

TỐI ƯU TIẾT DIỆN NGANG KẾT CẤU DÀN THEO ĐIỀU KIỆN BỀN VÀ CHUYỂN VỊ BẰNG THUẬT TOÁN RAO

Trần Thùy Dương^{a,*}, Nguyễn Quang Huy^a, Phạm Hoàng Anh^a

^a*Khoa Xây dựng Dân dụng & Công nghiệp, Trường Đại học Xây dựng Hà Nội,
55 đường Giải Phóng, quận Hai Bà Trưng, Hà Nội, Việt Nam*

Nhận ngày 13/08/2021, Sửa xong 12/09/2021, Chấp nhận đăng 13/09/2021

Tóm tắt

Thuật toán Rao, thuật toán mới được phát triển năm 2020, là thuật toán mê-ta-ơ-rít-tíc có cấu trúc rất đơn giản, cho kết quả tốt trong một số bài toán tối ưu hàm toán. Mặc dù mới ra đời, thuật toán Rao đã đón nhận sự quan tâm ngày càng tăng từ cộng đồng nghiên cứu. Đã có các nghiên cứu ứng dụng thuật toán Rao để giải quyết một số bài toán tối ưu trong kỹ thuật. Tuy nhiên, ứng dụng của thuật toán Rao trong tối ưu kết cấu công trình vẫn chưa được khai thác đầy đủ. Bài báo này nghiên cứu khả năng và tính ứng dụng của thuật toán Rao đối với bài toán tối ưu hóa tiết diện ngang của kết cấu dàn theo điều kiện bền và chuyển vị. Một kỹ thuật xử lý điều kiện ràng buộc, kỹ thuật tìm kiếm biên khả thi (FBS), được sử dụng trong nghiên cứu. Hiệu quả của phương pháp đề xuất được đánh giá thông qua một số ví dụ số.

Từ khoá: tối ưu kết cấu dàn; thuật toán mê-ta-ơ-rít-tíc; thuật toán Rao; tìm kiếm biên khả thi (FBS).

OPTIMAL SIZING OF SPATIAL TRUSS USING CONSTRAINED RAO ALGORITHM

Abstract

Rao algorithms are newly developed metaheuristics. The algorithms have a simple structure, require no other parameter than the population size to perform, and are quite competitive in solving optimization problems. It is worth to examine Rao algorithms in complex optimization problems like truss structures. In this paper, the optimal sizing of spatial truss using Rao algorithms is explored. A constrained Rao algorithm is developed by integrating the Feasible Boundary Search (FBS) technique with the Rao-1 strategy. It has been demonstrated that Rao-1 with FBS is robust and computationally efficient in obtaining the minimum weight of two space trusses.

Keywords: truss sizing; metaheuristics; Rao algorithms; feasible boundary search.

[https://doi.org/10.31814/stce.huice\(nuce\)2021-15\(5V\)-06](https://doi.org/10.31814/stce.huice(nuce)2021-15(5V)-06) © 2021 Trường Đại học Xây dựng Hà Nội (ĐHXDHN)

1. Đặt vấn đề

Kết cấu dàn là một trong những loại kết cấu được sử dụng phổ biến hiện nay nhờ khả năng vượt nhịp lớn, hình thức đẹp và phong phú, phát huy tối đa khả năng của vật liệu. Việc tối ưu hóa kết cấu dàn là một bài toán thiết kế kết cấu phổ biến. Thông thường, hàm mục tiêu là cực tiểu hóa trọng lượng của dàn với ràng buộc về ứng suất, chuyển vị và/hoặc tần số dao động tự nhiên. Bài toán tối ưu hóa này được biết đến là một bài toán phức tạp do các ràng buộc phi tuyến và miền nghiệm không lồi, rất khó giải quyết bằng các phương pháp dựa trên gradient truyền thống. Nhiều nghiên cứu đã tập trung vào việc phát triển các kỹ thuật tối ưu hóa để giải quyết các bài toán tối ưu hóa dàn phức tạp này [1].

*Tác giả đại diện. Địa chỉ e-mail: duongtt@nuce.edu.vn (Dương, T. T.)

Trong những năm gần đây, các thuật toán mê-ta-ơ-rít-tíc, như thuật toán di truyền, thuật toán tiến hóa, các thuật toán dựa trên mô phỏng tự nhiên ngày càng được sử dụng nhiều hơn trong lĩnh vực tối ưu hóa đàn [2–8]. Tuy nhiên, phần lớn các thuật toán này cần các tham số điều khiển, do đó làm hạn chế khả năng ứng dụng trong bài toán thực tế. Vì vậy, cần có các kỹ thuật tối ưu hóa đơn giản có thể cung cấp các giải pháp đáng tin cậy cho bài toán tối ưu kết cấu đàn. Từ nhu cầu đó, các siêu mô phỏng không có tham số đã được phát triển như thuật toán TLBO (Teaching-Learning-Based-Optimization) [9], thuật toán Jaya [10] và Rao [11]. Trong số các thuật toán tối ưu hóa này, TLBO và Jaya đã có một số ứng dụng trong thiết kế đàn [12–25]. Thuật toán Rao là thuật toán mới được phát triển, có cấu trúc rất đơn giản, cho kết quả tốt trong một số bài toán tối ưu hàm toán. Tuy nhiên, việc áp dụng các thuật toán Rao trong thiết kế tối ưu kết cấu đàn vẫn còn chưa được khai thác đầy đủ.

