**INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING**
***Int. J. Optim. Civil Eng., 2015; 5(1):37-52***

# AN IMPROVED INTELLIGENT ALGORITHM BASED ON THE GROUP SEARCH ALGORITHM AND THE ARTIFICIAL FISH SWARM ALGORITHM

Y. Y. Wang and L. J. Li*,†
*School of Civil and Transportation Engineering, Guangdong University Technology, Guangzhou, 510006, China*

## ABSTRACT

This article introduces two swarm intelligent algorithms, a group search optimizer (GSO) and an artificial fish swarm algorithm (AFSA). A single intelligent algorithm always has both merits in its specific formulation and deficiencies due to its inherent limitations. Therefore, we propose a mixture of these algorithms to create a new hybrid optimization algorithm known as the group search-artificial fish swarm algorithm (GS-AFSA). This algorithm has been applied to three different discrete truss optimization problems. The optimization results are compared with those obtained using the standard GSO, the AFSA and the quick group search optimizer (QGSO). The proposed GS-AFSA eliminated the shortcomings of GSO regarding falling into the local optimum by taking advantage of AFSA's stable convergence characteristics and achieving a better convergence rate and convergence accuracy than the GSO and the AFSA. Furthermore, the GS-AFSA has a superior convergence accuracy compared to the QGSO, all while solving a complicated structural optimization problem containing numerous design variables.

Received: 10 September 2014; Accepted: 20 December 2014

KEY WORDS: group search optimizer; artificial fish swarm algorithm; hybrid algorithm; structural optimization.

## 1. INTRODUCTION

Engineering structural optimization is increasingly demonstrating its utility and broad

*Corresponding author: School of Civil and Transportation Engineering, Guangdong University of Technology, Guangzhou, 510006, China
†E-mail address: lilj@gdut.edu.cn (L. J. Li)

prospects for application in a variety of areas to identify the most satisfactory of several feasible solutions. Engineering structural optimization has not only a crucial theoretical significance but also a practical value in reality. The techniques for searching for optimal solutions are known as optimization design methods and are generally classified into three categories: optimality criteria, mathematical programming and modern optimization algorithms. Optimality criteria's application is circumscribed. It requires different criteria for different types of constraints and the obtained solution is not necessarily the optimal solution [1, 2]. Mathematical programming performs a large amount of calculations and is characterized by its slow convergence rate [3]. As people deepen their understanding of the natural world, a new category of structural optimization techniques are being developed that borrow their working principles from natural phenomena. These include the genetic algorithm (GA) [4] and simulated annealing (SA) [5], etc. The field of nature-inspired algorithms is continuously growing. By the 1990s, swarm intelligence-based algorithms had been developed, including ant colony optimization (ACO) [6], particle swarm optimization (PSO) [7], the artificial fish swarm algorithm (AFSA) [8], the shuffled frog leaping algorithm (SFLA) [9], the group search optimizer (GSO) [10] and the meta-heuristic method [11, 12], all of which are inspired by social animal behaviors. The GSO is conceptually simple and easy to implement. It exhibits superior search performance in multi-modal function optimization problems [10]. The GSO has already been implemented for design optimization in truss structures and performs well when searching for the minimum weight [13, 14]. However, it has also been shown to fall into the local optimum when solving a complicated structural optimization problem space containing numerous design variables [14]. The AFSA has also been employed to solve truss optimization problems. Despite its slow convergence rate and ordinary convergence accuracy, it exhibits stable search performance and does not easily fall into the local optimum [15].

In the present work, to improve the search performance of GSOs when solving structural optimization problems, the AFSA is mixed with the GSO to create a new hybrid optimization algorithm known as the group search-artificial fish swarm algorithm (GS-AFSA). This algorithm exhibits a better convergence rate and convergence accuracy compared to the standard GSO [14], the AFSA and the quick group search optimizer (QGSO) [16].

## 2. GROUP SEARCH OPTIMIZER (GSO)

### *2.1 Multi-objective optimisation problems*

The group search optimizer is based on the model of producers and scroungers [17]. The population of the GSO is known as a group and each individual in the population is known as a member. In the GSO, a group consists of three types of members: producers, scroungers and rangers, with each playing a different part in the group. Each member starting an iteration in the most promising area, namely the area having the best fitness value, is chosen as the producer. The algorithm then stops and scans the environment to seek optima using a visual search. The other group members are selected as scroungers or rangers at random. The scroungers perform a random walk towards the producer, whereas the rangers perform a random walk in an arbitrary direction. Furthermore, these three different members do not differ in their relevant phenotypic characteristics, and can therefore switch between the three roles. The GSO behaves as follows [18]:

