swarm optimization and applied to truss structure optimization; the concept of combining chaos strategy and reverse learning strategy was introduced in the initialization to ensure the global search ability. And the inertia weighting factor and the learning factor were introduced into the chick position update process, so as to better combine the global and local search. Finally, the overall individual position of the algorithm was optimized by the differential evolution algorithm. The improved algorithm was tested by multipeak function and applied to the truss simulation experiment.

2. Chicken Swarm Optimization Algorithm

The traditional chicken swarm optimization mainly treats the optimization problem as the process of chickens' searching for food. The whole chicken swarm is divided into several chicken flocks, each of which has a cock, several hens, and several chicks. There is a competition between each chicken swarm, and the best cluster individuals are obtained through competitions. The process of simplifying is as follows:

(1) In each chicken swarm, there are many subchicken swarms, each of which included one cock, several hens, and several chicks.

(2) The chicken swarm divides several subchicken swarms and determines the fitness value of individuals on which cocks, hens, and chicks depend. Several individuals with the best fitness values can act as cocks. Each cock is a leader of a chicken swarm. The worst fitness can be used as a chicken, and the

rest can be hens. The hens randomly follow a cock, and the relationship between the hen and the chick is randomly formed.

(3) The dominance relationship, hierarchy, and mother-child relationship in the chicken swarm are unchanged; chickens regroup and update roles every G generation.

(4) The subchicken swarms look for food with the cock, the chicks look for food around the hens, and the individual has an advantage in finding food. The cocks, hens, and chicks in the chicken swarms perform different ways of optimizing.

Individuals in the chicken swarm move according to their own rules until they find the best position. Therefore, the individual position in the chicken swarm correspond to a solution to the optimization problem, and finding the best position is the optimal solution to the optimization problem. In the whole chicken swarm optimization, the number of individuals in all flocks is set to N, and the position of each chicken swarm individual is represented by $x_{i,j}(t)$, and its meaning indicate the position obtained in the t-th iteration of the i-th flock individual in the j-th dimension. Therefore, there are different positions for the three different types of chickens in the chicken swarm optimization; that is, the position update of the individual flocks is changed with different positions depending on the type of chicken. The cock has the best fitness value in each subgroup, and it can find and locate food in a wide range of spaces.

The position corresponding to the cock is updated as follows:

$$x_{i,j}(t+1) = x_{i,j}(t) * \left(1 + \text{Rand}\, n\left(0, \sigma^2\right)\right),$$
$$\sigma^2 = \begin{cases} 1, & f_i \le f_k, \\ \exp\left(\dfrac{f_k - f_i}{|f_i| + \varepsilon}\right), & f_i > f_k, \end{cases} \tag{1}$$

where $\text{Rand}\, n(0, \sigma^2)$ produces a mean of 0 and Gaussian distribution random number with standard deviation σ, ε is an extremely small number to prevent the denominator from being zero, f_i is the fitness value of individual i, and f_k is the fitness value of the individual k. Individual k is randomly selected from the rooster population and $k \neq i$.

The location of the hen is updated as follows:

$$\begin{aligned} x_{i,j}(t+1) &= x_{i,j}(t) + c_1 * \text{rand} * \left(x_{r1,j}(t) - x_{i,j}(t)\right) \\ &\quad + c_2 * \text{rand} * \left(x_{r2,j}(t) - x_{i,j}(t)\right), \\ c_1 &= \exp\left(\frac{(f_i - f_{r1})}{\text{abs}(f_1) + \varepsilon}\right), \\ c_1 &= \exp\left(f_{r2} - f_i\right), \end{aligned} \tag{2}$$

where rand is a random number between 0 and 1, $r1$ is the spouse cock of the i-th hen, $r2$ is any individual of all cocks and hens in the flock, and $r1 \neq r2$.

The location corresponding to the chick is updated as follows:

$$x_{i,j}(t+1) = x_{i,j}(t) + F * \left(x_{m,j}(t) - x_{i,j}(t)\right), \tag{3}$$

where m represents the hen corresponding to the i-th chick and F is the follow-up coefficient, which means that the chick follows the hen to find food.

3. Improved Chicken Swarm Optimization Algorithm

3.1. Initial Selection. As can be seen from above, the population described in the traditional chicken swarm optimization is not initialized, which brought disadvantages to the traditional algorithm, and there is no guarantee that the optimal solution of the algorithm can evenly distribute your distribution in the search space, which limits the efficiency of the algorithm and reduces the performance of the algorithm. Chaos strategy and reverse learning strategy were combined in the initialization of the algorithm [26–28]. Firstly, the chaotic strategy was used to make the state nonrepetitive and ergodic, and then the reverse learning strategy was used to reduce the blindness of the algorithm to expand the initial search range of the whole population; the global detection capability of the algorithm is ensured, so as not to fall into local optimum. The process was as follows:

(1) Select the chicken swarm optimization population $NP_1 = \{x_1(t), x_2(t), \ldots, x_N(t)\}$, and each individual in the population was $x_i(t) = (x_{i,1}(t), x_{i,2}(t), \ldots, x_{i,D}(t))$, $i \in N$.

(2) Generate chaotic sequences by model iterations and describe the logistic mapping of equation (4) $NP_2 = \{x_1(t)^*, x_2(t)^*, \ldots, x_N(t)^*\}$:

$$x_i(t)^* = \mu * x_i(t) * \left(1 - x_i(t)\right), \tag{4}$$

where the parameter $\mu = 4$.

(3) Find the inverse solution corresponding to NP_2 individual by equation (5) $\widetilde{x_i}(t) = (\widetilde{x_{i,1}}(t), \widetilde{x_{i,2}}(t), \ldots, \widetilde{x_{i,D}}(t))$; thus, the inverse population $NP_{\text{op}} = \{\widetilde{x_1}(t), \widetilde{x_2}(t), \ldots, \widetilde{x_N}(t)\}$ was obtained:

$$\widetilde{x_{i,D}}(t) = K * \left(x_i + y_i\right) - x_{i,D}(t), \tag{5}$$

where $x_{i,D}(t) \in [x_i, y_i]$, $[x_i, y_i]$ is the dynamic boundary of the search space and $K \in [0, 1]$ is a random number subject to uniform distribution.

(4) $NP_2 \cup NP_{\text{op}}$ and choose the best fitness for x_{best}, calculate $x_{\text{mean}} = (x_1 + x_2 + \cdots x_{2N})/2N$, and finally get the inverse optimal solution as follows:

$$x_{\text{opbest}} = \begin{cases} x_{\text{best}}, & f(x_{\text{best}}) < f(x_{\text{mean}}), \\ x_{\text{mean}}, & f(x_{\text{best}}) \ge f(x_{\text{mean}}). \end{cases} \tag{6}$$

By introducing chaos strategy and reverse learning strategy, the chicken swarm optimization could find the optimal solution in a larger search space and can guide the individual to evolve to the optimal solution, so that the overall convergence speed is improved.

3.2. Chick Location Update. In the traditional chicken swarm optimization, the position update of the chick is only related to the position of the hen, but not to the position of the cock with the best fitness in the algorithm [29, 30]. It can cause the algorithm to fall into local optimum to a certain extent, resulting in a lower overall efficiency. It is not difficult to see that the chicks in the chicken swarm optimization and the particles in the particle swarm optimization have similarities [31–33]. In order to improve the position update of the chicks, we referred to the particle swarm optimization to obtain the position of the particles locally and globally, so that the chicks can obtain the local position in the hen's side and the optimal position in the whole cock-guided group. In this paper, the inertia weight value ω and the learning factor φ in the particle swarm optimization algorithm were used to improve the chick position update. Equation (3) was improved as follows:

$$x_{i,j}(t+1) = \omega(t) * x_{i,j}(t) + \varphi_1 * \left(x_{m,j}(t) - x_{i,j}(t)\right) + \varphi_2\left(x_{r,j}(t) - x_{i,j}(t)\right), \tag{7}$$

where m is the individual in the subgroup and the individual corresponding to the hen; r is the individual of the cock corresponding to the chick in the subgroup; φ_1 and φ_2 are the two learning factors, respectively, indicating the degree of learning of the chick to the hen and the cock; and $\omega(t)$ is the weight value.

3.2.1. Setting the Weight Value. The setting of the weight value is related to whether the algorithm can better achieve local and global search, which is related to the diversity of the algorithm population in the later stage. This paper introduced a nonlinear inertia weight value $\omega(t)$, as follows:

$$\omega(t) = \text{floor}\left(\left(\omega_{\max} - \omega_{\min}\right) * \exp^3\left(-2 * \left(t/T_{\max}\right)\right) + \omega_{\min}\right), \tag{8}$$

where $\omega_{\max}$ is the maximum inertia weight, $\omega_{\min}$ is the minimum inertia weight, $T_{\max}$ is the maximum number of iterations, and $floor$ is the rounding function, in order to ensure that $\omega(t)$ is an integer.

After introducing the nonlinear weight value, it can ensure the global search of the individual chicken in the early stage of the algorithm, thereby improving the accuracy of the algorithm and strengthening the local search later.