Bài báo này nghiên cứu khả năng và tính ứng dụng của thuật toán Rao đối với bài toán tối ưu hóa tiến diện ngang của kết cấu đàn theo điều kiện bền và chuyển vị. Đặc biệt, một thuật toán Rao mới được thiết lập bằng cách kết hợp kỹ thuật tìm kiếm theo biên khả thi (FBS) với thuật toán Rao-1 để xử lý các ràng buộc một cách hiệu quả. Phương pháp đề xuất có tên RaoFBS, được kiểm tra qua các ví dụ số trên hai kết cấu đàn không gian với các biến liên tục. Hiệu quả của RaoFBS được so sánh với một số thuật toán mê-ta-ơ-rít-tíc khác.

2. Tối ưu kích thước đàn với ràng buộc ứng suất và chuyển vị

2.1. Biến thiết kế

Các biến thiết kế là diện tích tiết diện của các thanh đàn. Giá trị của các biến thiết kế được tìm kiếm trong phạm vi được xác định trước $A_{i,low} \leq A_i \leq A_{i,up}$ với $A_{i,low}$, $A_{i,up}$ là giới hạn dưới và giới hạn trên của diện tích mặt cắt ngang của thanh thứ i .

2.2. Hàm mục tiêu và các điều kiện ràng buộc

Mục tiêu của bài toán thiết kế là cực tiểu hóa trọng lượng của đàn bằng cách tìm tiết diện tối ưu của các thanh đàn sao cho ứng suất và chuyển vị phát sinh trong đàn nằm trong giới hạn cho phép (theo tiêu chuẩn). Với một kết cấu đàn cho trước, ta cần xác định một véc tơ của N_m diện tích mặt cắt ngang $\mathbf{x} = \{A_1, A_2, \dots, A_{N_m}\}$ sao cho tổng trọng lượng của đàn được cực tiểu hóa theo các ràng buộc đã cho. Bài toán được biểu diễn như sau:

$$\text{Minimize } W = \sum_{i=1}^{N_m} \rho_i L_i A_i \quad (1)$$

$$\begin{aligned} \text{s.t. } \quad & g_j \leq g_{0,j}, j = 1, \dots, N_c \\ & A_{i,low} \leq A_i \leq A_{i,up} \end{aligned} \quad (2)$$

với ρ_i và L_i lần lượt là trọng lượng riêng của vật liệu và chiều dài của thanh thứ i ; g_j là hàm ràng buộc thứ j ; $g_{0,j}$ là giá trị cho phép của g_j ; N_c là số lượng điều kiện ràng buộc. Trong nghiên cứu này, chỉ xem xét ràng buộc về ứng suất và chuyển vị.

3. Tìm kiếm theo biên khả thi

Xử lý ràng buộc là một trong những vấn đề quan trọng trong việc giải quyết các bài toán tối ưu hóa có ràng buộc. Một giải pháp thỏa mãn tất cả các ràng buộc được gọi là một giải pháp khả thi. Tập hợp các giải pháp khả thi tạo nên miền nghiệm (miền thiết kế). Trong bài toán tối ưu kết cấu đàn, ràng

buộc là ứng suất trong các thanh và chuyển vị nút không vượt quá giới hạn cho phép. Thực tế, giải pháp tối ưu thường nằm trên biên của miền thiết kế, do vậy việc giới hạn tìm kiếm trên biên của miền thiết kế sẽ đẩy nhanh quá trình giải bài toán tối ưu hóa. Phương pháp này được gọi là tìm kiếm theo biên khả thi (FBS), được giới thiệu bởi Baghlani và Makiabadi [26] trong bài toán tối ưu kích thước dàn bằng thuật toán bầy đom đóm (Firefly algorithm). FBS dựa trên mối quan hệ tuyến tính giữa tải trọng và đáp ứng (ứng suất và chuyển vị) của kết cấu dàn, và chỉ áp dụng hiệu quả cho bài toán phân tích tuyến tính. Phương pháp được trình bày như sau.