In an n-dimensional search space, the $i_{th}$ member at the $k_{th}$ searching bout (iteration) has a current position $X_i^k \in R^n$, a head angle $\varphi_i^k = (\varphi_{i1}^k, ..., \varphi_{i(n-1)}^k) \in R^{n-1}$ and a head direction $D_i^k(\varphi_i^k) = (d_{i1}^k, ..., d_{in}^k) \in R^n$, which can be calculated from $\varphi_i^k$ via a Polar to Cartesian coordinate transformation:

$$d_{i1}^k = \prod_{p=1}^{n-1} \cos(\varphi_{ip}^k) \tag{1}$$

$$d_{ij}^k = \sin(\varphi_{i(j-1)}^k) \cdot \prod_{p=i}^{n-1} \cos(\varphi_{ip}^k) \tag{2}$$

$$d_{in}^k = \sin(\varphi_{i(n-1)}^k) \tag{3}$$

In the GSO algorithm, the group consists of three individuals: producer, scroungers and rangers. At the $k_{th}$ iteration, the producer $X_p$ behaves as follows:

(1) The producer scans at zero degrees and then scans laterally by randomly sampling three points in the scanning field: one point at zero degrees, one point in the left hand side hypercube and one point in the right hand side hypercube:

$$X_z = X_p^k + r_1 l_{\max} D_p^k(\varphi^k) \tag{4}$$

$$X_l = X_p^k + r_1 l_{\max} D_p^k(\varphi^k - r_2\theta_{\max} / 2) \tag{5}$$

$$X_r = X_p^k + r_1 l_{\max} D_p^k(\varphi^k + r_2\theta_{\max} / 2) \tag{6}$$

where, $r_1 \in R^1$ is a normally distributed random number with a mean equal to 0 and standard deviation of 1 and $r_2 \in R^{n-1}$ is a random sequence in the range (0, 1). The variable $\theta_{\max} \in R^{n-1}$ is the maximum pursuit angle. The maximum pursuit distance $l_{\max}$ is calculated from:

$$l_{\max} = \| U_i - L_i \| = \sqrt{\sum_{i=1}^{n} (U_i - L_i)^2} \tag{7}$$

where $U_i$ and $L_i$ are the upper and lower bounds for the $i_{th}$ dimension, respectively.

(2)The producer will then find the best point with the best resource (fitness value). If the best point has a better resource than its current position, then it will move to this point. If not, will stay in its current position and examine a new angle:

$$\varphi^{k+1} = \varphi^{k} + r_2 \alpha_{\max} \tag{8}$$

where $\alpha_{\max}$ is the maximum turning angle.

(3) If the producer cannot find a better area after $a$ iterations, it will return to zero degrees:

$$\varphi^{k+a} = \varphi^{k} \tag{9}$$

where $a$ is a constant.

At the $k_{th}$ iteration, the area copying behavior of the $i_{th}$ scrounger can be modeled as a random walk towards the producer:

$$X_i^{k+1} = X_i^k + r_3 (X_p^k - X_i^k) \tag{10}$$

where, $r_3 \in R^n$ is a uniform random sequence in the range (0, 1).

In addition to the producers and the scroungers, a small number of rangers are also introduced into our GSO algorithm. Random walks, which are thought to be the most efficient searching method for randomly distributed resources, are employed by the rangers. If the $i_{th}$ group member is selected as a ranger at the $k_{th}$ iteration, the ranger generates a random head angle $\varphi_i$ :

$$\varphi^{k+1} = \varphi^{k} + r_2 \alpha_{\max} \tag{11}$$

where $\alpha_{\max}$ is the maximum turning angle. Next, it chooses a random distance:

$$l_i = a \cdot r_1 l_{\max} \tag{12}$$

and moves to the new point:

$$X_i^{k+1} = X_i^k + l_i D_i^k (\varphi^{k+1}) \tag{13}$$

## 3. ARTIFICIAL FISH SWARM ALGORITHM (AFSA)

The artificial fish swarm algorithm is a swarm intelligence optimization algorithm proposed by Li [19]. Inspired by the behavior of fish swimming in the ocean, Li presented a new fish swarm pattern with a bottom to top design philosophy. In nature, the areas containing large concentrations of fish are generally the most nutrient rich. This is because the fish can always find the more nutritious area by individual searching or by following after other fish.

Based on this peculiarity, the AFSA simulates fish behavior, such as preying, swarming, following and random motion to search for a globally optimal solution.

For a certain optimization problem, the position of each artificial fish represents a potential solution. The current position of an artificial fish $i$ can be expressed as $X_i = (x_{i1}, x_{i2}, ..., x_{in})$. The food concentration of fish $i$ is indicated as $Y_i = f(X_i)$, where $Y_i$ is the objective function. The variable parameters in the AFSA are listed in Table 1.

Table 1: Variable parameters of AFSA

| Variable | Variable meaning |
|---|---|
| $N$ | Size of artificial fish swarm |
| *Visual* | Sensor distance of artificial fish |
| $\delta$ | Congestion degree factor |
| *Step* | Maximum moving step of artificial fish |
| *Try_number* | Maximum attempt number of the preying behavior |

To reach a maximum objective function value, an example of the behavior of the artificial fish can be described as follows:

(1) Preying behavior: Assume $X_i$ is the current position of the artificial fish. Randomly select another position $X_j$ within the visual scope and compare the objective function value $f(X_i)$ with $f(X_j)$. If $f(X_i) < f(X_j)$, the artificial fish moves one step toward $X_j$, which is described by:

$$X_{next} = X_i + \frac{X_j - X_i}{\left\| X_j - X_i \right\|} \cdot step \cdot rand() \tag{4}$$

Otherwise, repeat the operation by selecting a position $X_j$ and estimating whether to move or not. If the artificial fish still stands in situ after $try\_number$ times, it will begin to implement random behavior.