3.2.2. Setting the Learning Factor. The learning factors φ_1 and φ_2, appearing in the chick position update, indicate the degree of learning of chicks to hens and cocks to some extent, just like the individual learning factors and social learning factors in the particle swarm optimization [33]. From the learning factor in the particle swarm optimization, it can be inferred that, in the initial stage of the search, the larger φ_1 makes the chicks search for a greater probability around the hen and more is to find the optimal solution globally; in the later stages of the search, the larger φ_2 makes the chicks search for a greater chance around the cocks, which is a local search near the optimal solution [34]. This paper introduced a nonlinear learning factor that allowed chicks to perform local and global searches. The formula was as follows:

$$\begin{aligned} \varphi_1 &= 1.3 + 1.2 * \cos\left(t\pi/T_{\max}\right), \\ \varphi_2 &= 2 - 1.2 * \cos\left(t\pi/T_{\max}\right). \end{aligned} \tag{9}$$

3.3. Select the Best Individual. In the selection of the optimal position of the chicken swarm optimization, a differential algorithm was used, which consists of three processes: variation, intersection, and selection [35, 36]. The main formula was as follows:

(1) *Variation Process.* Two bodies $x_{r1,j}(t)$ and $x_{r2,j}(t)$ of the same iteration number were randomly selected, and the mutation operation was performed according to the following equation:

$$v_{i,j}(t+1) = x_{i,j}(t) + F * \left(x_{r1,j}(t) - x_{r2,j}(t)\right), \tag{10}$$

where $v_{i,j}(t+1)$ is the mutated individual and F is a random factor of [0, 1], which controls the degree of expansion of the check score vector.

(2) *Cross Process.* The cross-probability factor P was introduced, and the cross operation was performed according to the following equation:

$$k_{i,j}(t) = \begin{cases} v_{i,j}(t+1), & P \in [0,1], \\ x_{i,j}(t), & \text{otherwise}. \end{cases} \tag{11}$$

(3) *Selection Process.* For comparing the fitness function values of two individuals, select individuals with large function values to perform mutation and crossover operations and compare the generated new individuals with the previous generation. If the former was larger than the latter, enter the next iteration; otherwise, remain unchanged.

3.4. Improved Chicken Swarm Optimization Steps and Flow Chart. The steps to improved chicken swarm optimization are described as follows:

Step 1: set relevant parameters of the algorithm, population size, cock, hen, chick scale factor, update iteration number, etc.

Step 2: initialize the population by combining chaos strategy and reverse learning strategy

Step 3: determine the fitness value of the individual, the ratio of cock, hen, and chicken, grouping, and various relationships; record the current global optimal fitness value, that is, the optimal individual position

Step 4: enter the iterative update to determine whether the update condition is met; the order of the flock, the mother-child relationship, and the partnership are updated; the position of the cock, hen, and chick is updated; and the boundary is processed

Step 5: select the optimal individual through three steps mutation, intersection, and selection

Step 6: when the number of iterations is less than the maximum number of iterations, go to step (4) to continue execution; otherwise, perform step (7)

Step 7: the individual who chooses the best position is the optimal solution

The flow chart of the improved chicken swarm optimization is shown in Figure 1.

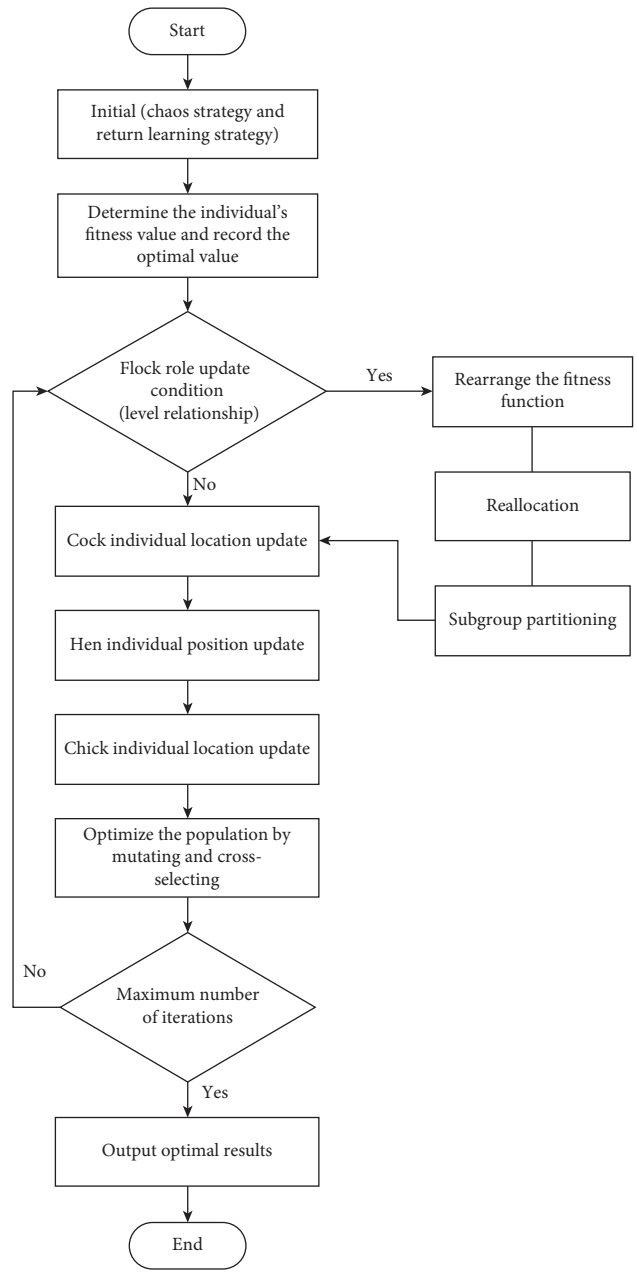


Figure 1: Flow chart of improved chicken swarm optimization.

4. Multipeak Function Tests

The experimental environment in this paper used 3.5 GHz CPU, 4 GB memory, 64 bit operating system, Windows 10 Professional, and the programming environment is MATLAB R2017b.

To verify the effectiveness of the improved chicken swarm optimization and to analyze the improved convergence and performance of the improved chicken swarm optimization, five well-known test functions, which are adopted from the IEEE CEC competitions, were applied in the experiments [37–40] and compared with the traditional chicken swarm optimization (CSO), bat algorithm (BA), and particle swarm optimization (PSO) [14, 41]. The setting parameters of the algorithm are shown in Table 1. The specific forms of the test functions are shown in Table 2.

Among them, the dimension of the test function was set to 30, and the number of iterations was 200. The calculation results are shown in Table 3, and Figures 2–6 are iterative diagrams of the test function.

It can be seen from Table 3 and the iterative graph of the test function that the improved chicken swarm optimization is more powerful than three other optimization algorithms, regardless of the convergence accuracy of the algorithm and the ability to find the optimal value or the robustness of the algorithm; and it is better than the traditional chicken swarm optimization in the optimization accuracy and convergence speed. Thus, after the multipeak function test, it shows that improved chicken swarm optimization has great advantages.

5. Engineering Applications

In industrial production, the cross-sectional area of the truss 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 structure, minimum displacement of specified nodes, and maximum natural frequency. In this paper, the optimal cross-sectional area of the member was found under the condition of the stress constraint of the member, so that the truss mass was minimized and the node displacement was minimized. The n-bar truss structure system was studied. The basic parameters of the system (including elastic modulus, material density, maximum allowable stress, and maximum allowable displacement) were known. Under the given load conditions, the optimal cross-sectional area of the N-bar truss was found to minimize the mass.

Four typical truss optimization examples were chosen to demonstrate the efficiency and reliability of the improved chicken swarm optimization for solving the shape and size of trusses with multiple frequency constraints.

5.1. Optimization Design of Cross Section of the 25 Bar Space Truss Structure. Figure 7 shows the 25 bar space truss structure model [42]. The known material density is

Table 1: The related parameter value.

Algorithm	Parameter
PSO	$c1 = c2 = 1.49445$, $w = 0.729$
BA	$\alpha = \gamma = 0.9$, $f_{\min} = 0$, $f_{\max} = 2$, $A_0 \in [0, 2]$, $r_0 \in [0, 1]$
CSO	$RN = 0.2 * N$, $HN = 0.6 * N$, $CN = N - RN - HN$, $MN = 0.1 * N$, $G = 10$, $FL \in [0.4, 1]$
Improved chicken swarm optimization	$RN = 0.2 * N$, $HN = 0.6 * N$, $CN = N - RN - HN, MN = 0.1 * N$, $G = 10$, $FL \in [0.4, 1]$, $\omega_{\max} = 0.9$, $\omega_{\min} = 0.4$, $K = 200$

Table 2: Test function.

