

Tối ưu kết cấu dàn thép sử dụng thuật toán bầy sói xám

Nguyễn Hữu Kỳ^{1*}, Lê Minh Hoàng²

¹ Công ty cổ phần thiết bị kiểm định xây dựng sài gòn (SAIGONIC), Tầng 2, 64 Trương Định, Phường 7, Quận 3, TP.HCM

² Khoa xây dựng, Trường Đại học Mở. TPHCM, số 35 - 37 Đường Hồ Hảo Hớn, Phường Cô Giang, Quận 1, TP.HCM

TỪ KHOÁ

Tối ưu kết cấu
Tối ưu dàn thép
Thuật toán thông minh
Thuật toán bầy sói

TÓM TẮT

Bài báo giới thiệu một kỹ thuật tối ưu mới trong bài toán thiết kế kết cấu dàn thép. Kỹ thuật tối ưu này dựa trên sự kết hợp giữa một thuật toán tối ưu hiệu quả được phát triển gần đây có tên là thuật toán bầy sói xám (Grey Wolf Optimizer) và quá trình cập nhật mô hình kết cấu một cách liên tục (model updating) để tối ưu hàm mục tiêu. Quá trình này sẽ được thực hiện lặp lại một cách liên tục cho đến khi hàm mục tiêu đạt được giá trị hội tụ. Để chứng minh được tính hiệu quả của thuật toán giới thiệu trong bài báo này, bốn hàm benchmark trong CEC2005 được sử dụng như một ví dụ đầu tiên để đánh giá khả năng tìm kiếm giá trị tối ưu toàn cục của thuật toán GWO. Sau đó một kết cấu dàn thép bao gồm 25 thanh dàn với 10 nút được bố trí trong không gian 3 chiều được sử dụng để áp dụng cho bài toán thiết kế tối ưu. Hàm mục tiêu của bài toán tối ưu được thiết lập dựa trên việc khống chế ứng suất trong mỗi thanh dàn và chuyển vị cho phép tại mỗi nút dàn không được vượt qua các giá trị giới hạn.

KEYWORDS

Structural optimization
Truss structures optimization
Optimization algorithm
Grey Wolf Optimizer

ABSTRACT

The paper introduces a new technique for the structural optimization of steel truss structures. This new technique was based on the combination of an effective optimization algorithm named Grey Wolf Optimizer (GWO) and the process of model updating to minimize the objective function. This process would be repeated continuously until the objective function reaches the convergence value. To prove the effectiveness of the optimization algorithm introduced, five benchmark functions were employed as the first example to evaluate the ability to find the global optimal. Then, a steel truss structure including 25 bar truss and 10 nodes were used to verify the proposed method. The objective function for the optimization problem was established based on the conditions of stress in each truss bar and the displacement at each node, which were not exceeded the limitation values.

1. Đặc điểm chung của thuật toán thông minh

Việc sử dụng thuật toán thông minh trong việc giải quyết các bài toán liên quan đến bài toán thiết kế, trong đó có bài toán về tối ưu tiến diện và khối lượng của kết cấu ngày càng trở nên phổ biến. Đã có nhiều thuật toán thông minh được đề xuất trong suốt hơn một thập kỷ qua. Một số thuật toán điển hình có thể kể đến như PSO [1], ACO [2], ABC [3], WSO [4], (BA) [5], WOA [6], (CS) [7],...Nhiều thuật toán trong đó lấy cảm hứng từ trí thông minh bầy đàn trong tự nhiên, mô phỏng chúng dưới dạng lý thuyết ngẫu nhiên để thiết lập nên quá trình di chuyển của mỗi cá thể trong bầy đàn. Việc hoàn thiện vị trí của các cá thể này được thực hiện thông qua quá trình lặp đi lặp lại đến khi đạt được giá trị hội tụ. Quá trình tự học này thông thường được chia làm hai nhiệm vụ chính của mỗi cá thể trong bầy đàn là quá trình khai thác (exploitation) và quá trình khám phá (exploration). Sự cân bằng giữa hai đặc điểm này đóng vai trò quan trọng cho sự thành công của thuật toán. Quá trình khai thác chính là quá trình tìm kiếm các giải pháp tiềm năng ở xung quanh vị trí của