(2) Swarming behavior: The artificial fish moves to the center of the group, which is considered to be a type of habit to ensure the existence of the colony and avoid dangers. Assume first that the current position of the artificial fish is $X_i$. The distance between the artificial fish is defined as $d_{i,j} = \left\| X_i - X_j \right\|$. In the range of $d_{i,j} < Visual$, the number of partners is $n_f$, the central position is $X_c$ and $\delta$ is the congestion degree factor. The expression $\left( Y_c / n_f \right) > \delta Y_i$ means the food concentration of the partner center is higher than at the current position and the congestion degree is not excessive. Therefore, the artificial fish moves one step toward $X_c$:

$$X_{next} = X_i + \frac{X_c - X_i}{\|X_c - X_i\|} \cdot step \cdot rand() \qquad (15)$$

Otherwise, the preying behavior is implemented.

(3) Following behavior: The artificial fish trails its neighboring partner within its visual scope, whose position has a higher food concentration. Suppose that $X_i$ is the current position of the artificial fish. The number of the artificial fish partners in the range from $d_{i,j} < Visual$ is $n_f$, and $X_j$ is the one with the highest food concentration $Y_j$ among the partners. The expression $(Y_j / n_f) > \delta Y_i$ means that the food concentration of $X_j$ is higher than at the current position and the congestion degree is excessive, meaning that the artificial fish moves one step toward $X_j$:

$$X_{next} = X_i + \frac{X_j - X_i}{\|X_j - X_i\|} \cdot step \cdot rand() \qquad (16)$$

Otherwise, implement the preying behavior.

(4) Random behavior: The artificial fish makes a random move within its visual scope. This is a default action for a preying behavior. The next position of $X_i$ is calculated from:

$$X_{next} = X_i + Visual \cdot rand() \qquad (17)$$

## 4. GROUP SEARCH HYBRID ARTIFICIAL FISH SWARM ALGORITHM (GS-AFSA)

The GS-AFSA can be said to be an improved version of the GSO to a certain extent. In this study, it has been combined with the AFSA search mechanism to avoid becoming entrapped in the local optimum, resulting in a higher probability of reaching the global optimum. The realization of the GS-AFSA is described as follows.

In an n-dimensional search space, the $i_{th}$ member at the $k_{th}$ searching bout (iteration) has a current position $X_i^k \in R^n$ and the position of each member is initialized by a random value prior to the start of the iterative search.

At the $k_{th}$ searching iteration, we calculate the fitness of each member. The best member is taken to be the global best member as well as the producer with a fitness value of $fvalue$. At this point, the GSO described in Eqs. (1-13) is implemented. The new group is now evaluated with the best member taken to be the producer with a fitness value of $fbestval$. If $fbestval < fvalue$, the producer is updated, and $fbestval$ replaces $fvalue$ and the search proceeds to next iteration. Otherwise, we implement the AFSA search mechanism.

In the AFSA search mechanism, the selection of artificial fish begins with the groups determined using the GSO. Each artificial fish implements the swarming and following behaviors, respectively. The fitness values from the new positions obtained from these two behaviors are determined to select the better one for the next fish position. It is worth noting that the preying behavior is contained in both the swarming and following behaviors as a supplement. Additionally, the random behavior is a default version of the preying behavior.

The pseudo code for the structural optimization using the GS-AFSA is listed in Table 2.

Table 2: Pseudo code for structural optimization by GS-AFSA

```
Set k=0;
Randomly initialize positions X_i and head angles φ_i of all members;
FOR (each member i in the group)
WHILE (the constraints are violated)
Randomly re-generate the current member X_i
END WHILE
END FOR
Calculate fitness: Calculate the fitness value of current member: f(X_i)
Choose producer: Find the producer X_p of the group;
WHILE (the termination condition is not met)
Set k=k+1;
FOR (each members i in the group)
The producer, scroungers and rangers update their positions by the equations of GSO.
IF (the variable boundary conditions are violated)
Make the variables which exceed the boundary fly back to their previous value
END IF
IF (the constraints are violated)
Ask the member fly back to its previous position
END IF
Calculate the fitness value of each member; update the producer and the search angle.
Check whether the algorithm move forward. If not, enter the AFSA. If it does, skip the
AFSA.
AFSA: Update the position of artificial fish by the equations of AFSA algorithm. Check
whether it violates the variable boundary conditions. If it does, make the variables which
exceed the boundary fly back to their previous value. Check whether it violates the
constrains. If it does, ask it fly back to its previous position.
Calculate the fitness value of each member; update the producer of GSO.
END FOR
END WHILE
```