Function form	Bounds	Optimum
$f_1(x) = \sum_{i=1}^{N} (\sum_{j=1}^{i} x_j)^2$	$[-100, 100]$	0
$f_2(x) = \sum_{i=1}^{N} \lvert x\rvert + \prod_{i=1}^{N} \lvert x\rvert$	$[-10, 10]$	0
$f_3(x) = \sum_{i=1}^{N} [x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-6, 6]$	0
$f_4(x) = (1/4000)\sum_{i=1}^{N} x_i^2 - \prod_{i=1}^{N} \cos(x^i/\sqrt{i}) + 1$	$[-600, 600]$	0
$f_5(x) = \sum_{i=1}^{N-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$	$[-2, 2]$	0

Table 3: Optimal calculation results.

Function	Algorithm	Best	Mean	Worst
$f_1(x)$	PSO	0	0	0
	BA	1.8674	2.9419	4.1871
	CSO	0	0	0
	Improved chicken swarm optimization	0	0	0
$f_2(x)$	PSO	5.8349	8.74784	10.5953
	BA	0.8076	$8.9964e+006$	$3.5767e+008$
	CSO	$1.1279e-07$	$5.53881e-07$	$1.0343e-06$
	Improved chicken swarm optimization	0	0	0
$f_3(x)$	PSO	86.4687	$1.14428e+02$	$1.46842e+02$
	BA	88.44729	121.99296	167.60654
	CSO	$3.3739e-08$	$5.5749e-02$	$1.83e-02$
	Improved chicken swarm optimization	0	0	0
$f_4(x)$	PSO	0.8973	1.06072	1.1596
	BA	0.004	2.82906	15.42094
	CSO	$1.6567e-07$	$1.092612e-02$	$2.067e-01$
	Improved chicken swarm optimization	0	0	0
$f_5(x)$	PSO	$7.1502e-10$	$2.78472e-06$	$4.3144e-06$
	BA	23.8966	101.8847	462.9705
	CSO	$2.4298e-10$	$3.15851e-06$	$1.86591e-06$
	Improved chicken swarm optimization	0	0	0

$\rho = 2786\,\mathrm{kg}/m^3$, elastic modulus $E = 68974\,\mathrm{MPa}$, stress constraint $[-275.8, 275.8]$ MPa, and $L = 635\,\mathrm{mm}$; the maximum vertical displacement of nodes 1 and 2 does not exceed $d_{\max} = 8.899\,\mathrm{mm}$ and $d_{\max} = 8.899\,\mathrm{mm}$. According to the symmetry, 25 rods are divided into 8 groups, and the design variable is 8. The control parameters of the algorithm are set as follows: the maximum number of iterations is 500; the search space dimension is 10. We compared improved chicken swarm optimization with improved particle swarm optimization (IPSO), improved fruit fly optimization algorithm (IFFOA), and improved genetic algorithm (IGA) [43–45]. The node loads are shown in Table 4, and the bars are grouped in Table 5.

Results after optimization are shown in Table 6.

It can be concluded from Table 6 that, under the same constraints, the improved chicken swarm optimization algorithm was used to optimize the 25 bar truss structure. The optimized total mass of the structure is 211.48 kg, which is lighter than the improved particle swarm optimization algorithm, and the quality is reduced to $(219.38 - 211.48)/211.48 = 3.74\%$; compared with the improved genetic algorithm, the quality is reduced to $(222.49 - 211.48)/211.48 = 5.20\%$; the quality of the results obtained with the improved fruit fly optimization algorithm is reduced to $(216.11 - 211.48)/211.48 = 2.19\%$; and the optimization results are improved. The four algorithm optimization iteration curves are shown in Figure 8. As can be seen from Figure 8, the improved chicken swarm optimization can

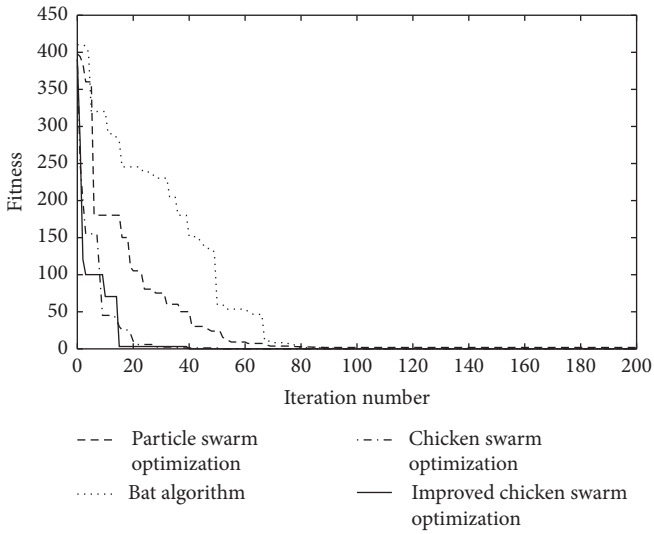


Figure 2: The experimental result of $f_1(x)$ with 30 dimensions.

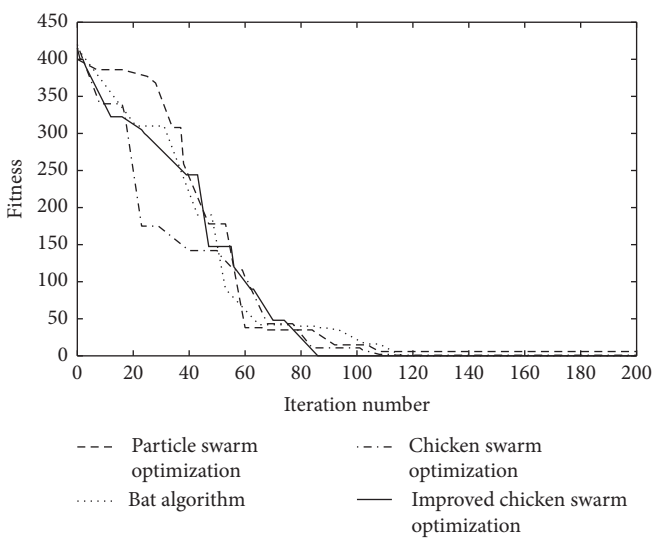


Figure 3: The experimental result of $f_2(x)$ with 30 dimensions.

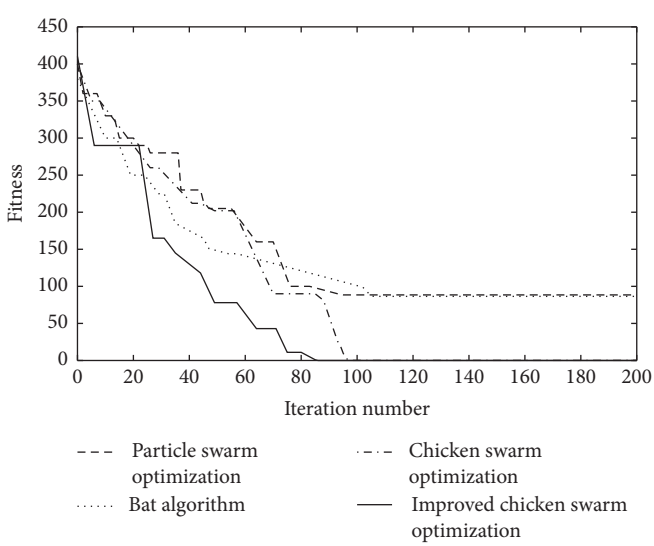


Figure 4: The experimental result of $f_3(x)$ with 30 dimensions.

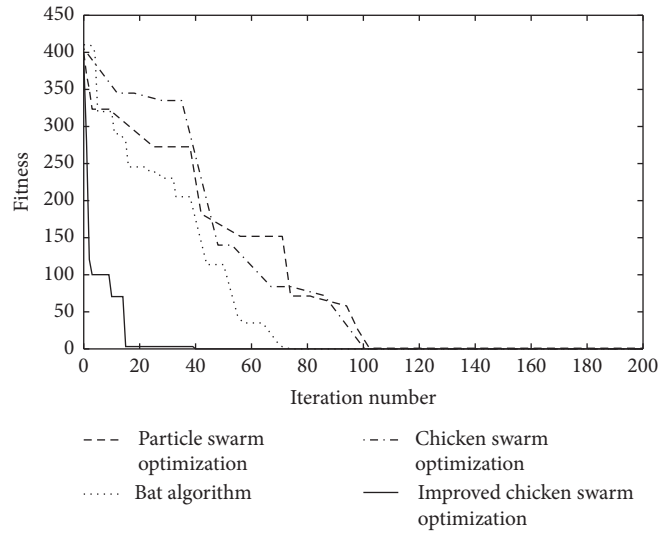


Figure 5: The experimental result of $f_4(x)$ with 30 dimensions.

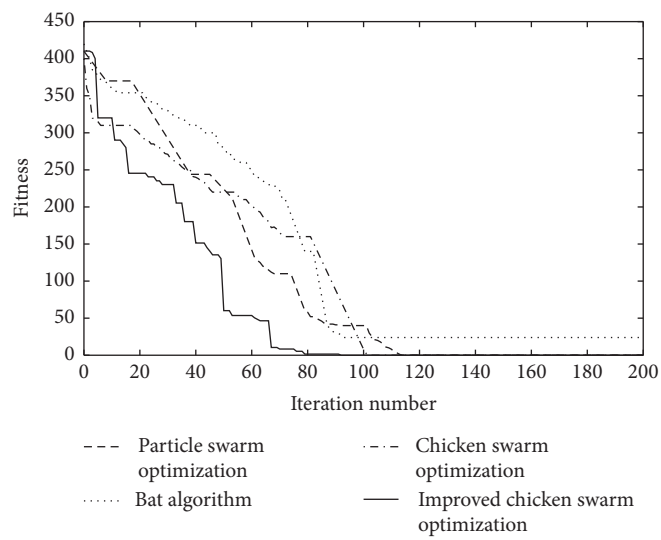


Figure 6: The experimental result of $f_5(x)$ with 30 dimensions.