giá trị tối ưu toàn cục được tìm thấy ở vòng lặp trước. Quá trình khám phá trong mỗi thuật toán nhằm đảm bảo khả năng tránh được các khả năng bị mắc kẹt tại các giá trị tối ưu cục bộ. Quá trình này thông thường sẽ có khoảng không gian tìm kiếm rộng và sẽ tập trung vào những vùng tìm kiếm mới có khoảng cách xa hơn so với vùng có giá trị tối ưu toàn cục. Quá trình áp dụng các thuật toán tìm kiếm thông minh để tối ưu hóa kết cấu đã chứng minh được tính hiệu quả và mức độ tin cậy thông qua các nghiên cứu đã được ông bố [8-12]. Việc tìm một thuật toán thông minh phù hợp và hiệu quả để giải quyết bài toán tối ưu kết cấu đã và đang tiếp tục được nghiên cứu khi trong vòng một thập kỷ qua. Ngày càng có nhiều thuật toán mới được đề xuất mới để giải quyết các bài toán thực tế trong đó có các bài toán về kết cấu. Trong những thuật toán được đề xuất mới, thuật toán bầy sói xám (GWO) được đánh giá là mang tính hiệu quả và đạt được độ ổn định cao, thuật toán này đã và đang được sử dụng nhiều để giải quyết các bài toán tối ưu ở nhiều ngành và nhiều lĩnh vực khác nhau.

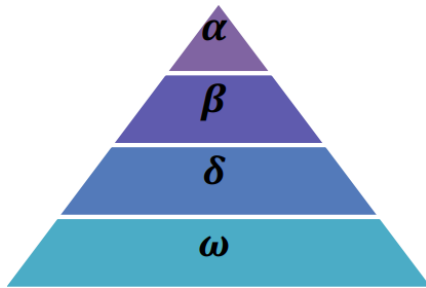
*Liên hệ tác giả: nguyenuhuukyngic@gmail.com

Nhận ngày 05/10/2021, sửa xong ngày 04/12/2021, chấp nhận đăng 01/04/2022

Link DOI: <https://doi.org/10.54772/jomc.02.2022.264>

2. Thuật toán bầy sói xám (Grey wolf optimizer)

Thuật toán bầy sói xám lấy cảm hứng từ việc phân cấp xã hội trong mỗi đàn sói xám. Theo đó trong bầy sói xám thông thường được phân cấp thành bốn nhóm được gọi tên theo thứ tự lần lượt từ cao nhất đến thấp nhất là; anpha (α), Bêta (β), delta (δ) và omega (ω) như thể hiện ở **Hình 1**. Mỗi vị trí trong bầy sói xám đảm bảo một nhiệm vụ khác nhau đảm bảo cho sự phát triển và tồn tại của bầy sói. Trong đó anpha (α) đóng vai trò như những nhóm cá thể đứng đầu trong bầy sói, chúng có nhiệm vụ đưa ra những quyết định về chiến lược săn mồi, tìm chỗ ở và phân chia thức ăn. Nhóm vị trí thứ hai là Bêta (β), chúng có nhiệm vụ hỗ trợ cho nhóm anpha (α) đưa ra các quyết định. Những nhóm Bêta (β) này có khả năng trở thành nhóm anpha (α) trong trường hợp các con trong nhóm anpha (α) rời đi hoặc chết đi. Nhóm cuối cùng trong phân cấp xã hội của bầy sói là nhóm omega (ω), nhóm này ở tầng thấp nhất và phải luôn tuân thủ các mệnh lệnh. Chúng có nhiệm vụ duy trì số lượng của bầy đàn. Nhóm còn lại nếu không thuộc vào những nhóm đã đề cập bên trên thì sẽ thuộc vào nhóm delta (δ).



Hình 1. Sự Phân cấp của bầy sói trong tự nhiên.