## 5. NUMERICAL EXAMPLES

In this section, two planar truss structures and one spatial truss structure commonly seen in the literature are selected as benchmark structures to test the GS-AFSA. For the GSO, 20% of the population are selected as rangers and the initial head angle $\varphi_0$ for each member is set to $\pi / 4$. The constant $a$ is provided by $round\left(\sqrt{n+1}\right)$. The maximum pursuit angle $\theta_{max}$ is $\pi / a^2$. The maximum turning angle $\alpha_{max}$ is set to be $\pi / 2a^2$. For the QGSO, when the target proceeds forward, the parameters are set as: information transfer factors of $W_1 = W_2 = 4$, a selected probability of $W_3$=0.2 and a component mutation probability of $W_4$=0.65. Other parameters are set as: information transfer factors of $W_1 = 0.8$ and $W_2 = 1.5$, a selected probability of $W_3$=0.35 and a component mutation probability of $W_4$=0.85. For the AFSA, the artificial fish sensor distance, $Visual$, is set to $X_{max} / 2.5$ where $X_{max}$ is the maximum size of the search space. The maximum moving step for the artificial fish, known as $Step$, is $Visual / 10$. The congestion degree factor $\delta$ is 0.618. The maximum attempt number of the preying behavior, $try\_number$, is set to 50. For the GS-AFSA, the parameters are set to be the same as in the GSO and AFSA. For all the four algorithms, the population size is set to 50, and the termination condition is to permit the maximum number of iterations.

### *5.1 Example 1: The 10-bar planar truss structure*

The 10-bar planar truss structure is shown in Fig. 1. The material density is 0.1 lb/in$^3$ and the modulus of elasticity is 10$^4$ ksi. The stress limits of all the members are ±25 ksi. Nodes 1~4 in all directions are subjected to the displacement limits of ±2.0 in. The load case is listed in Table 3. The cross-sectional area of each bar is treated as an independent design variable, meaning there are 10 design variables in this optimization problem. There are two cases involved in the solution of this problem and each case has different set of the optional discrete variables, as follows. For case 1, the optional discrete variables are: D={1.62, 1.82, 1.99, 2.13, 2.38, 2.62, 2.63, 2.88, 2.93, 3.09, 3.13, 3.38, 3.47, 3.55, 3.63, 3.84, 3.87, 3.88, 4.18, 4.22, 4.49, 4.59, 4.80, 4.97, 5.12, 5.74, 7.22, 7.97, 11.50, 13.50, 13.90, 14.20, 15.50, 16.00, 16.90, 18.80, 19.90, 22.00, 22.90, 26.50, 30.00, 33.50}(in$^2$). For case 2, the optional discrete variables are: D={0.1, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 8.5, 9.0, 9.5, 10.0, 10.5, 11.0, 11.5, 12.0, 12.5, 13.0, 13.5, 14.0, 14.5, 15.0, 15.5, 16.0, 16.5, 17.0, 17.5, 18.0, 18.5, 19.0, 19.5, 20.0, 20.5, 21.0, 21.5, 22.0, 22.5, 23.0, 23.5, 24.0, 24.5, 25.0, 25.5, 26.0, 26.5, 27.0, 27.5, 28.0, 28.5, 29.0, 29.5, 30.0, 30.5, 31.0, 31.5}(in$^2$). The maximum number of iterations is set to 1000.

Tables 4 and 5 show the optimization results for the 10-bar planar truss structure. Similarly to the results from the QGSO, the GS-AFSA yields an identical design weight of 5490.74 lb for case 1, and 5067.33 lb for case 2, which are the best known solutions to the problem. The final designs attained using the GSO and AFSA are slightly larger: 5558.20 lb and 5763.95 lb, respectively for case 1, and 5074.79 lb and 5554.14 lb, respectively for case 2. The variations in the best feasible design obtained thus far in the search processes with the four algorithms are plotted in Figs 2 and 3. The convergence rates of GS-AFSA and QGSO are faster than for the GSO and AFSA. The AFSA has the slowest convergence rate of all the algorithms.

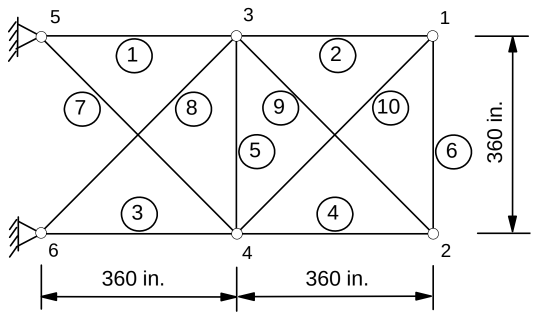


Figure 1. The 10-bar planar truss structure

Table 3: Load cases for the 10-bar planar truss structure

| Node | Load cases | |
|---|---|---|
| | $P_X$ (kips) | $P_Y$ (kips) |
| 1 | 0.0 | 0.0 |
| 2 | 0.0 | -100.0 |
| 3 | 0.0 | 0.0 |
| 4 | 0.0 | -100.0 |