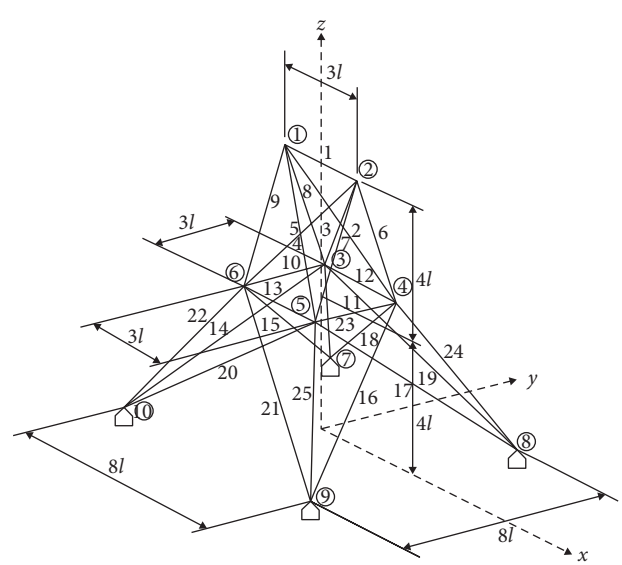


Figure 7: Schematic diagram of the 25 bar space truss structure.

Table 4: 25 bar space truss load.

Node number	F_x (kN)	F_y (kN)	F_z (kN)
1	4.445	44.45	−22.23
2	0	44.45	−22.23
3	2.223	0	0
6	2.667	0	0

Table 5: 25 bar space truss classification.

Group number	Rod number
1	A_1
2	$A_2 \sim A_5$
3	$A_6 \sim A_9$
4	$A_{10} \sim A_{11}$
5	$A_{12} \sim A_{13}$
6	$A_{14} \sim A_{17}$
7	$A_{18} \sim A_{21}$
8	$A_{22} \sim A_{25}$

Table 6: Comparison of optimization results for 25 bar space truss.

Numbering	Rod cross-sectional area (mm^2)			
	IPSO [43]	IFFOA [44]	IGA [45]	ICSO
1	79.33	62.52	65.79	75.55
2	186.54	191.55	234.57	197.55
3	2145.95	2191.54	2229.55	2178.55
4	326.57	66.52	64.32	286.16
5	848.12	1086.77	1220.62	984.14
6	375.65	615.16	500.25	410.69
7	454.32	398.09	91.15	420.16
8	2275.64	2183.55	2567.85	2194.41
Weight (kg)	219.38	216.11	222.49	211.48

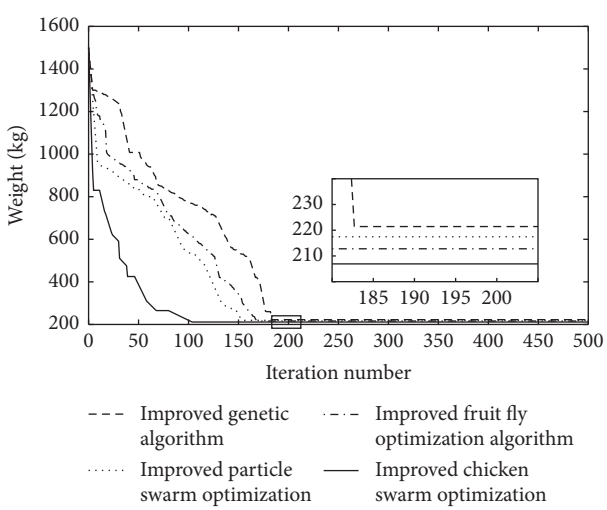


Figure 8: Four algorithms for finding the iterative curve of 25 bars.

search for the global optimal solution. It has higher convergence precision and convergence speed, and the effect is obvious.

5.2. Optimization Design of Cross Section of the 52 Bar Plane Truss Structure. As shown in Figure 9, the space 52 truss structure model is established, and the bars are divided into 12 groups according to the force of the members [44]. The specific grouping situation is shown in Table 7. We compared improved chicken swarm optimization with improved particle swarm optimization (IPSO), improved fruit fly optimization algorithm (IFFOA), and improved genetic

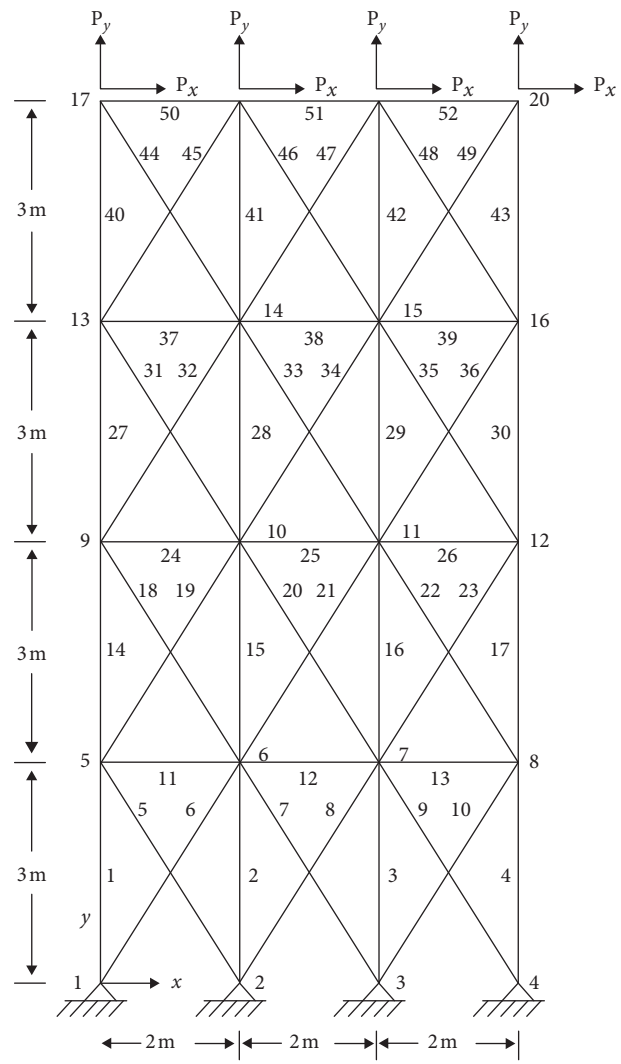


Figure 9: Schematic diagram of the 52 bar plane truss structure.

Table 7: 52 bar plane truss structure grouping.

Group number	Rod number
A_1	1, 2, 3, 4
A_2	5, 6, 7, 8, 9, 10
A_3	11, 12, 13
A_4	14, 15, 16, 17
A_5	18, 19, 20, 21, 22, 23
A_6	24, 25, 26
A_7	27, 28, 29, 30
A_8	31, 32, 33, 34, 35, 36
A_9	37, 38, 39
A_{10}	40, 41, 42, 43
A_{11}	44, 45, 46, 47, 48, 49
A_{12}	50, 51, 52

Table 8: Comparison of optimization results of the 52 bar plane truss structure.

Numbering	Rod cross-sectional area (mm^2)			
	IPSO [43]	IFFOA [44]	IGA [45]	ICSO
1	4656.06	4648.06	4652.15	4650.55
2	1164.29	1141.29	1165.15	1160.29
3	516.45	454.19	360.23	363.26
4	3323.22	3333.22	3306.32	3303.22
5	948.00	920.00	943.14	935.26
6	484.19	490.19	486.15	494.19
7	2270.32	2218.71	2205.16	2210.39
8	1028.39	1028.39	1010.18	1005.59
9	2260.32	474.19	382.12	388.386
10	1555.48	1253.87	1286.17	1280.59
11	1065.16	1181.29	1160.13	1154.66
12	501.45	490.19	786.15	792.256
Weight (kg)	1938.36	1900.37	1912.61	1895.45