Sự phân cấp xã hội của bầy sói xám và chiến lược săn mồi của bầy sói được mô phỏng lại dưới dạng toán học để thiết lập các hướng di chuyển của mỗi con sói xám trong bầy. Quá trình di chuyển này phải đảm bảo có sự hoàn thiện thông qua mỗi lần lặp lại, hay nói cách khác, xác suất của lần di chuyển sau phải tìm được vị trí tốt hơn so với lần di chuyển trước. Trong thuật toán GWO, vị trí tối ưu nhất sau mỗi lần lặp được xem như là giá trị anpha (α) và các con sói trong đàn sẽ có hai xu hướng di chuyển để tấn công con mồi (khả năng exploitation) để hoàn thiện độ chính xác hoặc rời xa con mồi (khả năng exploration) để thoát được các điểm tối ưu cục bộ (local optimal). Quá trình này được mô tả dưới dạng các công thức toán học như sau:

2.1. Mô tả toán học của sự bao vây con mồi của đàn sói

Chiến lược bao vây con mồi của bầy sói được mô tả bởi hai phương trình Eq. (1) và Eq. (2) có dạng như sau:

$$\vec{D} = |\vec{C}\vec{X}_p(t) - \vec{X}(t)| \tag{1}$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A}\cdot\vec{D} \tag{2}$$

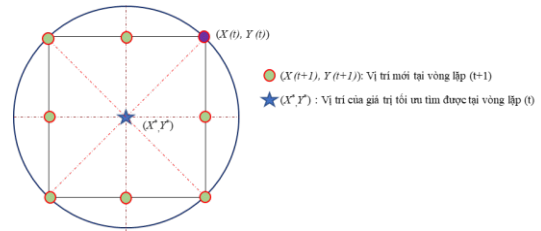
Trong đó (t) được xác định là số vòng lặp thứ t^{th} , \vec{A} và \vec{C} là hai véc tơ hệ số sẽ nhận giá trị ngẫu nhiên trong toàn bộ quá trình lặp. \vec{X}_p là vị trí của con mồi, và $\vec{X}(t)$ là vị trí hiện tại của mỗi con sói xám trong đàn tại vòng lặp thứ t^{th} .

Hai véc tơ \vec{A} và \vec{C} sẽ là những véc tơ đảm bảo yếu tố ngẫu nhiên trong thuật toán GWO và được tính toán theo các công thức Eq. (3) và Eq.(4) sau

$$\vec{A} = 2\vec{a}\cdot\vec{r}_1 - \vec{a} \tag{3}$$

$$\vec{C} = 2\cdot\vec{r}_2 \tag{4}$$

Trong đó \vec{a} là một đại lượng vô hướng sẽ có giá trị giảm từ 2 về 0 trong suốt quá trình lặp và \vec{r}_1, \vec{r}_2 là hai vector ngẫu nhiên mà mỗi giá trị của nó nhận giá trị trong khoảng [0, 1]. Có thể thấy được nhờ các phương trình Eq. (1) đến Eq.(4), vị trí của mỗi con sói trong hệ tọa độ (X,Y) có thể đạt được các vị trí khác nhau xung quanh con mồi như trong **Hình 2**.



Hình 2. Quá trình cập nhật vị trí giữa vòng lặp thứ t^{th} và vòng lặp thứ $(t+1)^{th}$ trong không gian 2 chiều.

2.2. Mô tả toán học của chiến lược săn mồi của bầy sói xám

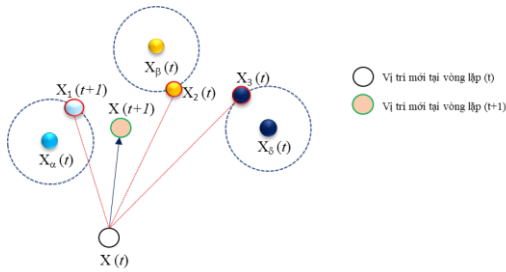
Sau khi bao vây con mồi, đàn sói xám sẽ tiến hành chiến lược săn mồi của mình, chiến lược này sẽ được dẫn dắt bởi ba nhóm là; anpha (α), Bêta (β) và delta (δ) được biết đến như là ba giá trị tốt nhất của hàm mục tiêu tại mỗi vòng lặp. Những vị trí này được xem là những vị trí tiềm năng, theo đó tại mỗi vòng lặp t^{th} , những vị trí này sẽ được lưu lại và sẽ đóng một vai trò như một sự định hướng cho lần cập nhật vị trí tại bước lặp thứ $(t+1)^{th}$. Những sự định hướng của ba nhóm này được miêu tả như các phương trình Eq. (5), Eq. (6), Eq. (7) và quá trình cập nhật vị trí của mỗi con sói được minh họa như ở **Hình 3**.