Giả thiết quan hệ ứng suất-biến dạng là quan hệ tuyến tính. Chuyển vị tại các nút được xác định theo từ quan hệ:

$$\mathbf{K} \cdot \mathbf{u} = \mathbf{f} \quad (3)$$

trong đó $\mathbf{K}$ là ma trận độ cứng; $\mathbf{f}$ là véc tơ lực nút; $\mathbf{u}$ là véc tơ chuyển vị nút. Lực dọc và ứng suất trong các thanh được xác định từ véc tơ chuyển vị nút. Có thể thấy, nếu nhân ma trận độ cứng với một số $\beta > 0$ có nghĩa diện tích tiết diện của tất cả các thanh được nhân với β , chuyển vị và ứng suất sẽ giảm đi $1/\beta$. Do đó, để thỏa mãn ràng buộc thứ j chỉ cần nhân diện tích tiết diện của tất cả các thanh dàn đang xét với một hệ số:

$$\beta_j = \max \left\{ 0, \frac{g_j}{g_{0,j}} \right\} \quad (4)$$

trong đó g_j là giá trị của ràng buộc thứ j (chuyển vị hoặc ứng suất) tương ứng với phương án đang xét. Để thỏa mãn tất cả các ràng buộc, ta nhân diện tích tiết diện của các thanh với hệ số:

$$\beta_{\max} = \max\{\beta_j\}, \quad j = 1, \dots, N_c \quad (5)$$

Để đảm bảo các thanh có diện tích tiết diện không nhỏ hơn giá trị nhỏ nhất cho phép, diện tích tiết diện các thanh được nhân với hệ số:

$$\alpha_{\max} = \max \left\{ 1, \frac{A_{i,low}}{A_i} \right\}, \quad i = 1, \dots, N_m \quad (6)$$

4. Thuật toán Rao tối ưu tiết diện thanh dàn

Thuật toán Rao, bao gồm Rao-1, Rao-2 và Rao -3, là những thuật toán mê-ta-ơ-rít-tíc không tham số dựa trên quần thể được phát triển bởi Rao [11]. Rao cũng là cha đẻ của hai thuật toán mê-ta-ơ-rít-tíc không tham số nổi tiếng khác là TLBO and Jaya. Các thuật toán Rao sử dụng các quy tắc rất đơn giản để tạo ra một giải pháp thiết kế mới từ một nhóm NP các giải pháp.

Tại bất kỳ bước lặp t nào, giải pháp thiết kế thứ p được cập nhật là:

- Thuật toán Rao-1:

$$\mathbf{x}_p^{new} = \mathbf{x}_p + \mathbf{r}_1 \cdot (\mathbf{x}_{best} - \mathbf{x}_{worst}) \quad (7)$$

- Thuật toán Rao-2:

$$\mathbf{x}_p^{new} = \mathbf{x}_p + \mathbf{r}_1 \cdot (\mathbf{x}_{best} - \mathbf{x}_{worst}) + \mathbf{r}_2 \cdot d \cdot (|\mathbf{x}_p| - |\mathbf{x}_r|) \quad (8)$$

- Thuật toán Rao-3:

$$\mathbf{x}_p^{new} = \mathbf{x}_p + \mathbf{r}_1 \cdot (\mathbf{x}_{best} - |\mathbf{x}_{worst}|) + \mathbf{r}_2 \cdot d \cdot (|\mathbf{x}_p| - \mathbf{x}_r) \quad (9)$$

với $\mathbf{x}_p$ là giải pháp ứng viên thứ p ; $\mathbf{x}_{best}$ và $\mathbf{x}_{worst}$ lần lượt là giải pháp tốt nhất và xấu nhất trong quần thể giải pháp; $\mathbf{x}_r$ là một giải pháp thiết kế được chọn ngẫu nhiên từ quần thể giải pháp; $\mathbf{r}_1$ và $\mathbf{r}_2$ là hai

véc tơ lấy ngẫu nhiên trong khoảng $[0, 1]$; d một số nhận giá trị 1 hoặc -1 tùy thuộc vào mối quan hệ thứ tự giữa $\mathbf{x}_p$ và $\mathbf{x}_r$: $d = 1$ nếu $\mathbf{x}_p$ tốt hơn $\mathbf{x}_r$; trường hợp khác $d = -1$.

Trong nghiên cứu này, các tác giả đề xuất thuật toán RaoFBS, là sự kết hợp của thuật toán Rao-1 với kỹ thuật FBS, để tối ưu kích thước tiết diện thanh dàn. Cấu trúc của thuật toán như sau:

- Định nghĩa hàm mục tiêu W , các điều kiện ràng buộc, NP , $Tmax$.

- Tạo quần thể gồm NP các giải pháp thiết kế, tạo giá trị ban đầu diện tích tiết diện thanh thứ i của phương án thiết kế $\mathbf{x}_p$:

$$A_{p,i} = A_{i,low} + rand() \times (A_{i,up} - A_{i,low})$$

$$p = 1, \dots, NP; \quad i = 1, \dots, N_m$$

- Xác định hàm mục tiêu W_p và các ràng buộc g_j cho $\mathbf{x}_p$.