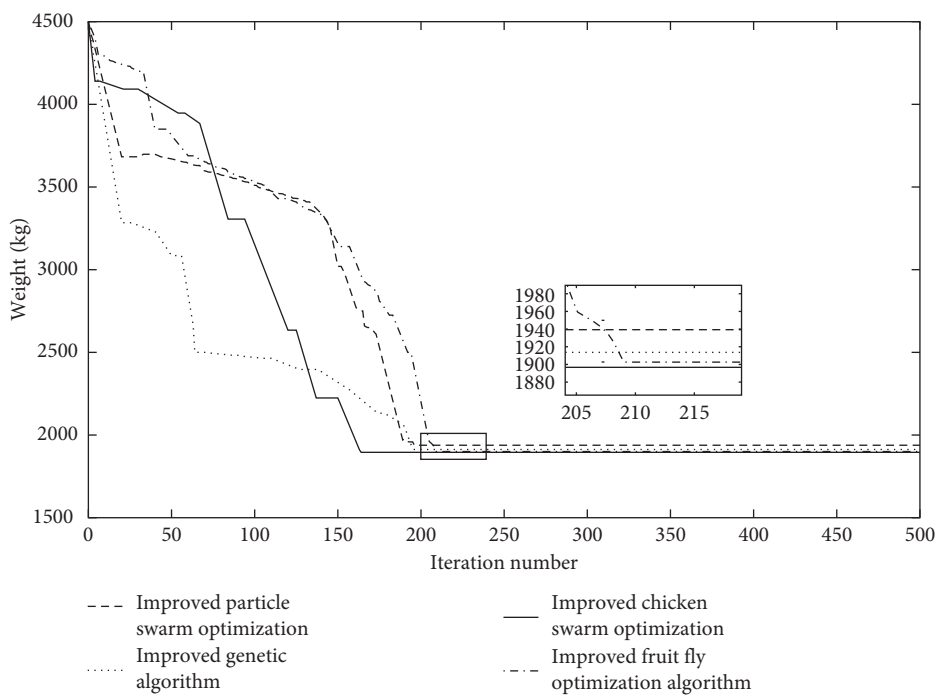


Figure 10: Four algorithms for finding the iterative curve of 52 bars.

algorithm (IGA) [43–45]. The optimization results of each algorithm are shown in Table 8. The structural members are all made of the same material; material density is $\rho = 7860.0\,\text{kg/m}^3$, the elastic modulus $E = 2.07 \times 10^5$ MPa, the lateral load $P_x = 100$ kN, and the vertical load $P_y = 200$ kN. The allowable stress of each member in the structure is $\pm$ 180 MPa.

It can be concluded from Table 8 that, under the same constraints, the improved chicken swarm optimization was used to optimize the 52-bar truss structure, and the total mass of the optimized structure is 1895.45 kg. Compared with the improved particle swarm algorithm, the quality is reduced to $(1938.36 - 1895.45)/1895.45 = 2.26\%$; compared with the improved genetic algorithm, the quality is reduced to $(1912.61 - 1895.45)/1895.45 = 0.91\%$; compared with the improved fruit fly optimization algorithm, the quality is reduced to $(1900.37 - 1895.45)/1895.45 = 0.26\%$; and the optimization results have been improved very well. The four algorithm optimization iteration curves are shown in Figure 10. As can be seen from Figure 10, the improved chicken swarm optimization can search for the global optimal solution. Compared with the other three optimization algorithms, it has a higher convergence precision and convergence speed, and the effect is obvious.

5.3. Optimization Design of Cross Section of the 72 Bar Space Truss Structure. The 72 bar space truss structure is shown in

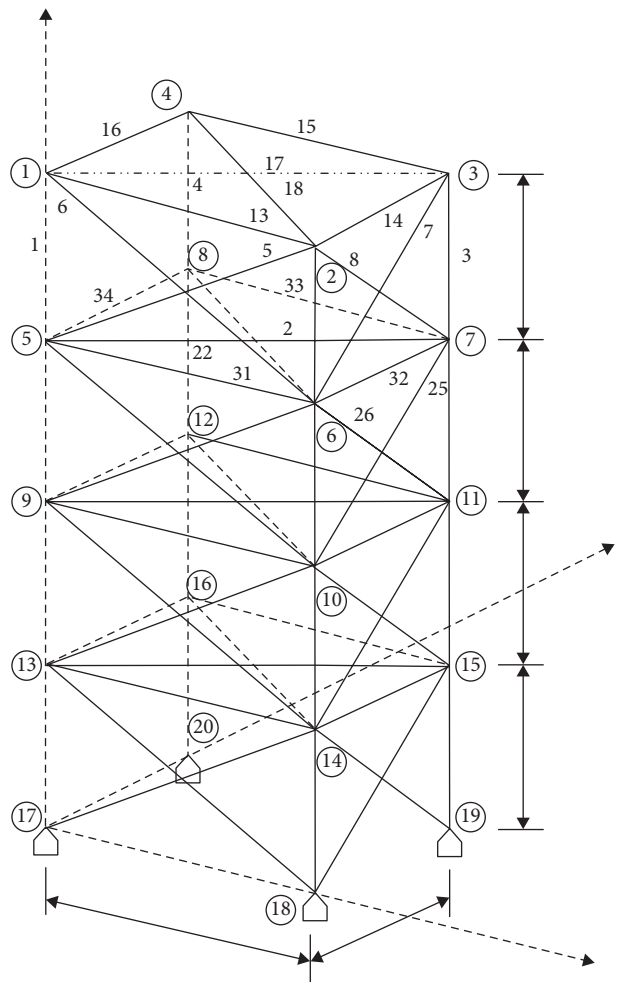


Figure 11: Schematic diagram of the 72 bar space truss structure.

Table 9: 72 bar space truss load.

Node	Working condition 1			Working condition 2		
	F_x (kN)	F_y (kN)	F_z (kN)	F_x (kN)	F_y (kN)	F_z (kN)
1	22250	22250	−22250	0	0	−22250
2				0	0	−22250
3				0	0	−22250
4				0	0	−22250

Figure 11 [46]. The structure considers two load conditions. The specific working conditions are shown in Table 9. According to the force of the rod, the 72 rods in the structure are divided into 16 groups. The specific grouping of the rods is shown in Table 10. The rods are made of the same material, the material density is $\rho = 2\,678\ \mathrm{kg/m^3}$, the elastic modulus is $E = 68950$ MPa, the maximum displacement variation range of each joint of the rods in each direction is ± 6.35 mm, the allowable stress range is [−172.375, 172.375], we compared the results with the improved particle swarm optimization algorithm (IPSO), the improved fruit fly optimization algorithm (IFFOA), and the improved genetic algorithm (IGA) [43–45], and the optimized results are shown in Table 11.

Table 10: 72 bar space truss classification.

Group number	Rod number
A_1	1, 2, 3, 4
A_2	5, 6, 7, 8, 9, 10, 11, 12
A_3	13, 14, 15, 16
A_4	17, 18
A_5	19, 20, 21, 22
A_6	23, 24, 25, 26, 27, 28, 29, 30
A_7	31, 32, 33, 34
A_8	35, 36
A_9	37, 38, 39, 40
A_{10}	41, 42, 43, 44, 45, 46, 47, 48
A_{11}	49, 50, 51, 52
A_{12}	53, 54
A_{13}	55, 56, 57, 58
A_{14}	59, 60, 61, 62, 63, 64, 65, 66
A_{15}	67, 68, 69, 70
A_{16}	71, 72

It can be concluded from Table 11 that, under the same constraints, the improved chicken swarm optimization was used to optimize the design of the 72 bar truss structure, and the total mass of the optimized structure is 170.24 kg.

Table 11: Comparison of optimization results for 72 bar space truss.

Numbering	Rod cross-sectional area (mm^2)			
	IPSO [43]	IFFOA [44]	IGA [45]	ICSO
1	185.74	161.26	101.34	95.34
2	351.28	370.97	345.54	325.54
3	200.21	279.27	264.26	244.26
4	300.69	381.99	367.29	357.29
5	325.93	156.51	326.91	318.91
6	350.70	330.46	335.48	326.48
7	58.52	60.52	64.52	62.53
8	58.52	60.52	64.52	62.53
9	820.23	713.98	825.87	810.87
10	318.26	373.81	332.13	321.13
11	58.52	60.52	64.52	62.52
12	58.52	60.52	64.52	62.52
13	1178.00	1330.02	1224.07	1204.07
14	310.44	324.88	330.77	335.77
15	58.53	60.52	64.52	62.54
16	58.53	60.52	64.52	62.54
Weight (kg)	182.43	172.37	191.56	170.24

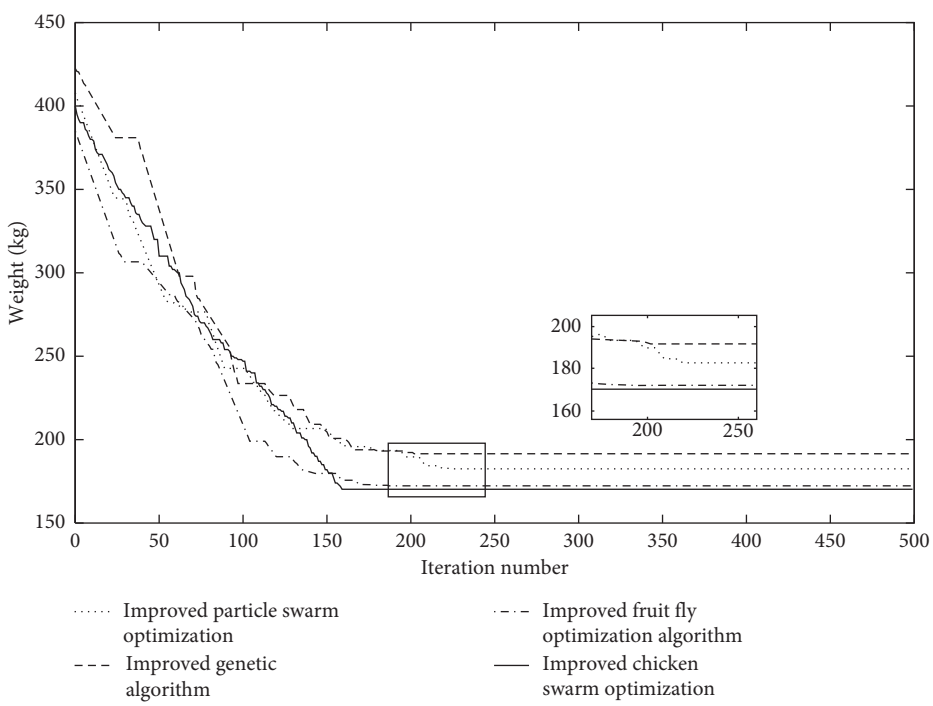


Figure 12: Four algorithms for finding the iterative curve of 72 bars.