$$\vec{D}_\alpha = |\vec{C}_1\vec{X}_\alpha(t) - \vec{X}(t)|, \vec{D}_\beta = |\vec{C}_2\vec{X}_\beta(t) - \vec{X}(t)|, \vec{D}_\delta = |\vec{C}_3\vec{X}_\delta(t) - \vec{X}(t)| \tag{5}$$

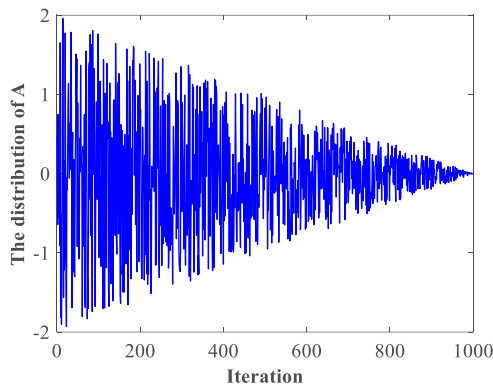
$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1\cdot(\vec{D}_\alpha), \vec{X}_2 = \vec{X}_\beta - \vec{A}_2\cdot(\vec{D}_\beta), \vec{X}_3 = \vec{X}_\delta - \vec{A}_3\cdot(\vec{D}_\delta) \tag{6}$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{7}$$

Sự cân bằng giữa khả năng khai thác và khả năng khám phá của thuật toán GWO được đảm bảo nhờ sự phân bố của giá trị $\vec{A} = 2\vec{a}\cdot\vec{r}_1 - \vec{a}$ được cho trong công thức Eq. (3) và được thể hiện ở **Hình 4**. Theo đó có thể nhận thấy rằng giá trị của A có thể nhận giá trị dương hoặc giá trị âm. Nếu A nhận giá trị dương, điều kiện khai thác các vị trí xung quanh giá trị tối ưu cục bộ được đảm bảo và ngược lại, nếu A nhận giá trị âm, không gian tìm kiếm sẽ được mở rộng tại các vị trí xa với các vị trí của tối ưu toàn cục được tìm thấy ở vòng lặp trước nó.



Hình 3. Quá trình cập nhật vị trí giữa vòng lặp thứ t^h và vòng lặp thứ $(t + 1)^h$ trong thuật toán GWO.



Hình 4. Sự phân bố của giá trị của A theo số vòng lặp tương ứng.

3. Ví dụ số

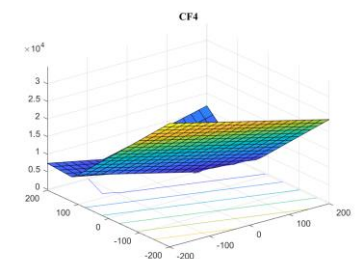
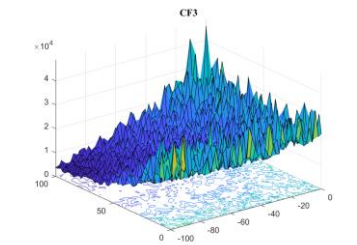
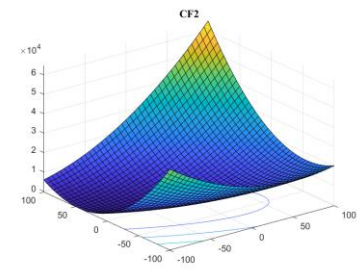
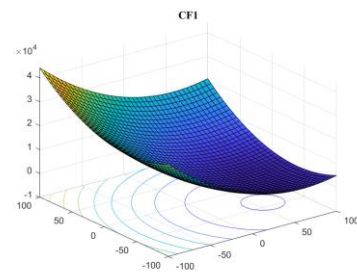
Để đánh giá độ tin cậy và tính hiệu quả của thuật toán GWO bốn hàm số được đặt tên là CF1, CF2, CF3, CF4 trong các hàm số của CEC 2005 [13] được lựa chọn để khảo sát khả năng tìm kiếm giá trị tối ưu toàn cục cũng như khả năng thoát khỏi giá trị tối ưu cục bộ của thuật toán GWO. Các Hàm số được sắp xếp từ đơn giản đến phức tạp và được cho ở trong **Bảng 1**, các đồ thị tương ứng trong không gian 2 chiều được minh họa ở Hình 5. Đặc điểm của những hàm số này là có mức độ phức tạp cao và có nhiều điểm tối ưu cục bộ như thể hiện ở hàm CF3, Hình 5. Không gian tìm kiếm được lựa chọn $D = 10$, Các thông số ban đầu của thuật toán GWO được thiết lập như sau:

- Số lượng các con sói xám trong đàn: $N = 30$
- Số vòng lặp thực hiện: $T = 2000$;
- Thuật toán GWO sẽ được thực thi với 30 lần độc lập với nhau để thu được các giá trị đặc trưng là giá trị nhỏ nhất (best value), giá trị lớn nhất (worst value) giá trị giá trị trung bình (mean value). Các giá trị được cho ở

Dựa vào kết quả thống kê ở **Bảng 2** có thể nhận thấy rằng các giá trị tối ưu toàn cục của các hàm số đều có thể được tìm thấy bởi thuật toán GWO. Dựa vào chiến lược di chuyển phù hợp, với những bước di chuyển đủ ngắn giữa hai lần lặp, thuật toán mức độ chính xác của thuật toán GWO được đánh giá là hoàn thiện và có thể vượt qua được những vấn đề về tối ưu cục bộ.

Bảng 1. Bốn hàm số CF1, CF2, CF3, CF4 trong các hàm CEC2005.

Hàm số	Miêu tả	Số biến của hàm số	Không gian tìm kiếm	Giá trị tối ưu toàn cục
Unimodal functions				
CF1	Shifted Sphere Function	10	$[-100, 100]$	-450
CF2	Shifted Schwefel's Problem 1.2	10	$[-100, 100]$	-450
CF3	Shifted Schwefel's Problem 1.2 with Noise in Fitness	10	$[-100, 100]$	-450
CF4	Schwefel's Problem 2.6 with Global Optimum on Bounds	10	$[-100, 100]$	-310



Hình 5. Đồ thị của các hàm số CF1, CF2, CF3, CF4 trong không gian 2 chiều.

Bảng 2. Kết quả các giá trị nhỏ nhất, lớn nhất và trung bình của bốn hàm CEC2005 sau 30 lần chạy độc lập sử dụng thuật toán GWO.

Giá trị	Hàm số			
	CF1	CF2	CF3	CF4
Best	-450,000	-449,950	-449,841	-309,671
Worst	-335,470	2525,290	3921,182	154,361
Mean	-431,910	-225,840	479,632	-162,675

4. Sử dụng thuật toán gwo để tối ưu kết cấu dàn

Thuật toán GWO đã chứng minh được tính hiệu quả và mức độ tin cậy thông qua ví dụ ở mục trước. Trong phần này, một kết cấu dàn thép được sử dụng để tối ưu tiết diện và khối lượng bằng GWO. Kết cấu bao gồm 25 thanh dàn được bố trí theo dạng không gian 3 chiều với tổng cộng 25 thanh và 10 nút.

4.1. Thiết lập hàm mục tiêu cho bài toán tối ưu kết cấu dàn

Mục tiêu của bài toán tối ưu kết cấu dàn thông thường là tập trung vào việc tối ưu khối lượng và tiết diện các thanh dàn với điều kiện đảm bảo được các yêu cầu về ứng suất và chuyển vị cho phép. Khi kích thước hình học của thanh dàn là cố định, bài toán tối ưu khối lượng có thể được biết dưới dạng sau:

Tìm giá trị nhỏ nhất của tổng khối lượng:

$$w = \sum_{e=1}^{N_m} \gamma_e L_e A_e \tag{8}$$

Thỏa mãn các điều kiện sau:

$$\begin{aligned} \sigma^L &\leq \sigma_e \leq \sigma^U \\ \delta^L &\leq \delta_c \leq \delta^U \\ A^L &\leq A_e \leq A^U \end{aligned}$$

Trong đó:

w: là khối lượng của hệ kết cấu thanh dàn với số lượng thanh dàn trong kết cấu là N_m thanh.

γ_e : Là trọng lượng riêng của vật liệu thanh thứ e

L_e : Là chiều dài của thanh thứ e

A_e : Là tiết diện của thanh thứ e

Các thanh dàn thiết kế có ứng suất trong thanh và chuyển vị tương ứng tại nút tương ứng là σ_e , δ_c phải thỏa mãn các điều kiện bền tương ứng σ^U và δ^U .

Các lý hiệu phụ "U" và "L" trong Eq. (8) tương ứng là các điều kiện biên và cận dưới của các thành phần ứng suất hoặc chuyển vị.