- Xác định β_{max} , α_{max} cho bởi công thức (5) và (6), áp dụng kỹ thuật tìm kiếm FBS:

$$\mathbf{x}_p = \alpha_{max}\beta_{max}\mathbf{x}_p$$

$t = 1$;

while $t < Tmax$ **do**

 Tìm $\mathbf{x}_{best}$ và $\mathbf{x}_{worst}$ trong quần thể NP

for $p = 1$ **to** NP **do**

 Tạo véc tơ ngẫu nhiên $\mathbf{r}_1$

$$\mathbf{x}_p^{new} = \mathbf{x}_p + \mathbf{r}_1 \cdot (\mathbf{x}_{best} - \mathbf{x}_{worst});$$

 Xác định g_j và β_{max} , và α_{max} for $\mathbf{x}_p^{new}$

 Áp dụng FBS: $\mathbf{x}_p^{new} = \alpha_{max}\beta_{max}\mathbf{x}_p^{new}$;

 Tính toán W_p^{new} for $\mathbf{x}_p^{new}$

if $W_p^{new} \leq W_p$ **then**

$$\mathbf{x}_p = \mathbf{x}_p^{new};$$

end if

end for

$t = t + 1$;

end while

Cập nhật và xuất kết quả $\mathbf{x}_{best}$ và W_{best}

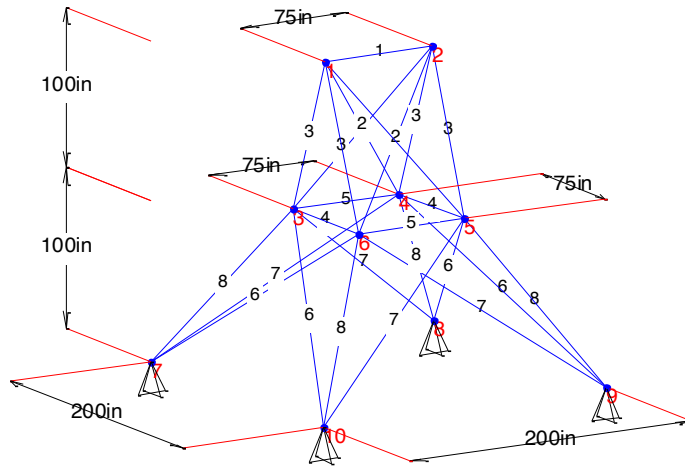
5. Áp dụng

Trong phần này, thuật toán RaoFBS được áp dụng để tối ưu kích thước tiết diện thanh của hai kết cấu dàn không gian 25 thanh và 72 thanh. Để làm rõ hiệu quả của thuật toán RaoFBS, kết quả tính toán theo RaoFBS sẽ được so sánh với kết quả tính toán theo một số thuật toán mê-ta-ơ-rít-tíc khác cũng sử dụng FBS (Thuật toán bầy đàn đốm kết hợp với FBS, FBSFA [26]; thuật giải thụ phấn hoa kết hợp với FBS, FBS-FPA [27] và thuật toán tiến hóa vi phân cải tiến (ODE-NNC) [28]). Các phương pháp dùng để so sánh được lựa chọn trong số những phương pháp hiệu quả nhất đã được áp dụng cho bài toán này. Kết quả tính toán còn được so sánh với thuật toán Rao-1 và Rao-2 sử dụng quy tắc xử lý ràng buộc của Deb [29].

5.1. Miêu tả bài toán

a. Dàn 25 thanh

Sơ đồ kết cấu dàn không gian 25 thanh được mô tả trên Hình 1. Mô đun đàn hồi và khối lượng riêng của vật liệu tương ứng là 104 ksi và $0,1 \text{ lb/in}^3$. Diện tích tiết diện của các thanh được chia thành 8 nhóm. Chuyển vị nút cho phép là 0,35 in. Ứng suất kéo cho phép là 40 ksi, và ứng suất nén cho phép (có xét đến yêu cầu ổn định của thanh chịu nén) đối với 8 nhóm là -35,092; -11,590; -17,305; -35,092; -35,092; -6,759; -6,959; -11,082 (ksi). Hai trường hợp tải trọng được xem xét có dữ liệu được cho trong Bảng 1. Diện tích tiết diện thanh (in^2) được chọn trong khoảng [0,01; 3,4].



