

TỐI ƯU GIÀN THÉP PHI TUYẾN CÓ BIẾN THIẾT KẾ RỜI RẠC BẰNG THUẬT TOÁN TIẾN HÓA VI PHÂN CẢI TIẾN

OPTIMIZATION OF NONLINEAR INELASTIC STEEL TRUSSES WITH DISCRETE DESIGN VARIABLES USING AN IMPROVED DIFFERENTIAL EVOLUTION

TS. HÀ MẠNH HÙNG

Khoa Xây dựng dân dụng và công nghiệp, Trường đại học Xây dựng

Tóm tắt: Thiết kế tối ưu đang dần trở thành yêu cầu căn bản trong thiết kế công trình hiện nay nhằm tiết kiệm nguồn lực và chi phí. Tùy theo đặc điểm của biến thiết kế mà thiết kế tối ưu sẽ được phân ra thành thiết kế tối ưu với biến liên tục và thiết kế tối ưu với biến rời rạc. Thiết kế tối ưu với biến rời rạc thường gặp nhiều khó khăn hơn do đặc tính phi tuyến của bài toán cao hơn, tính không liên tục cũng gây ra khó khăn cho các chương trình tối ưu trong việc tìm kiếm cực trị toàn cục. Trong bài báo này, một thuật toán tối ưu cải tiến dựa trên thuật toán tiến hóa vi phân và nguyên lý bộ giá trị (Set-based Differential Evolution, S-DE) được trình bày nhằm tối ưu hóa dàn thép phi tuyến có biến thiết kế rời rạc. S-DE hoàn toàn tuân theo quy trình ban đầu của thuật toán DE nhưng có sự thay đổi về các khái niệm cơ bản liên quan đến thực hiện các toán tử trong kỹ thuật đột biến nhằm phù hợp với không gian rời rạc của biến thiết kế. Một giàn không gian 72 thanh được nghiên cứu để đánh giá hiệu quả của thuật toán được đề xuất.

Từ khóa: tiến hóa vi phân; giàn thép; phân tích trực tiếp; tối ưu.

Abstract: Optimization design is gradually becoming a basic requirement in today's building design to save resources and costs. Depending on the characteristics of design variables, the optimization design is divided into (1) optimization design with continuous variables and with discrete variables. Optimization design with discrete variables is often more nonlinear, and the discontinuity makes it difficult for optimization algorithms to find global solutions. In this paper, an improved optimization algorithm based on differential evolution algorithm and discrete-set characteristics (Set-based Differential Evolution) (S-DE) is presented to optimize nonlinear steel trusses with discrete design variables. S-DE completely

follows the original process of the DE algorithm, but there is a change in the basic concepts related to the implementation of the mutation technique to fit the discrete space of variables. A 72-bar steel truss is studied to evaluate the effectiveness of the proposed method.

Keywords: Differential evolution; Steel truss; Direct design; Optimzation.

1. Đặt vấn đề

Tính hiệu quả về mặt giá thành luôn được xem là một trong những tiêu chí quan trọng đối với công tác thiết kế công trình. Do vậy, thiết kế tối ưu được xem như một yêu cầu cơ bản cho các kỹ sư. Tuy nhiên, sự phức tạp của các công trình về mặt ứng xử, số lượng biến thiết kế, tải trọng tác dụng, tính hợp lý về mặt cấu tạo,... khiến cho bài toán thiết kế tối ưu công trình có tính phi tuyến và độ phức tạp cao, đặc biệt thường được xem xét dưới dạng các biến thiết kế là rời rạc. Đối với dạng bài toán thiết kế tối ưu này, các thuật toán metaheuristic (mê-ta heuristic) được xem là một công cụ hiệu quả. Ưu điểm của các thuật toán metaheuristic là khả năng cân bằng giữa việc tìm kiếm toàn cục và địa phương thông qua cơ chế tìm kiếm "khôn ngoan", "có định hướng" nhiều vòng dựa trên thông tin của các ứng viên thiết kế tiềm năng đã biết trước [1-5]. Về mặt cơ bản, cơ chế tìm kiếm "khôn ngoan" và "có định hướng" được thực hiện thông qua 3 bước: (1) đột biến (mutation), (2) lai tạo (crossover) và (3) chọn lọc. Trong phần lớn các thuật toán metaheuristic hiện nay, quá trình này được xây dựng thường dựa trên mô phỏng các đặc tính cơ bản của các biến thiết kế dạng liên tục. Việc xử lý các biến dạng rời rạc thường được sử dụng bằng các thủ thuật làm tròn, hiệu chỉnh từ các giá trị liên tục xác định được trên miền liên tục. Chính vì thế chúng không đem lại hiệu quả cao nhất đối với các bài toán có biến dạng

rời rạc hoặc có biến dạng hỗn hợp cả liên tục và rời rạc. Do vậy, việc xây dựng và áp dụng các thuật toán metaheuristic hiệu quả chuyên biệt cho các biến thiết kế dạng rời rạc thu hút được sự quan tâm rất lớn trong thời gian gần đây.