Để đảm bảo giới hạn của điều kiện bền và điều kiện chuyển vị, kỹ thuật hàm phạt (penalty method) được sử dụng. Theo đó sẽ có hai hàm phạt được sử dụng trong bài toán tối ưu, hàm phạt thứ nhất là hàm phạt về ứng suất được ký hiệu là ϕ_σ^e của thanh thứ "e". Hàm phạt thứ hai là hàm phạt về chuyển vị tại mỗi nút thứ "c" được ký hiệu là $\phi_\delta^e(x, y, z)$

Theo đó hàm phạt ứng suất đối với các thanh dàn được miêu tả dựa vào các điều kiện ở các công thức sau:

$$If \sigma^L \leq \sigma^e \leq \sigma^U, \phi_\sigma^e = 0 \tag{9}$$

$$If \sigma^e < \sigma^L \text{ or } \sigma^e > \sigma^U, \phi_\sigma^e = \left| \frac{\sigma^e - \sigma^{L,U}}{\sigma^{L,U}} \right|$$

Tổng giá trị hàm phạt theo điều kiện ứng suất cho vòng lặp thứ t^{th} được tính toán theo công thức sau:

$$\phi_\sigma(t) = \sum_{e=1}^{N_m} \phi_\sigma^e \tag{10}$$

Hàm phạt theo điều kiện chuyển vị tại nút được thiết lập dựa trên giá trị chuyển vị theo 3 phương x, y và z trong hệ tọa độ tổng thể theo các điều kiện sau:

$$If \delta^L \leq \delta_c(x, y, z) \leq \delta^U, \delta_c(x, y, z) = 0$$

$$If \delta_c(x, y, z) < \delta^L \text{ or } \delta_c(x, y, z) > \delta^U, \delta_c(x, y, z) = \left| \frac{\delta_c(x, y, z) - \delta^{L,U}}{\delta^{L,U}} \right| \tag{11}$$

Tổng giá trị hàm phạt theo điều kiện chuyển vị cho vòng lặp thứ t^{th} được tính toán theo công thức sau:

$$\phi_\delta(t) = \sum_{c=1}^{N_c} [\phi_\delta^e(x) + \phi_\delta^e(y) + \phi_\delta^e(z)] \tag{12}$$

Trong đó N_c là tổng số nút trong kết cấu thanh dàn.

Như vậy tổng hàm phạt của hệ kết cấu thanh dàn ở vòng lặp thứ t^{th} được tính toán như sau

$$\psi(t) = \left(\sum_{e=1}^{N_m} \phi_\sigma^e + \sum_{c=1}^{N_c} [\phi_\delta^e(x) + \phi_\delta^e(y) + \phi_\delta^e(z)] \right)^\epsilon \tag{13}$$

Trong đó ϵ là trọng số hàm phạt có giá trị lớn hơn 1 và luôn luôn là số dương.

Dựa vào các phương trình từ Eq. (8) đến Eq. (13). Hàm mục tiêu của bài toán thiết kế dàn tại vòng lặp thứ t^{th} được thiết lập theo công thức sau:

$$F(t) = \psi(t).w(t) \tag{14}$$

Trong đó $w(t)$ là tổng khối lượng các thanh dàn tại vòng lặp thứ t^{th} được tính toán theo công thức Eq. (8).

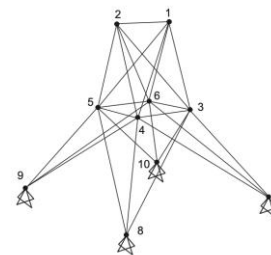
4.2. Tối ưu kết cấu hệ thanh dàn 25 thanh

Ví dụ đầu tiên là tối ưu tiết diện của một hệ kết cấu dàn thép có kích thước hình học như Hình 6. Hệ kết cấu bao gồm 10 nút với tọa độ mỗi nút được cho trong

Bảng 3. Hai trường hợp tải trọng với các điểm đặt lực trên các nút khác nhau được miêu tả chi tiết như trong

Bảng 4. Hệ kết cấu được chia thành 8 nhóm với đặc điểm là tiết diện ở mỗi nhóm là bằng nhau, số thanh ở mỗi nhóm và ứng suất cho phép tương ứng được miêu tả trong

Bảng 5. Giá trị chuyển vị cho phép tại mỗi nút trong hệ thanh dàn không được vượt quá ± 0.35 in theo mỗi phương (X,Y,Z).