Hình 1. Sơ đồ dàn không gian 25 thanh

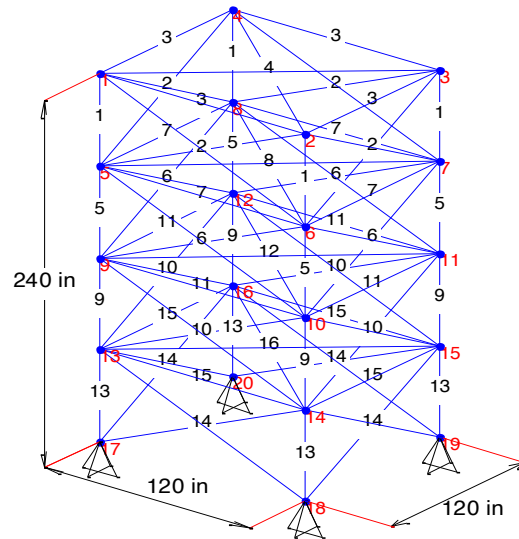
Bảng 1. Số liệu tải trọng cho dàn không gian 25 thanh

Trường hợp	Nút	F_x (kips)	F_y (kips)	F_z (kips)
1	1	0	20	-5
	2	0	-20	-5
2	1	1	10	-5
	2	0	10	-5
	3	0,5		
	6	0,5		

b. Dàn 72 thanh

Sơ đồ kết cấu dàn không gian 72 thanh được cho trên Hình 2. Các thanh được chia thành 16 nhóm, diện tích mặt cắt ngang được chọn trong khoảng $[0,1; 2,5] \text{ (in}^2\text{)}$. Các ràng buộc bao gồm chuyển vị nút cho phép 0,25 in theo cả hai phương x, y , ứng suất cho phép ± 25 ksi. Khối lượng riêng và mô đun đàn hồi của vật liệu lần lượt là $0,1 \text{ lb/in}^3$ và 104 ksi. Số liệu tải trọng được cho trong Bảng 2.

Thuật toán RaoFBS, Rao-1 và Rao-2 được sử dụng để tối ưu mỗi dàn với 20 lần chạy độc lập. Sử dụng phương pháp phần tử hữu hạn để tính toán chuyển vị nút và ứng suất trong các thanh. Kích thước quần thể cho cả hai trường hợp là $NP = 20$ và 25. Bước lặp tối đa là 350 cho cả dàn 25 thanh và dàn 72 thanh.



Hình 2. Sơ đồ dàn không gian 72 thanh

Bảng 2. Số liệu tải trọng cho dàn không gian 72 thanh

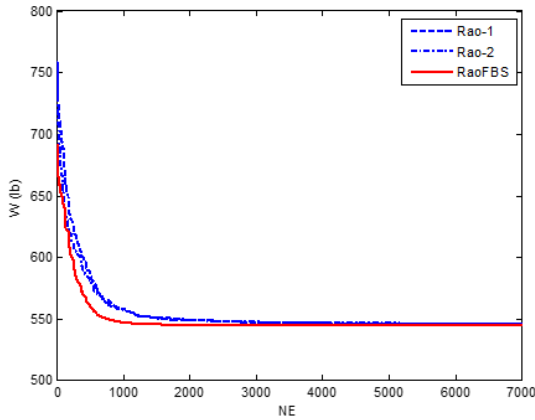
Trường hợp	Nút	F_x (kips)	F_y (kips)	F_z (kips)
1	1	5	5	-5
2	1	0	0	-5
	2	0	0	-5
	3	0	0	-5
	4	0	0	-5

5.2. Kết quả tính toán

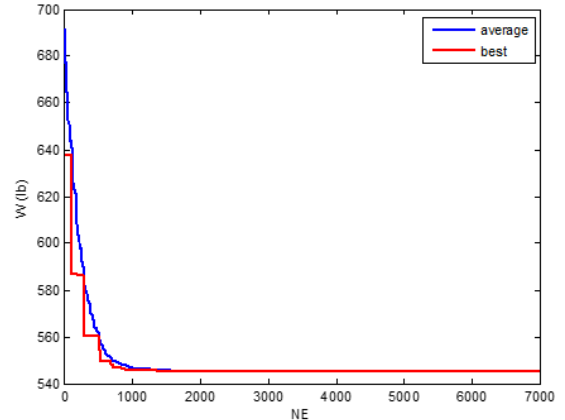
a. Dàn 25 thanh

Hình 3 mô tả đồ thị hội tụ kết quả trọng lượng tối ưu bằng thuật toán Rao. Hình 3(a) so sánh kết quả tính toán theo Rao-1, Rao-2 và RaoFBS. Hình 3(b) mô tả các kết quả trọng lượng tối ưu theo RaoFBS, bao gồm đường cong hội tụ trung bình (“average”) và đường cong trường hợp tốt nhất (“best”). Theo quan sát, RaoFBS hội tụ nhanh hơn Rao-1 và Rao-2. Theo RaoFBS, đường cong “best” hội tụ nhanh hơn đường cong trung bình.