Table 4: Optimization results for the 10-bar planar truss structure (case 1)

| Variables | Optimal cross-sectional areas ($in^2$) | | | |
|---|---|---|---|---|
| | QGSO [16] | GSO [14] | AFSA | GSAFSA |
| $A_1$ | 33.500 | 26.500 | 33.500 | 33.500 |
| $A_2$ | 1.620 | 1.620 | 2.130 | 1.620 |
| $A_3$ | 22.900 | 26.500 | 22.000 | 22.900 |
| $A_4$ | 14.200 | 15.500 | 13.500 | 14.200 |
| $A_5$ | 1.620 | 1.620 | 2.380 | 1.620 |
| $A_6$ | 1.620 | 1.620 | 3.840 | 1.620 |
| $A_7$ | 7.970 | 11.500 | 13.500 | 7.970 |
| $A_8$ | 22.900 | 22.000 | 22.000 | 22.900 |
| $A_9$ | 22.000 | 22.000 | 18.800 | 22.000 |
| $A_{10}$ | 1.620 | 1.800 | 4.220 | 1.620 |
| Weight (lb) | 5490.74 | 5558.20 | 5763.95 | 5490.74 |

Table 5: Optimization results for the 10-bar planar truss structure (case 2)

| Variables | Optimal cross-sectional areas ($in^2$) | | | |
|---|---|---|---|---|
| | QGSO [16] | GSO [14] | AFSA | GSAFSA |
| $A_1$ | 29.500 | 28.500 | 28.000 | 29.500 |
| $A_2$ | 0.100 | 0.100 | 2.500 | 0.100 |
| $A_3$ | 23.500 | 23.000 | 26.000 | 24.000 |
| $A_4$ | 15.500 | 16.500 | 16.500 | 15.000 |

| | | | | |
|---|---|---|---|---|
| $A_5$ | 0.100 | 0.100 | 1.000 | 0.100 |
| $A_6$ | 0.500 | 0.500 | 2.500 | 0.500 |
| $A_7$ | 7.500 | 7.500 | 11.500 | 7.500 |
| $A_8$ | 21.500 | 22.000 | 20.000 | 21.000 |
| $A_9$ | 21.500 | 21.500 | 20.500 | 22.000 |
| $A_{10}$ | 0.100 | 0.100 | 3.000 | 0.100 |
| Weight (lb) | 5067.33 | 5074.79 | 5554.14 | 5067.33 |

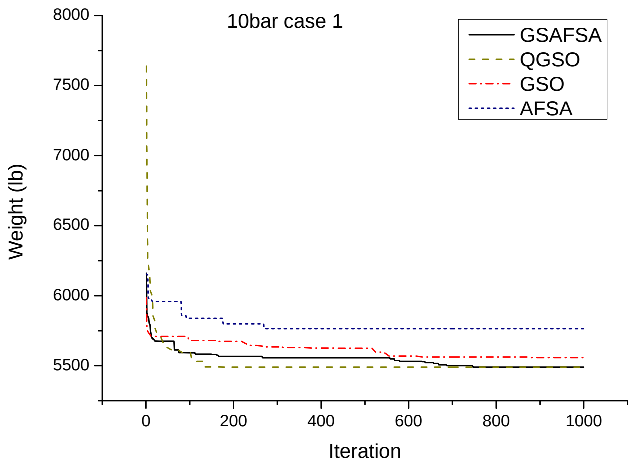


Figure 2. Convergence rate of the four algorithms for the 10-bar planar truss structure (case 1)

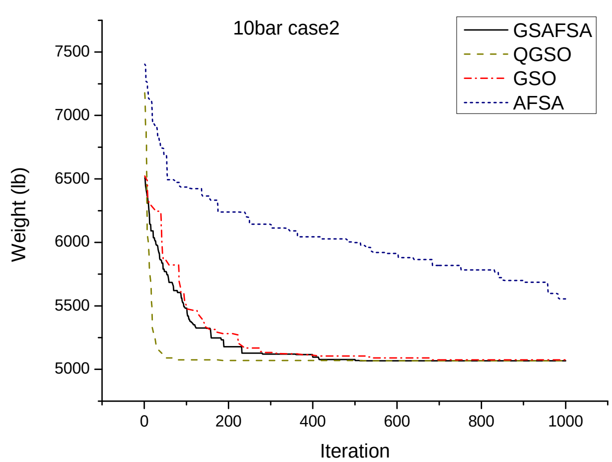


Figure 3. Convergence rate of the four algorithms for the 10-bar planar truss structure (case 2)

*5.2 Example 2: The 15-bar planar truss structure*

The 15-bar planar truss structure is shown in Fig. 4. The material density is 7800 kg/m$^3$ and the modulus of elasticity is 200 GPa. The stress limits of all the members are ±120 MPa. All nodes in all directions are subjected to the displacement limits of ±10 mm. The load cases are listed in Table 6. The cross-sectional area of each bar is treated as an independent design variable, meaning there are 15 design variables in this optimization problem. The optional discrete variables are: D={113.2, 143.2, 145.9, 174.9, 185.9, 235.9, 265.9, 297.1, 308.6, 334.3, 338.2, 497.8, 507.6, 736.7, 791.2, 1063.7}(mm$^2$). The maximum number of iterations is set to 500.