Compared with the improved particle swarm algorithm, the quality is reduced to $(182.43 - 170.24)/170.24 = 7.16\%$; compared with the improved genetic algorithm, the quality is reduced to $(191.56 - 170.24)/170.24 = 12.52\%$; and compared with the improved fruit fly optimization algorithm, the results are optimized $(172.37 - 170.24)/170.24 = 1.25\%$. The optimization results have been improved very well. The four algorithm optimization iteration curves are shown in Figure 12. It can be seen from Figure 12 that the improved chicken swarm optimization can search for the global optimal solution. Compared with the improved genetic algorithm (IGA), the improved particle swarm optimization algorithm (IPSO), and the improved fruit fly optimization algorithm (IFFOA), it has higher convergence precision and convergence speed, and the effect is obvious.

5.4. Optimization Design of Cross Section of the 200 Bar Plane Truss Structure. The structure of the 200 bar truss is shown in Figure 13 [47]. The grouping of the structural members is shown in Table 12. The specific optimization results are shown in Table 13. The modulus of elasticity is

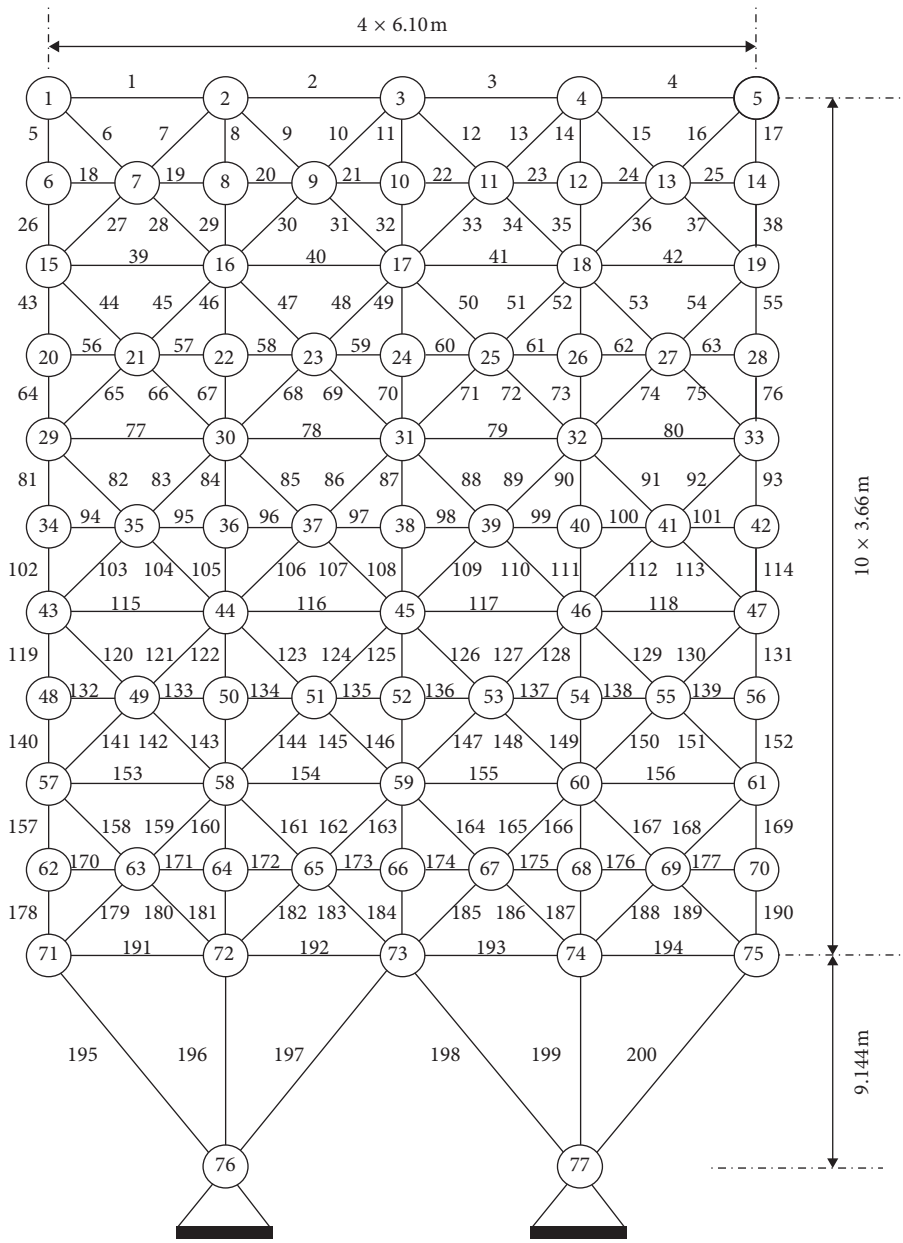


Figure 13: Schematic diagram of the 200 bar plane truss structure.

$E = 2.1 \times 10^{11}\,\mathrm{N/m^2}$, and the density of the rod material is $\rho = 7860\,\mathrm{kg/m^3}$. The minimum allowable cross-sectional area of the truss member is set to $0.1\,\mathrm{cm^2}$. The structure free point is given a nonstructural mass of 50 kg. The design variable has 29 dimensions and is a relatively complex structural optimization problem. We compared the improved chicken swarm optimization with the results of the improved gravitational search algorithm (IGSA), improved genetic algorithm (IGA), and improved fruit fly optimization algorithm (IFFOA) [44, 47, 48].

It can be concluded from Table 13 that, under the same constraints, the improved chicken swarm optimization was used to optimize the 200 bar truss structure, and the optimized total mass of the structure is 2147.92 kg. Compared with the improved gravitational search algorithm, the quality is reduced to $(2155.64 - 2147.92)/2147.92 = 3.59\%$. Compared with the improved genetic algorithm, the quality is reduced to $(2296.15 - 2147.92)/2147.92 = 6.90\%$; compared with the improved fruit fly optimization algorithm, the optimization results are optimized to $(2156.73 - 2147.92)/2147.92 = 4.10\%$, and the optimization results are improved.

It can be seen from Figure 14 that the improved chicken swarm optimization can find the global optimal solution after iteration for about 170 times. The gravitational search algorithm is iterated about 210 times, and the genetic algorithm is iterated about 220 times to find the global optimal solution, and the improved fruit fly

Table 12: 200 bar plane truss structure grouping.

Group number	Rod number
A_1	1, 2, 3, 4
A_2	5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17
A_3	19, 20, 21, 22, 23, 24
A_4	18, 25, 56, 63, 94, 101, 132, 139, 170, 177
A_5	26, 29, 32, 35, 38
A_6	6, 7, 9, 10, 12, 13, 15, 16, 27, 28, 30, 31, 33, 34
A_7	39, 40, 41, 42
A_8	43, 46, 49, 52, 55
A_9	57, 58, 59, 60, 61, 62
A_{10}	64, 67, 70, 73, 76
A_{11}	44, 45, 47, 48, 50, 51, 53, 54, 65, 66, 68, 69, 71, 72, 74, 75
A_{12}	77, 78, 79, 80
A_{13}	81, 84, 87, 90, 93
A_{14}	95, 96, 97, 98, 99, 100
A_{15}	102, 105, 108, 111, 114
A_{16}	82, 83, 85, 86, 88, 89, 91, 92, 103, 104, 106, 107, 109, 110, 112, 113
A_{17}	115, 116, 117, 118
A_{18}	119, 122, 125, 128, 131
A_{19}	133, 134, 135, 136, 137, 138
A_{20}	140, 143, 146, 149, 152
A_{21}	120, 121, 123, 124, 126, 127, 129, 130, 141, 142, 144, 145, 147, 148, 150, 151
A_{22}	153, 154, 155, 156
A_{23}	157, 160, 163, 166, 169
A_{24}	171, 172, 173, 174, 175, 176
A_{25}	178, 181, 184, 187, 190
A_{26}	158, 159, 161, 162, 164, 165, 167, 168, 179, 180, 182, 183, 185, 186, 188, 189
A_{27}	191, 192, 193, 194
A_{28}	195, 197, 198, 200
A_{29}	196, 199

Table 13: Comparison of optimization results of the 200 bar plane truss structure.