Bảng 3 đưa ra kết quả tối ưu thu được bởi RaoFBS so sánh với kết quả tối ưu theo một số thuật toán mê-ta-ơ-rít-tíc đã được nghiên cứu trước đó bao gồm ODE-NNC [28], FBSFA [26], và FBS-FPA [27]. Trong đó, FBSFA và FBS-FPA cũng sử dụng FBS để xử lý các ràng buộc. Kết quả cho thấy, trọng lượng tối ưu theo RaoFBS so với các phương pháp khác chênh lệch rất ít. RaoFBS cũng cho kết quả ổn định với độ lệch rất nhỏ của trọng lượng trong 20 lần chạy. Ngoài ra, giải pháp tối ưu của RaoFBS trong 20 lần chạy thỏa mãn tất cả các ràng buộc. Các giá trị ràng buộc mà RaoFBS thu được bằng giới hạn cho phép. Với NP = 20, RaoFBS thu được trọng lượng nhỏ hơn so với trường hợp NP = 25, trong khi số lần phân tích kết cấu lại nhỏ hơn.



(a) So sánh giữa các thuật toán Rao



(b) Quá trình hội tụ theo RaoFBS

Hình 3. Quá trình tối ưu hóa cho dàn 25 thanh theo RaoFBS

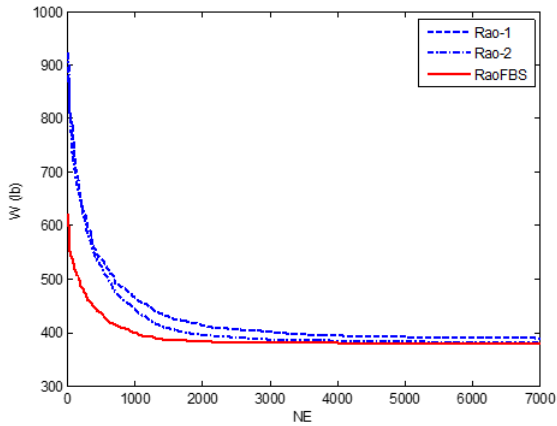
Bảng 3. Kết quả tối ưu cho dàn 25 thanh bằng RaoFBS và các phương pháp khác

Tham số	ODE-NNC [28]	FBSFA [26]	FBS-FPA [27]	RaoFBS NP = 25	RaoFBS NP = 20
Nhóm 1	0,01	0,01000	0,010002	0,010032	0,010103
Nhóm 2	1,98708	1,99248	1,9838	1,9962	1,9788
Nhóm 3	2,99347	2,98329	2,998	2,9784	3,005
Nhóm 4	0,01	0,01000	0,01001	0,010038	0,010009
Nhóm 5	0,01	0,01000	0,010052	0,01005	0,010035
Nhóm 6	0,68369	0,68417	0,68541	0,68457	0,68567
Nhóm 7	1,67688	1,6771	1,6775	1,6762	1,6786
Nhóm 8	2,66249	2,66454	2,659	2,6655	2,6565
W_{best} (lb)	545,1630	545,17	545,1630	545,1739	545,1685
W_{mean} (lb)	545,1649	-	545,1654	545,1819	545,1832
W_{worst} (lb)	-	-	545,1698	545,1992	545,2108
Độ lệch chuẩn	0,00252	-	0,00148	0,00609	0,01133
Ứng suất lớn nhất (ksi)	-	-6,9547	-6,95879	-6,959	-6,959
Chuyển vị lớn nhất (in)	-	0,35000	0,35	0,35	0,35
Số lần phân tích	5000	-	7025	8750	7000

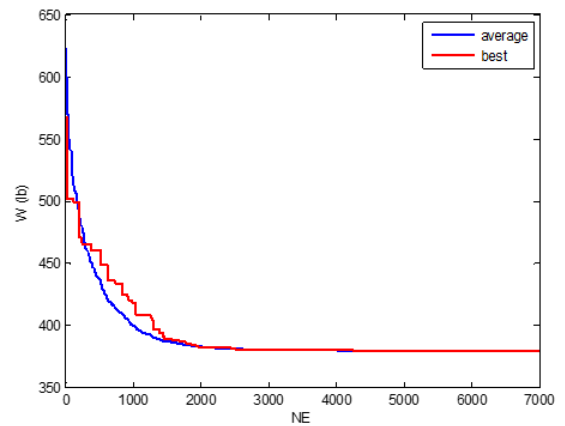
b. Dàn 72 thanh

Quá trình tối ưu hóa của các thuật toán Rao được thể hiện trên Hình 4. Một lần nữa, RaoFBS có tốc độ hội tụ cao hơn nhiều so với Rao-1 và Rao-2 (Hình 4(a)). Đối với ví dụ này, đường cong hội tụ phương án tốt nhất của RaoFBS hơi chậm hơn so với đường cong hội tụ trung bình (Hình 4(b)).

Kết quả tính toán theo RaoFBS được cho trong Bảng 4 và so sánh với một số phương pháp khác. Trong ví dụ này RaoFBS có số lần phân tích ít hơn các thuật toán khác và cho kết quả tương đương. Hơn nữa, RaoFBS là thuật toán ổn định nhất với độ lệch của trọng lượng tối ưu trong 20 lần chạy nhỏ hơn so với các thuật toán khác. Kết quả của RaoFBS cho trường hợp NP = 20 và NP = 25 không có nhiều khác biệt. Thậm chí với NP = 20, RaoFBS cho kết quả trung bình và độ ổn định tốt hơn so với NP = 25.