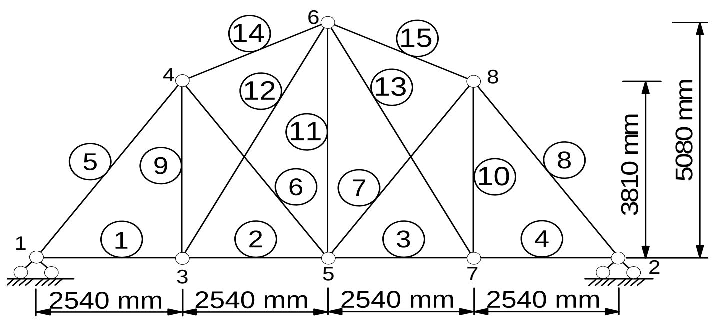


Figure 4. The 15-bar planar truss structure

Table 6: Load cases for the 15-bar planar truss structure

| Node | Load case 1 | | Load case 2 | | Load case 3 | |
|---|---|---|---|---|---|---|
| | $P_X$(kN) | $P_Y$(kN) | $P_X$(kN) | $P_Y$(kN) | $P_X$(kN) | $P_Y$(kN) |
| 4 | 0.0 | -35.0 | 0.0 | -35.0 | 0.0 | -35.0 |
| 6 | 0.0 | -35.0 | 0.0 | 0.0 | 0.0 | -35.0 |
| 8 | 0.0 | -35.0 | 0.0 | -35.0 | 0.0 | 0.0 |

Table 7 shows that the GS-AFSA, the QGSO and the GSO algorithms all achieve the same result of 105.735 kg after 500 iterations, whereas the AFSA algorithm reaches a relatively heavier weight of 123.874 kg. Moreover, Fig. 5 shows that again the convergence rate of the AFSA is much slower than in the other three algorithms.

Table 7: Optimization results for the 15-bar planar truss structure

| Variables | Optimal cross-sectional areas (mm$^2$) | | | |
|---|---|---|---|---|
| | QGSO [16] | GSO [14] | AFSA | GSAFSA |
| $A_1$ | 113.200 | 113.200 | 145.900 | 113.200 |
| $A_2$ | 113.200 | 113.200 | 265.900 | 113.200 |

| | | | | |
|---|---|---|---|---|
| $A_3$ | 113.200 | 113.200 | 145.900 | 113.200 |
| $A_4$ | 113.200 | 113.200 | 113.200 | 113.200 |
| $A_5$ | 736.700 | 736.700 | 736.700 | 736.700 |
| $A_6$ | 113.200 | 113.200 | 143.200 | 113.200 |
| $A_7$ | 113.200 | 113.200 | 145.900 | 113.200 |
| $A_8$ | 736.700 | 736.700 | 736.700 | 736.700 |
| $A_9$ | 113.200 | 113.200 | 113.200 | 113.200 |
| $A_{10}$ | 113.200 | 113.200 | 143.200 | 113.200 |
| $A_{11}$ | 113.200 | 113.200 | 265.900 | 113.200 |
| $A_{12}$ | 113.200 | 113.200 | 143.200 | 113.200 |
| $A_{13}$ | 113.200 | 113.200 | 185.900 | 113.200 |
| $A_{14}$ | 334.300 | 334.300 | 338.200 | 334.300 |
| $A_{15}$ | 334.300 | 334.300 | 334.300 | 334.300 |
| Weight (kg) | 105.735 | 105.735 | 123.874 | 105.735 |

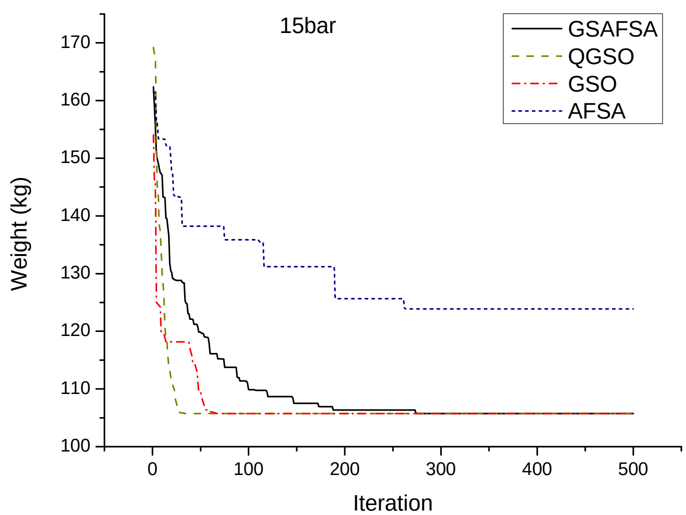


Figure 5. Convergence rate of the four algorithms for the 15-bar planar truss structure

### *5.3 Example 3: The 72-bar spatial truss structure*

The 72-bar spatial truss structure is shown in Fig. 6. The material density is 0.1 lb/in$^3$ and the modulus of elasticity is $10^7$ psi. The stress limits of all the members are ±25 ksi. All nodes in all directions are subjected to the displacement limits of ±0.25 in. The load cases are listed in Table 8. There are 72 bars, which are divided into 16 groups: (1) $A_1$~$A_4$, (2) $A_5$~$A_{12}$, (3) $A_{13}$~$A_{16}$, (4) $A_{17}$~$A_{18}$, (5) $A_{19}$~$A_{22}$, (6) $A_{23}$~$A_{30}$, (7) $A_{31}$~$A_{34}$, (8) $A_{35}$~$A_{36}$, (9) $A_{37}$~$A_{40}$, (10) $A_{41}$~$A_{48}$, (11) $A_{49}$~$A_{52}$, (12) $A_{53}$~$A_{54}$, (13) $A_{55}$~$A_{58}$, (14) $A_{59}$~$A_{66}$, (15) $A_{67}$~$A_{70}$ and (16) $A_{71}$~$A_{72}$. The optional discrete variables are: D={0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 3.0, 3.1, 3.2}(in$^2$). The maximum number of iterations is set to 1000.