Tiến hóa vi phân (Differential Evolution-DE) là một thuật toán metaheuristic cho biến thiết kế liên tục khá đơn giản, dễ hiểu và dễ sử dụng nhưng lại rất mạnh mẽ và hiệu quả được Storn và Price đề xuất năm 1995 [6]. Đây là một loại thuật toán tiến hóa (Evolutionary Algorithms - EA) với nền tảng cơ bản là kỹ thuật tìm kiếm ngẫu nhiên dựa trên một tập hợp các ứng viên tiềm năng (quần thể) và sử dụng các toán tử đột biến, lai tạo và lựa chọn ở mỗi thế hệ để tìm kiếm kết quả tối ưu. Bằng cách sử dụng sơ đồ đột biến đơn giản và hiệu quả này, DE đã cho thấy hiệu suất rất hứa hẹn trong các bài toán tối ưu hóa số [7-8] và thu được một loạt thành tích khá tốt trong các cuộc thi tối ưu hóa với tham số thực [9-10]. Từ sự thành công của DE đối với biến liên tục, ngày càng nhiều nghiên cứu nhằm mở rộng DE sang biến thiết kế rời rạc. Nhìn chung, các phương pháp này có thể chia thành 3 dạng. Dạng thứ nhất là thực hiện chuyển đổi không gian dựa trên cách tiếp cận ngắt quãng miền liên tục. Trong cách tiếp cận này, các cá thể mới đầu tiên vẫn được xác định dựa theo các kỹ thuật trong DE thông thường như là một biến liên tục nhưng sẽ được chuyển đổi sang miền rời rạc trước khi hàm mục tiêu được tính toán [11-12]. Cách tiếp cận này là hợp lý nếu các biến rời rạc là dạng số thứ tự hoặc một thứ tự nhất định giữa các biến có thể được thiết lập. Nhưng đối với các vấn đề liên quan đến các biến phân loại rời rạc, không tồn tại thứ tự ngầm định giữa các biến, chúng không còn hiệu quả nữa. Cách tiếp cận thứ hai là định nghĩa mỗi cá thể như là một hoán vị của các số [13-14]. Các cá thể mới được tạo ra trên cơ sở tạo ra một vec-tơ hoán vị dựa trên các toán tử hoán đổi từ các cá thể hiện tại trong quần thể. Tuy nhiên cách tiếp cận này dĩ chệch khỏi ý tưởng cơ bản của thuật toán DE ban đầu bởi đã thêm một vec-tơ vi phân vào vec-tơ cơ sở của cá thể mục tiêu. Cách tiếp cận thứ ba là kết hợp thêm một số thuật toán tìm kiếm địa phương vào thuật toán DE truyền thống. Nhìn chung, các cách tiếp cận trên đều dựa trên nguyên tắc cơ bản của DE về đột biến, lai tạo và lựa chọn. Tuy nhiên,

chúng lại không tuân thủ một cách chính xác và do đó làm cho lợi thế của DE trong việc tìm kiếm giải pháp tối ưu trong miền tìm kiếm liên tục không được vận dụng triệt để khi chuyển đổi sang miền tìm kiếm rời rạc.

Gần đây, một phương pháp chuyển đổi DE để áp dụng cho biến rời rạc là vận dụng kỹ thuật đại diện dựa trên tập hợp (set-based DE) (S-DE) [15]. Trong S-DE, miền giá trị của hàm tương ứng với biến liên tục được xem là một tập phổ quát, các giải pháp sẽ tương ứng với một tập con cụ thể của tập phổ quát đó. Trong quá trình tối ưu, tại mỗi bước lặp, các cá thể hiện tại của quần thể và các cá thể thử nghiệm vẫn được đảm bảo là các giải pháp khả thi trong tập phổ quát trong khi các cá thể tạo ra bởi kỹ thuật đột biến là các biến trung gian thay vì các giải pháp khả thi. Tất cả các toán tử liên quan đến kỹ thuật đột biến đều được định nghĩa lại dựa trên các tập con xác định của các cá thể nhằm cho phép S-DE vẫn giữ lại các đặc tính của DE và qua đó giữ được khả năng tìm kiếm của nó. Các kết quả tính toán thu được trong [15] cho thấy S-DE có các ưu điểm vượt trội so với các thuật toán khác và là một thuật toán hứa hẹn để giải quyết các bài toán tối ưu với biến rời rạc. Tuy nhiên, ứng dụng của S-DE mới dừng lại ở tối ưu các hàm toán học chuẩn. Việc đánh giá tính hiệu quả của S-DE đối với các bài toán thiết kế thực tế như bài toán tối ưu hóa giàn thép trong nghiên cứu này là hết sức cần thiết. Tuy nhiên S-DE hội tụ quá nhanh dẫn đến kết quả tối ưu tìm được không tốt bằng các phương pháp khác.