(a) So sánh giữa các thuật toán Rao



(b) Quá trình hội tụ theo RaoFBS

Hình 4. Quá trình tối ưu hóa cho dàn 72 thanh theo RaoFBS

Bảng 4. Kết quả tối ưu cho dàn 72 thanh bằng RaoFBS và các phương pháp khác

Tham số	ODE-NNC [28]	FBSFA [26]	FBS-FPA [27]	RaoFBS NP = 25	RaoFBS NP = 20
Nhóm 1	0,15648	0,15632	0,15642	0,15649	0,1564
Nhóm 2	0,54481	0,53325	0,54627	0,54642	0,54783
Nhóm 3	0,40971	0,43573	0,40783	0,41134	0,41655
Nhóm 4	0,56884	0,56554	0,57432	0,5747	0,56692
Nhóm 5	0,52508	0,57209	0,52565	0,51817	0,52146
Nhóm 6	0,51517	0,52720	0,51585	0,51745	0,51709
Nhóm 7	0,10000	0,10003	0,10001	0,10005	0,1
Nhóm 8	0,1	0,10237	0,10003	0,10009	0,1
Nhóm 9	1,26685	1,26497	1,2723	1,2639	1,2783
Nhóm 10	0,51361	0,50802	0,51141	0,51212	0,50883
Nhóm 11	0,1	0,10001	0,1	0,10001	0,10004
Nhóm 12	0,1	0,10003	0,1	0,10003	0,10005
Nhóm 13	1,88493	1,90107	1,8844	1,8914	1,877
Nhóm 14	0,51398	0,49688	0,51193	0,50978	0,51149
Nhóm 15	0,10000	0,10003	0,10006	0,10001	0,1
Nhóm 16	0,1	0,10003	0,1	0,10002	0,1
W_{best} (lb)	379,6175	379,93	379,6228	379,6284	379,6295
W_{mean} (lb)	379,6422	-	379,6503	379,6543	379,6480
W_{worst} (lb)	-	-	379,6888	379,7073	379,6864
Độ lệch chuẩn	0,02381	-	0,01865	0,01718	0,01263
Ứng suất lớn nhất (ksi)	-	24,9667	24,99808	25,0	25,0
Chuyển vị lớn nhất (in)	-	0,25000	0,25	0,25	0,25
Số lần phân tích	10000	-	9025	8750	7000

6. Kết luận

Nghiên cứu đã sử dụng thuật toán Rao thực hiện thành công bài toán tối ưu trọng lượng dàn với các biến liên tục. Hơn nữa, một thuật toán Rao (RaoFBS) đã được xây dựng bằng cách kết hợp kỹ thuật tìm kiếm trên miền khả thi (FBS) với thuật toán Rao-1. Thuật toán RaoFBS đã được chứng minh tương đương về độ tin cậy và hiệu quả tính toán so với một số thuật toán mê-ta-ơ-rít-tíc khác, bao gồm hai thuật toán có sử dụng kỹ thuật FBS trong bài toán tối ưu hóa hai dàn không gian. Ưu điểm của RaoFBS là thuật toán này không yêu cầu các tham số điều khiển như các thuật toán khác. Do đó, việc sử dụng RaoFBS sẽ thuận lợi trong thực tế thiết kế tối ưu kết cấu dàn.

Lời cảm ơn

Nghiên cứu này được tài trợ bởi Trường Đại học Xây dựng Hà Nội (HUCE) trong đề tài mã số 48-2021/KHXD-TĐ.