Hình 6. Hình dạng kết cấu hệ dàn 25 thanh.

Bảng 3. Tọa độ mỗi nút trong hệ tọa độ tổng thể.

Node	X (in)	Y (in)	Z (in)
1	-37,5	0,0	200,0
2	37,5	0,0	200,0
3	-37,5	37,5	100,0
4	37,5	37,5	100
5	37,5	-37,5	100
6	-37,5	-37,5	100
7	-100	100	0,0
8	100	100	0,0
9	100	-100	0,0
10	-100	-100	0,0

Bảng 4. Hai trường hợp tải trọng tác dụng và tên nút tác dụng tương ứng.

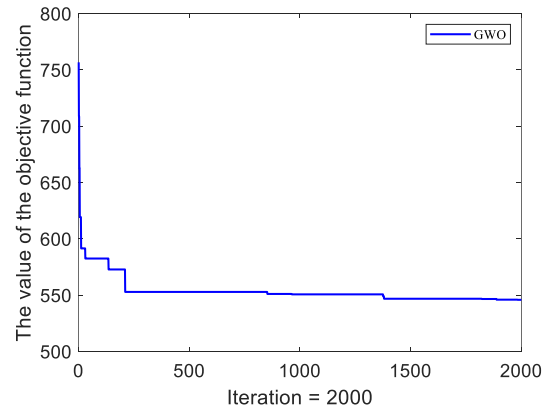
Case	Node	P_x (kips)	P_y (kips)	P_z (kips)
1	1	1,0	10,0	-5,0
	2	0,0	10,0	-5,0
	3	0,5	0,0	0,0
	6	0,5	0,0	0,0
2	1	0,0	20,0	-5,0
	2	0,0	-20,1	-5,0

Bảng 5. Phân bố số thanh trong mỗi nhóm và ứng suất cho phép trong mỗi nhóm.

Nhóm	Số thanh trong mỗi nhóm	Giá trị ứng suất nén cho phép (ksi)	Giá trị ứng suất kéo cho phép (ksi)
1	1	35,092	35,0
2	2;3;4;5	11,590	35,0
3	6;7;8;9	17,305	35,0
4	10;11	35,092	35,0
5	12;13	35,092	35,0
6	14;15;16;17	6,759	35,0
7	18;19;20;21	6,959	35,0
8	22;23;24;25	11,082	35,0

Hệ kết cấu được mô hình bằng phần mềm MATLAB với việc sử dụng phần tử dạng thanh (chi chịu nén và kéo). Theo đó, trong quá trình thực thi thuật toán GWO, tất cả các ứng suất trong 25 thanh dàn và chuyên vị tại mỗi nút sẽ được tính toán bằng phương pháp phần tử hữu hạn. Những kết quả này sau đó được chuyển đến thuật toán GWO để tính toán hàm mục tiêu. Quá trình sẽ được thực hiện một cách liên tục cho đến khi tìm được giá trị hội tụ. Điều kiện ban đầu của thuật toán GWO được thiết lập với số lượng các con sói

xám trong đàn là $N = 30$, Tổng số vòng lặp thực hiện là $T = 2000$. Kết quả của hàm mục tiêu tương ứng với số vòng lặp được miêu tả ở Hình 7. Kết quả tiết diện tối ưu và khối lượng đạt được sau $T = 2000$ sẽ được so sánh với các kết quả đạt được từ các thuật toán khác như GA [14] và CMLPSA [15]. Các kết quả so sánh được trình bày trong Bảng 6. Căn cứ vào kết quả so sánh có thể nhận thấy rằng, thuật toán GWO cho kết quả tốt nhất so với các thuật toán còn lại khi khối lượng tìm được bởi thuật toán GWO là $w = 545.7549$ (lb) chỉ với 2000 vòng lặp.



Hình 7. Lịch sử hội tụ của hàm mục tiêu.

Bảng 6. Kết quả tiết diện tối ưu và khối lượng tối ưu tìm được sau $T = 2000$ vòng lặp.