Numbering	Rod cross-sectional area (cm^2)			
	IGSA [47]	IGA [48]	IFFOA [44]	ICSO
1	0.239	0.293	0.289	0.265
2	0.466	0.556	0.488	0.445
3	0.120	0.295	0.100	0.100
4	0.120	0.190	0.100	0.100
5	0.479	0.834	0.499	0.484
6	0.864	0.645	0.804	0.845
7	0.115	0.177	0.103	0.145
8	1.353	1.479	1.377	1.371
9	0.112	0.449	0.100	0.112
10	1.544	1.458	1.698	1.682
11	1.141	1.123	1.156	1.142
12	0.121	0.273	0.131	0.135
13	3.032	1.865	3.261	3.021
14	0.114	0.117	0.101	0.102
15	3.254	3.555	3.216	3.264
16	1.646	1.568	1.621	1.625
17	0.227	0.265	0.209	0.148
18	5.152	5.165	5.020	4.164
19	0.124	0.659	0.133	0.154
20	5.485	4.987	5.453	5.684
21	2.907	1.843	2.113	2.123

Table 13: Continued.

Numbering	Rod cross-sectional area (cm^2)			
	IGSA [47]	IGA [48]	IFFOA [44]	ICSO
22	0.659	1.789	0.723	0.712
23	7.658	8.175	7.724	7.154
24	0.144	0.325	0.182	0.189
25	8.052	10.597	7.971	7.894
26	2.788	2.986	2.996	2.963
27	10.477	10.598	10.206	10.459
28	21.325	20.559	20.669	20.164
29	10.511	18.648	11.555	11.459
Weight (kg)	2155.64	2296.15	2156.73	2147.92

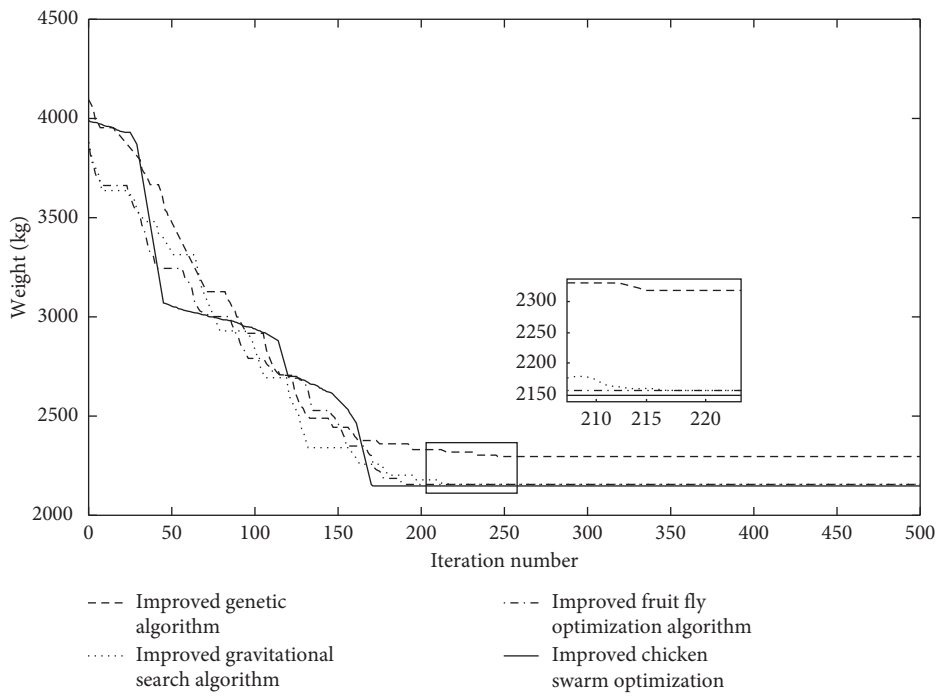


Figure 14: Four algorithms for finding the iterative curve of 200 bars.

optimization algorithm is iterated about 180 times to find the global optimal solution. For the quality problem of the optimal global solution, the improved chicken swarm optimization is better than other three optimization algorithms. It is proved that, in the process of optimizing in the face of complex optimization problems, the improved chicken swarm optimization is stable and not easy to fall into the local optimal solution. It can be seen that the improved chicken swarm optimization is effective.

6. Conclusion

Structure optimization is more and more important in civil engineering. In order to find more effective optimization method, the chicken swarm optimization was introduced, For the NP hard problems, the chicken swarm optimization performs better than common algorithms. It has the shortcoming of the premature. So, we improved the algorithm and applied to truss structure optimization; the concept of combining chaos strategy and reverse learning strategy was introduced in the initialization to ensure the global search ability. And the inertia weighting factor and the learning factor were introduced into the chick position update process, so as to better combine the global and local search. Finally, the overall individual position of the algorithm was optimized by the differential evolution algorithm. The multipeak function was used to prove its validity. Finally, the improved chicken swarm optimization was applied to the truss structure optimization design. The mathematical model of the truss section optimization was given. The example verification shows that the improved chicken swarm optimization has a faster convergence speed which makes the results more optimal.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

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] L. A. Schmit and B. Farshi, "Some approximation concepts for structural synthesis," *AIAA Journal*, vol. 12, no. 5, pp. 692–699, 1973.

[2] V. J. Savsani, G. G. Tejani, and V. K. Patel, "Truss topology optimization with static and dynamic constraints using modified subpopulation teaching-learning-based optimization," *Engineering Optimization*, vol. 48, no. 11, p. 11, 2016.

[3] A. Shakya, P. Nanakorn, and W. Petprakob, "A ground-structure-based representation with an element-removal algorithm for truss topology optimization," *Structural and Multidisciplinary Optimization*, vol. 58, no. 2, pp. 657–675, 2018.

[4] A. Asadpoure, J. K. Guest, and L. Valdevit, "Incorporating fabrication cost into topology optimization of discrete structures and lattices," *Structural and Multidisciplinary Optimization*, vol. 51, no. 2, pp. 385–396, 2015.

[5] S. Bobby, A. Suksuwan, S. M. J. Spence, and A. Kareem, "Reliability-based topology optimization of uncertain building systems subject to stochastic excitation," *Structural Safety*, vol. 66, pp. 1–16, 2017.

[6] Z. Zhang, K. Wang, L. Zhu, and Y. Wang, "A Pareto improved artificial fish swarm algorithm for solving a multi-objective fuzzy disassembly line balancing problem," *Expert Systems with Applications*, vol. 86, pp. 165–176, 2017.

[7] J.-Y. Park and S.-Y. Han, "Topology optimization for nonlinear structural problems based on artificial bee colony algorithm," *International Journal of Precision Engineering and Manufacturing*, vol. 16, no. 1, pp. 91–97, 2015.

[8] S. L. Tilahun and J. M. T. Ngnotchouye, "Firefly algorithm for discrete optimization problems: a survey," *KSCE Journal of Civil Engineering*, vol. 21, no. 2, pp. 535–545, 2017.

[9] A. Fiore, G. C. Marano, R. Greco, and E. Mastromarino, "Structural optimization of hollow-section steel trusses by differential evolution algorithm," *International Journal of Steel Structures*, vol. 16, no. 2, pp. 411–423, 2016.

[10] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: artificial bee colony (ABC) algorithm and applications," *Artificial Intelligence Review*, vol. 42, no. 1, pp. 21–57, 2014.

[11] A. E. Kayabekir, G. Bekdaş, S. M. Nigdeli, and X.-S. Yang, "A comprehensive review of the flower pollination algorithm for solving engineering problems," *Nature-Inspired Algorithms and Applied Optimization*, vol. 40, no. 7, pp. 171–188, 2018.

[12] Z. Chen and L. Yu, "An improved PSO-NM algorithm for structural damage detection," *Advances in Swarm and Computational Intelligence*, pp. 124–132, 2015.

[13] A. Cincin, B. Ozben, and O. Erdogan, "Memetic algorithms in engineering and design," *Handbook of Memetic Algorithms*, pp. 241–260, Springer, Berlin, Heidelberg, 2012.

[14] X. Meng, Y. Liu, X. Gao, and H. Zhang, "A new bio-inspired algorithm: chicken swarm optimization," *Lecture Notes in Computer Science*, pp. 86–94, 2014.

[15] D. Wu, K. Fei, W. Gao, Y. Shen, and Z. Ji, "Improved chicken swarm optimization," in *Proceedings of the Paper presented at the IEEE International Conference on Cyber Technology in Automation*, Shenyang, China, June 2015.

[16] W. Yu, X. Qu, and Y. Bo, "Improved chicken swarm optimization method for reentry trajectory optimization," *Mathematical Problems in Engineering*, vol. 2018, Article ID 8135274, 13 pages, 2018.