Kết cấu giàn thép được sử dụng rất phổ biến trong các công trình xây dựng nhờ hình thức đẹp, trọng lượng nhẹ, khả năng vượt nhịp lớn và tối ưu hóa được sự làm việc của vật liệu. Trong thiết kế kết cấu giàn, nhờ sự phát triển của khoa học máy tính, các phương pháp thiết kế trực tiếp ngày càng được sử dụng nhiều hơn nhằm mô tả chính xác hơn ứng xử của kết cấu. Ưu điểm chính của các phân tích trực tiếp là ứng xử của kết cấu được tính toán theo các bước tải trọng nhỏ một cách liên tục và qua đó ứng xử phi tuyến tính phi toàn hồi của toàn bộ kết cấu theo tải trọng sẽ được ghi nhận. Từ đó, khả năng chịu tải của toàn bộ công trình được xác định. Sự an toàn của công trình sẽ được đánh giá khá đơn giản thông qua việc so sánh giữa khả năng chịu tải của cả công trình và áp lực do tải trọng gây ra. Quá trình kiểm tra từng cấu kiện riêng

lẽ như trong các phương pháp thiết kế truyền thống hoàn toàn được loại bỏ. Ứng dụng các phương pháp phân tích trực tiếp trong thiết kế công trình giàn đã được công bố trong một số nghiên cứu gần đây [16 –21].

Trong nghiên cứu này, tác giả trình bày bài toán tối ưu khối lượng của giàn thép với các biến thiết kế dạng rời rạc. Ứng xử phi tuyến tính phi đàn hồi của kết cấu được tính toán thông qua phân tích trực tiếp. Một thuật toán nâng cấp từ S-DE bằng cách tích hợp một kỹ thuật tìm kiếm địa phương hiệu quả được sử dụng cho quá trình tối ưu. Thuật toán đề xuất được đặt tên là ISDE (Improved Set-based Differential Evolution). Biến thiết kế là tiết diện của các thanh giàn. Điều kiện ràng buộc bao gồm các điều kiện theo các tổ hợp tải trọng cường độ và sử dụng trong tiêu chuẩn AISC-LRFD [22]. Các nội dung tiếp theo của bài báo được cấu trúc như sau: Phần hai trình bày thiết lập bài toán tối ưu giàn thép sử dụng phân tích trực tiếp, phần ba giới thiệu về thuật toán DE, S-DE và ISDE, phần bốn là kết quả tính toán tối ưu một giàn thép không gian 72 thanh và cuối cùng là kết luận.

2. Thiết lập bài toán tối ưu khối lượng giàn thép sử dụng phân tích trực tiếp [21]:

Lựa chọn hàm mục tiêu của bài toán là tổng khối lượng của kết cấu, và được tối thiểu hóa theo phương trình sau:

$$Min W(\mathbf{Y}) = \rho \sum_{i=1}^d \left(\gamma_i \sum_{j=1}^{d_i} L_{ij} \right) \tag{1}$$

trong đó: $\mathbf{Y} = (\gamma_1, \gamma_2, \dots, \gamma_d)$ véc tơ biến thiết kế (lần lượt là diện tích tiết diện của các thanh dàn); ρ - khối lượng riêng của vật liệu, d - số lượng biến thiết kế; d_i - số thanh dàn trong nhóm phần tử thanh thứ i ; L_{ij} - chiều dài của thanh dàn thứ j trong nhóm

$$W_{uncstr}(\mathbf{Y}) = (1 + \alpha_{str}\beta_1 + \alpha_{disp}\beta_2 + \alpha_{fre}\beta_3) \times \rho \sum_{i=1}^d \left(\gamma_i \sum_{j=1}^{d_i} L_{ij} \right) \tag{5}$$

trong đó: $\beta_1 = \sum (\max(C_k^{str}, 0))$; $\beta_2 = \sum \left(\sum_{j=1}^{nn} \max(C_{j,l}^{disp}, 0) \right)$; $\beta_3 = \sum \left(\sum_{j=1}^{nm} \max(C_{j,m}^{fre}, 0) \right)$ (6)

với α_{str} , α_{disp} và α_{fre} - các tham số phạt tương ứng với các điều kiện ràng buộc về cường độ, chuyển vị và tần số dao động riêng. Công thức (5) cho thấy rằng nếu một thiết kế mà vi phạm điều kiện ràng buộc thì hàm mục tiêu tương ứng sẽ được

phần tử thứ i . Biến thiết kế γ_i - được chọn từ một tập hợp các giá trị rời rạc cho trước trong các bài toán có biến thiết kế rời rạc.