Table 8: Load cases for the 72-bar spatial truss structure

| Node | Load case 1 | | | Load case 2 | | |
|---|---|---|---|---|---|---|
| | $P_X$ (kips) | $P_Y$ (kips) | $P_Z$ (kips) | $P_X$ (kips) | $P_Y$ (kips) | $P_Z$ (kips) |
| 17 | 5.0 | 5.0 | -5.0 | 0.0 | 0.0 | -5.0 |
| 18 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -5.0 |
| 19 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -5.0 |
| 20 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -5.0 |

Table 9 shows that the GS-AFSA achieves a design weight of 371.42 lb, which is better than any results previously reported using the various algorithms in the literature. Additionally, Fig. 7 shows that the GS-AFSA and the QGSO have the fastest convergence rates. The AFSA exhibits a fast convergence to its optimum, but the final result is unsatisfactory. The GSO becomes trapped in a local minimum.

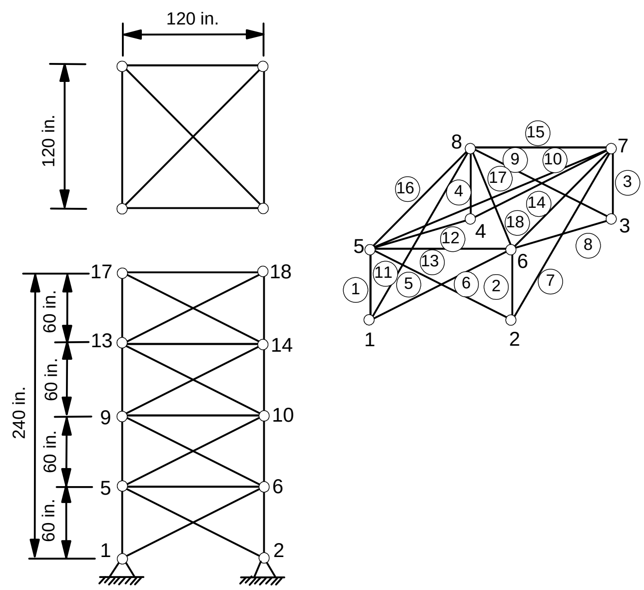


Figure 6. The 72-bar spatial truss structure

Table 9: Optimization results for the 72-bar spatial truss structure

| Variables | Optimal cross-sectional areas (in$^2$) | | | |
|---|---|---|---|---|
| | QGSO [16] | GSO [14] | AFSA | GSAFSA |
| $A_1$~$A_4$ | 2.0 | 3.0 | 1.6 | 1.8 |
| $A_5$~$A_{12}$ | 0.5 | 1.5 | 0.4 | 0.5 |
| $A_{13}$~$A_{16}$ | 0.1 | 0.1 | 0.4 | 0.1 |

| | | | | |
|---|---|---|---|---|
| $A_{17}$~$A_{18}$ | 0.1 | 0.1 | 0.6 | 0.1 |
| $A_{19}$~$A_{22}$ | 1.3 | 2.6 | 1.4 | 1.3 |
| $A_{23}$~$A_{30}$ | 0.5 | 1.5 | 0.5 | 0.5 |
| $A_{31}$~$A_{34}$ | 0.1 | 0.1 | 0.4 | 0.1 |
| $A_{35}$~$A_{36}$ | 0.1 | 0.1 | 0.7 | 0.1 |
| $A_{37}$~$A_{40}$ | 0.5 | 1.6 | 0.6 | 0.6 |
| $A_{41}$~$A_{48}$ | 0.5 | 1.4 | 0.7 | 0.5 |
| $A_{49}$~$A_{52}$ | 0.1 | 0.1 | 0.4 | 0.1 |
| $A_{53}$~$A_{54}$ | 0.1 | 0.4 | 0.6 | 0.1 |
| $A_{55}$~$A_{58}$ | 0.2 | 0.4 | 0.4 | 0.1 |
| $A_{59}$~$A_{66}$ | 0.6 | 1.6 | 0.6 | 0.5 |
| $A_{67}$~$A_{70}$ | 0.4 | 1.3 | 0.4 | 0.5 |
| $A_{71}$~$A_{72}$ | 0.6 | 1.3 | 1.2 | 0.5 |
| Weight (lb) | 385.54 | 967.68 | 514.15 | 371.42 |