Tài liệu tham khảo

- [1] Stolpe, M. (2015). [Truss optimization with discrete design variables: a critical review](#). *Structural and Multidisciplinary Optimization*, 53(2):349–374.
- [2] Pham, H. A. (2016). [Truss optimization with frequency constraints using enhanced differential evolution based on adaptive directional mutation and nearest neighbor comparison](#). *Advances in Engineering Software*, 102:142–154.
- [3] Pham, A. H. (2016). [Discrete optimal sizing of truss using adaptive directional differential evolution](#). *Advances in Computational Design*, 1(3):275–296.
- [4] Anh, P. H. (2016). [An evolutionary-based optimization algorithm for truss sizing design](#). *Vietnam Journal of Mechanics*, 38(4):307–317.
- [5] Truong, V.-H., Kim, S.-E. (2018). [Reliability-based design optimization of nonlinear inelastic trusses using improved differential evolution algorithm](#). *Advances in Engineering Software*, 121:59–74.
- [6] Kaveh, A., Zolghadr, A. (2018). [Meta-heuristic methods for optimization of truss structures with vibration frequency constraints](#). *Acta Mechanica*, 229(10):3971–3992.
- [7] Degertekin, S. O. (2012). [Improved harmony search algorithms for sizing optimization of truss structures](#). *Computers & Structures*, 92-93:229–241.
- [8] Bureerat, S., Pholdee, N. (2016). [Optimal Truss Sizing Using an Adaptive Differential Evolution Algorithm](#). *Journal of Computing in Civil Engineering*, 30(2):04015019.
- [9] Rao, R. V., Savsani, V. J., Vakharia, D. P. (2011). [Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems](#). *Computer-Aided Design*, 43(3):303–315.
- [10] Rao, R. V. (2016). [Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems](#). *International Journal of Industrial Engineering Computations*, 19–34.
- [11] Rao, R. V. (2020). [Rao algorithms: Three metaphor-less simple algorithms for solving optimization problems](#). *International Journal of Industrial Engineering Computations*, 107–130.
- [12] Degertekin, S. O., Hayalioglu, M. S. (2013). [Sizing truss structures using teaching-learning-based optimization](#). *Computers & Structures*, 119:177–188.
- [13] Dede, T. (2014). [Application of Teaching-Learning-Based-Optimization algorithm for the discrete optimization of truss structures](#). *KSCE Journal of Civil Engineering*, 18(6):1759–1767.
- [14] Camp, C. V., Farshchin, M. (2014). [Design of space trusses using modified teaching–learning based optimization](#). *Engineering Structures*, 62-63:87–97.
- [15] Tejani, G. G., Savsani, V. J., Patel, V. K. (2016). [Modified sub-population teaching-learning-based optimization for design of truss structures with natural frequency constraints](#). *Mechanics Based Design of Structures and Machines*, 44(4):495–513.

- [16] Degertekin, S. O., Lamberti, L., Ugur, I. B. (2018). [Sizing, layout and topology design optimization of truss structures using the Jaya algorithm](#). *Applied Soft Computing*, 70:903–928.
- [17] Degertekin, S. O., Lamberti, L., Ugur, I. B. (2019). [Discrete sizing/layout/topology optimization of truss structures with an advanced Jaya algorithm](#). *Applied Soft Computing*, 79:363–390.
- [18] Farshchin, M., Camp, C. V., Maniat, M. (2016). [Multi-class teaching–learning-based optimization for truss design with frequency constraints](#). *Engineering Structures*, 106:355–369.
- [19] Baghlani, A., Makiabadi, M. H., Maheri, M. R. (2017). [Sizing Optimization of Truss Structures by an Efficient Constraint-Handling Strategy in TLBO](#). *Journal of Computing in Civil Engineering*, 31(4): 04017004.
- [20] Savsani, V. J., Tejani, G. G., Patel, V. K. (2016). [Truss topology optimization with static and dynamic constraints using modified subpopulation teaching–learning-based optimization](#). *Engineering Optimization*, 48(11):1990–2006.
- [21] Grzywinski, M., Dede, T., Ozdemir, Y. I. (2019). [Optimization of the braced dome structures by using Jaya algorithm with frequency constraints](#). *Steel and Composite Structures*, 30(1):47–55.
- [22] Dede, T., Grzywiński, M., Rao, R. V. (2019). [Jaya: A New Meta-heuristic Algorithm for the Optimization of Braced Dome Structures](#). *Advanced Engineering Optimization Through Intelligent Techniques*, Springer Singapore, 13–20.
- [23] Artar, M., Daloglu, A. T. (2019). [Optimum design of steel space truss towers under seismic effect using Jaya algorithm](#). *Structural Engineering and Mechanics*, 71(1):1–12.
- [24] Kaveh, A., Hamedani, K. B., Kamalinejad, M. (2020). [Set theoretical variants of the teaching–learning-based optimization algorithm for optimal design of truss structures with multiple frequency constraints](#). *Acta Mechanica*, 231(9):3645–3672.
- [25] Grzywiński, M., Selejdak, J., Dede, T. (2020). [Truss optimization with frequency constraints based on TLBO algorithm](#). *In AIP Conference Proceedings*, AIP Publishing.
- [26] Baghlani, A., Makiabadi, M. H. (2014). [Weight optimization of truss structures by a new feasible boundary search technique hybridized with firefly algorithm](#). *KSCE Journal of Civil Engineering*, 18(4):1105–1118.
- [27] Anh, P. H. (2016). [Optimal truss sizing with stress and displacement constraints](#). *Journal of Science and Technology in Civil Engineering (STCE) - HUCE*, 10(2):69–77. (in Vietnamese).
- [28] Anh, P. H. (2016). [Truss sizing optimization using enhanced differential evolution with opposition-based mutation and nearest neighbor comparison](#). *Journal of Science and Technology in Civil Engineering (STCE) - HUCE*, 10(5):3–10.
- [29] Deb, K. (2000). [An efficient constraint handling method for genetic algorithms](#). *Computer Methods in Applied Mechanics and Engineering*, 186(2-4):311–338.