[17] S. Torabi and F. Safi-Esfahani, "A dynamic task scheduling framework based on chicken swarm and improved raven roosting optimization methods in cloud computing," *Journal of Supercomputing*, vol. 74, no. 6, pp. 2581–2626, 2018.

[18] H. Hu, J. Li, and J. Huang, "Economic operation optimization of micro-grid based on chicken swarm optimization algorithm," *High Voltage Apparatus*, vol. 53, no. 1, pp. 119–125, 2017.

[19] N. Irsalinda, I. T. R. Yanto, H. Chiroma, and T. Herawan, "A framework of clustering based on chicken swarm optimization," *Advances in Intelligent Systems and Computing*, vol. 549, pp. 336–343, 2016.

[20] A. I. Hafez, H. M. Zawbaa, E. Emary, H. A. Mahmoud, and A. E. Hassanien, "An innovative approach for feature selection based on chicken swarm optimization," in *Proceedings of the 2015 7th International Conference of Soft Computing and Pattern Recognition (SoCPaR)*, Japan, 2015.

[21] M. A. Shayokh and S. Y. Shin, "Bio inspired distributed WSN localization based on chicken swarm optimization," *Wireless Personal Communications*, vol. 97, no. 4, pp. 5691–5706, 2017.

[22] Roslina, M. Zarlis, I. T. R. Yanto, and D. Hartama, "A framework of training ANFIS using chicken swarm optimization for solving classification problems," in *Proceedings of the 2016 International Conference on Informatics and Computing (ICIC)*, Mataram, Indonesia, October 2016.

[23] M. Awais, Z. U. Abadeen, T. Bilal, Z. Faiz, M. Junaid, and N. Javaid, "Home Energy Management Using Enhanced Differential Evolution and Chicken Swarm Optimization Techniques," *Advances in Intelligent Networking and Collaborative Systems*, vol. 8, pp. 468–478, 2017.

[24] Z. Faiz, T. Bilal, M. Awais, Pamir, S. Gull, and N. Javaid, "Demand side management using chicken swarm optimization," *Advances in Intelligent Networking and Collaborative Systems*, vol. 8, pp. 155–165, 2017.

[25] K. Ahmed, A. E. Hassanien, and S. Bhattacharyya, "A novel chaotic chicken swarm optimization algorithm for feature selection," in *Paper presented at the Third International Conference on Research in Computational Intelligence & Communication Networks*, Kolkata, India, 2017.

[26] M. A. Ahandani and H. Alavi-Rad, "Opposition-based learning in shuffled frog leaping: an application for parameter identification," *Information Sciences*, vol. 291, pp. 19–42, 2015.

[27] S. Y. Park and J. J. Lee, "Stochastic opposition-based learning using a beta distribution in differential evolution," *IEEE Transactions on Cybernetics*, vol. 46, no. 10, pp. 2184–2194, 2016.

[28] B. Bentouati, S. Chettih, R. Djekidel, and R. A.-A. El-Sehiemy, "An efficient chaotic cuckoo search framework for solving non-convex optimal power flow problem," *International Journal of Engineering Research in Africa*, vol. 33, pp. 84–99, 2017.

[29] M. Abdollahi, A. Bouyer, and D. Abdollahi, "Improved cuckoo optimization algorithm for solving systems of nonlinear equations," *Journal of Supercomputing*, vol. 72, no. 3, pp. 1–24, 2016.

[30] D. Chitara, K. R. Niazi, A. Swarnkar, and N. Gupta, "Cuckoo search optimization algorithm for designing of a

multimachine power system stabilizer," *IEEE Transactions on Industry Applications*, vol. 54, no. 4, pp. 3056–3065, 2018.

[31] M. R. Bonyadi and Z. Michalewicz, "Impacts of coefficients on movement patterns in the particle swarm optimization algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 3, pp. 378–390, 2017.

[32] Y.-p. Chen and P. Jiang, "Analysis of particle interaction in particle swarm optimization," *Theoretical Computer Science*, vol. 411, no. 21, pp. 2101–2115, 2010.

[33] B. Nasiri, M. R. Meybodi, and M. M. Ebadzadeh, "History-driven particle swarm optimization in dynamic and uncertain environments," *Neurocomputing*, vol. 172, no. 3, pp. 356–370, 2016.

[34] M. A. Shayokh and S. Y. Shin, "Bio inspired distributed WSN localization based on chicken swarm optimization," *Wireless Personal Communications*, vol. 97, no. 4, pp. 5691–5706, 2017.

[35] A. Orlandi, "Differential evolutionary multiple-objective sequential optimization of a power delivery network," *IEEE Transactions on Electromagnetic Compatibility*, vol. 99, pp. 1–7, 2018.

[36] A. W. Mohamed, "RDEL: restart differential evolution algorithm with local search mutation for global numerical optimization," *Egyptian Informatics Journal*, vol. 15, no. 3, pp. 175–188, 2014.

[37] P. N. Suganthan, J. J. Liang, N. Hansen et al., "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization," Technical Report 2005005, Nanyang Tech. Uni. Singapore, May 2005 and KanGAL Report, IIT Kanpur, India, 2005.

[38] A. Auger, N. Hansen, J. M. P. Zerpa, R. Ros, and M. Schoenauer, "Experimental Comparisons of derivative free optimization algorithms," *Experimental Algorithms*, pp. 3–15, Springer, Berlin, Heidelberg, 2009.

[39] U. K. Chakraborty, S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition-based differential evolution," in *Studies in Computational Intelligence*, Springer, Berlin, Heidelberg, 2008.

[40] P. W. Tsai, J. S. Pan, P. Shi, and B. Y. Liao, "A new framework for optimization based-on hybrid swarm intelligence," in *Handbook of Swarm Intelligence*, vol. 23, pp. 421–449, Springer, Berlin, Heidelberg, 2011.

[41] Z. Wang and C. Yin, "Chicken swarm optimization algorithm based on behavior feedback and logic reversal," *Journal of Beijing Institute of Technology*, vol. 27, no. 3, pp. 34–42, 2018.

[42] A. Mortazavi, V. Toğan, and A. Nuhoğlu, "Weight minimization of truss structures with sizing and layout variables using integrated particle swarm optimizer," *Journal of Civil Engineering & Management*, vol. 23, no. 8, pp. 985–1001, 2017.

[43] L. J. Li, Z. B. Huang, F. Liu, and Q. H. Wu, "A heuristic particle swarm optimizer for optimization of pin connected structures," *Computers & Structures*, vol. 85, no. 7-8, pp. 340–349, 2007.

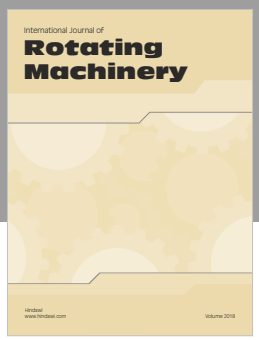
[44] Y. Li and S. Lian, "Improved fruit fly optimization algorithm incorporating Tabu search for optimizing the selection of elements in trusses," *KSCE Journal of Civil Engineering*, vol. 22, no. 12, pp. 4940–4954, 2018.

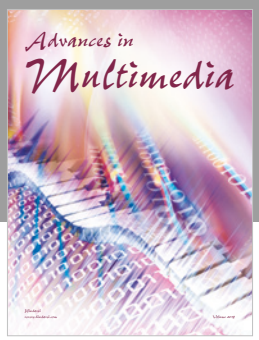
[45] F. Erbatur, O. Hasançebi, İ. Tütüncü, and H. Kılıç, "Optimal design of planar and space structures with genetic algorithms," *Computers & Structures*, vol. 75, no. 2, pp. 209–224, 2000.

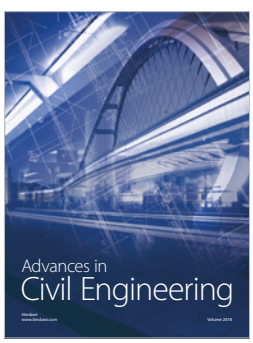
[46] H. Assimi, A. Jamali, and N. Nariman-Zadeh, "Multi-objective sizing and topology optimization of truss structures using genetic programming based on a new adaptive mutant operator," *Neural Computing & Applications*, vol. 23, no. 24, pp. 1–21, 2018.

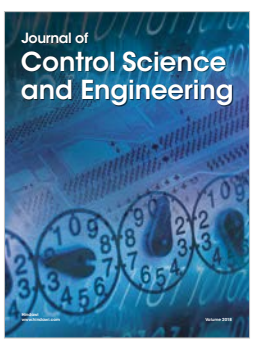
[47] M. Khatibinia and S. Sadegh Naseralavi, "Truss optimization on shape and sizing with frequency constraints based on orthogonal multi-gravitational search algorithm," *Journal of Sound and Vibration*, vol. 333, no. 24, pp. 6349–6369, 2014.

[48] T. Xu, W. Zuo, T. Xu, G. Song, and R. Li, "An adaptive reanalysis method for genetic algorithm with application to fast truss optimization," *Acta Mechanica Sinica*, vol. 26, no. 2, pp. 225–234, 2010.









Hindawi

Submit your manuscripts at
www.hindawi.com