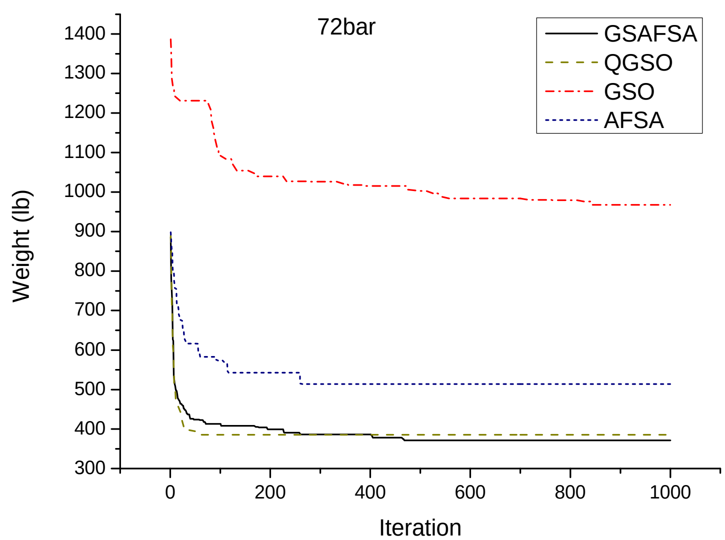


Figure 7. Convergence rate of the four algorithms for the 72-bar spatial truss structure

## 6. CONCLUSIONS

In this paper, a new hybrid optimization algorithm known as the group search-artificial fish swarm algorithm (GS-AFSA) is presented. It not only overcomes the deficiencies observed in both the GSO and AFSA but also retains the merits of both. Furthermore, compared to the quick group search optimizer (QGSO), the GS-AFSA has nearly the same convergence rate and exhibits better convergence accuracy while handling a relatively complicated structural optimization problem space with numerous design variables. The numerical results of these three examples demonstrate the efficiency of the GS-AFSA for structural optimization problems, and we expect the algorithm to be utilized to optimize complex real-world structures.

**Acknowledgements:** This work is supported by the National Natural Science Foundation of China (51178121) and the Guangdong Natural Science Foundation (S2012020011082).

## REFERENCES

1. Sui YK, Qiao ZH, Ye BR. The cross-section optimum design of truss structure using the full stress design method, in *Proceedings of the 12th National Structural Engineering Academic Conference in Chongqing, China*, 2003: pp. 1671-1677.
2. Liu ST. An improved tooth method for size optimum design of truss structure with displacement constraint, in *Proceedings of the 18th National Structural Engineering Academic Conference in Guangzhou, China*, 2009: pp. 419-424.
3. Cai X, Guo XW, Zhang XM. *Engineering Structural Optimization Design*, China, Water Power Press, Beijing, 2003.
4. Holland JH. *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, 1975.
5. Kirpatric S, Gelatt Jr CD, Vecchi MP. Optimization by simulated annealing, *Science* 1983; **220**: pp. 671-80.
6. Colorni A, Dorigo M, Maniezzo V. Distributed Optimization by Ant Colonies, in *Proceedings of the First European Conference on Artificial Life in Paris, France*, 1991, MIT Press, pp. 134-142.
7. Kenndy J, Eberhart RC. Particle swarm optimization, in *Proceedings of the 1995 IEEE International Conference on Neural Networks in Piscataway, NJ, USA,* 1995: pp. 1942-1948.
8. Li XL, Shao ZJ, Qian JX. An optimizing method based on autonomous animals: fish-swarm algorithm, *Systems Engineering-Theory & Practice* 2002, **22**(11): 32-8.
9. Eusuff MM, Lansey KE. Optimization of Water Distribution Network Design Using Shuffled Frog Leaping Algorithm, *Journal of Water Resources Planning and Management* 2003; **129**(3) 210-25.
10. He S, Wu QH, Sanders JR. A novel group search optimizer inspired by animal behavioural ecology, in *2006 IEEE Congress on Evolutionary Computation in Vancouver, BC, Canada,* 2006: pp. 4415-4421.
11. A. Kaveh and M. Khayatazad, A new meta-heuristic method: ray optimization, *Computers and Structures* 2012; **112-113**: pp. 283–94.
12. Kaveh A., Mahdavai, V.R. Colliding bodies optimization: A novel meta-heuristic method, *Computers and Structures* 2014; **139**: pp. 18-27.
13. Liu F, Xu XT, Li LJ. The group search optimizer and its application on truss structure design, in *The 4th International Conference on Natural Computation in Jinan, China,* 2008: pp. 688-692.

14. Li LJ, Xu XT, Liu F. The group search optimizer based on swarm intelligence and its application on optimization design with discrete variables, in *Proceedings of Steel Structure Stability and Fatigue Branch of China in Shenyang, China,* 2008: pp. 592-596.
15. Peng Y. The improved artificial fish swarm algorithm and its application to the structural optimal design, Guangdong University of Technology, 2012.
16. Liu F, Qin G, Li LJ. A quick group search optimizer and its application research, *Engineering Mechanics* 2010, **27**(7): pp. 38-44.
17. Barnard CJ, Sibly RM. Producers and scroungers: a general model and its application to captive flocks of house sparrows, *Animal Behaviour* 1981, **29**: pp. 543-550.
18. Li LJ, Liu F. *Group Search Optimization for Applications in Structural Design*, Springer Berlin, Heidelberg, 2011.
19. Li XL. *A New Intelligent Optimization Algorithm: Artificial Fish Swarm Algorithm*, Zhejiang University, 2003.