Nhóm	Phần tử thanh	Tiết diện tối ưu sử dụng thuật toán GWO (in ²)	Tiết diện tối ưu sử dụng thuật toán GA (in ²)	Tiết diện tối ưu sử dụng thuật toán CMLPSA (in ²)
1	1	0,08129	0,01	0.01
2	2;3;4;5	1,9597	2,0119	1.987
3	6;7;8;9	3,0596	2,9493	2.9935
4	10;11	0,01018	0,01	0.01
5	12;13	0,013185	0,0295	0.01
6	14;15;16;17	0,67888	0,6838	0.684
7	18;19;20;21	1,6696	1,6798	1.6769
8	22;23;24;25	2,6529	2,6759	2.6621
Khối lượng tối ưu (lb)		545,7549	545,8	547,86

5. Kết luận

Một thuật toán hiệu quả và có độ tin cậy cao dựa vào chiến lược săn mồi của bầy sói xám tên là GWO được giới thiệu trong bài báo này để áp dụng vào bài toán thiết kế tối ưu kết cấu dàn thép. Hai ví dụ được sử dụng trong bài báo này bao gồm bốn hàm số được lựa chọn trong 25 hàm số trong CEC2005 và một kết cấu dàn thép trong không gian 3 chiều với 25 thanh và 10 nút được sử dụng để chứng minh mức độ hiệu quả của thuật toán. Kết quả trong bài báo này đã

chứng minh rằng thuật toán GWO có thể tìm được giá trị tối ưu toàn cục với số vòng lặp đủ lớn ($T = 1000$ cho ví dụ 1 và $T = 2000$ cho ví dụ 2). Đặc biệt, đối với bài toán tối ưu kết cấu dàn thép, thuật toán GWO cho kết quả tốt nhất khi so sánh với hai thuật toán khác là GA và CMLPSA. Qua đó chứng minh được mức độ tin cậy và kỳ vọng của thuật toán GWO trong việc áp dụng để tối ưu các bài toán kết cấu có độ phức tạp hơn về mặt hình học cũng như số lượng thanh dàn. Đồng thời thuật toán GWO cũng là một thuật toán tiềm năng để giải quyết các bài toán tối ưu trong các lĩnh vực khác.

Tài liệu tham khảo

- [1]. R. Eberhart, J. Kennedy, Particle swarm optimization, *Proceedings of the IEEE international conference on neural networks*, vol 4, Citeseer, 1995, pp. 1942-1948.
- [2]. M. Dorigo, V. Maniezzo, A. Colomi, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 26 (1996) 29-41.
- [3]. B. Basturk, An artificial bee colony (ABC) algorithm for numeric function optimization, *IEEE Swarm Intelligence Symposium*, Indianapolis, IN, USA, 2006, 2006.
- [4]. P.C. Pinto, T.A. Runkler, J.M. Sousa, Wasp swarm algorithm for dynamic MAX-SAT problems, *International conference on adaptive and natural computing algorithms*, Springer, 2007, pp. 350-357.
- [5]. X.-S. Yang, A new metaheuristic bat-inspired algorithm, *Nature inspired cooperative strategies for optimization (NICSO 2010)*, Springer, 2010, pp. 65-74.
- [6]. S. Mirjalili, A. Lewis, *Advances in engineering software* 95 (2016) 51-67.
- [7]. X.-S. Yang, S. Deb, Cuckoo search via Lévy flights, *2009 World congress on nature & biologically inspired computing (NaBIC)*, Ieee, 2009, pp. 210-214.
- [8]. S.-J. Wu, P.-T. Chow, *Computers & structures* 56 (1995) 979-991.
- [9]. S. Huanchun, W. Yuefang, H. Jifeng, *Journal of DaLian University of Technology* 1 (1995).
- [10]. M. Sonmez, *Applied Soft Computing* 11 (2011) 2406-2418.
- [11]. V.B. Venkayya, *Computational Mechanics* 5 (1989) 1-21.
- [12]. C.V. Camp, M. Farshchin, *Engineering Structures* 62-63 (2014) 87-97.
- [13]. P.N. Suganthan, N. Hansen, J.J. Liang, K. Deb, Y.-P. Chen, A. Auger, S. Tiwari, KanGAL report 2005005 (2005) 2005.
- [14]. G. Cao, *Optimized design of framed structures using a genetic algorithm*, The University of Memphis, 1996.
- [15]. L. Lamberti, *Computers & Structures* 86 (2008) 1936-1953.