Đối với tổ hợp trạng thái giới hạn cường độ, bằng việc sử dụng phân tích trực tiếp cho phép tính toán khả năng chịu tải của cả công trình, điều kiện ràng buộc được thể hiện bằng công thức (2):

$$C_k^{str} = 1 - \frac{R_k}{S_k} \leq 0 \tag{2}$$

trong đó: R_k - khả năng chịu tải của kết cấu đối với tổ hợp tải trọng thứ k và S_k - hiệu ứng do tổ hợp tải trọng cường độ thứ k gây ra.

Đối với tổ hợp trạng thái giới hạn sử dụng, điều kiện về chuyển vị sẽ được xem xét thông qua công thức (3).

$$C_{j,l}^{disp} = \left| \frac{\Delta_{j,l}}{\Delta_{j,l}^u} \right| - 1 \leq 0, \quad j = 1, \dots, nn \tag{3}$$

trong đó: nn - số nút dàn được xét điều kiện chuyển vị, $\Delta_{j,l}$ và $\Delta_{j,l}^u$ - chuyển vị và giới hạn chuyển vị của nút thứ j tương ứng với tổ hợp trạng thái giới hạn sử dụng thứ l .

Điều kiện ràng buộc về tần số dao động riêng của kết cấu được thể hiện như (4).

$$C_m^{fre} = \frac{f_{j,m}}{f_{j,m}^u} - 1 \leq 0, \quad j = 1, \dots, nm \tag{4}$$

trong đó: nm - số tần số dao động riêng được xét đến, $f_{j,m}$ và $f_{j,m}^u$ - tần số dao động riêng thứ j của kết cấu và giá trị cho phép của nó.

Đối với bài toán tối ưu có điều kiện ràng buộc ở trên, nghiên cứu này sử dụng phương pháp hàm phạt làm kỹ thuật để xử lý các điều kiện ràng buộc thông (áp dụng các thuật toán metaheuristic). Đây là kỹ thuật khá đơn giản và hiệu quả tốt cho hầu hết các loại ràng buộc khác nhau. Khi đó, hàm mục tiêu của bài toán được viết lại như sau:

cộng thêm một giá trị gọi là giá trị phạt tương ứng cho vi phạm đó. Do quá trình tối ưu là tối thiểu hóa hàm mục tiêu, các thiết kế vi phạm điều kiện ràng buộc sẽ dần dần bị loại bỏ. Giá trị của các tham số phạt này không phụ thuộc vào bài toán tối ưu, tuy

nhiên thường được lấy giá trị đủ lớn nhằm loại bỏ các thiết kế bị vi phạm và chỉ còn lại các thiết kế thỏa mãn tất cả các điều kiện ràng buộc. Trong nghiên cứu này, các tham số phạt được lấy bằng 10.000.

3. Thuật toán tiến hóa vi phân cho bài toán tối ưu với biến rời rạc (S-DE)

3.1 Thuật toán tiến hóa vi phân cơ bản

Thuật toán tối ưu DE được Storn và Price giới thiệu lần đầu tiên vào năm 1995 cho các bài toán tối ưu với biến thiết kế là liên tục [6]. Các cơ chế chính được xây dựng cho DE như sau:

a) Khởi tạo: NP cá thể $\mathbf{X}_i = (x_j) (j = 1, \dots, D)$ được lựa chọn một cách ngẫu nhiên từ miền giá trị cho trước của các biến thiết kế.

$$u_j = \begin{cases} v_j & \text{khi } (rand(0,1) < CR) \text{ hoac } (j = I) \\ x_{ij} & \end{cases} \tag{9}$$

d) Lựa chọn: \mathbf{U} sẽ được lựa chọn thay thế cho vị trí của \mathbf{X}_i trong quần thể mới nếu giá trị hàm mục tiêu của nó tốt hơn của \mathbf{X}_i .

3.2 Thuật toán tiến hóa vi phân S-DE cho biến thiết kế rời rạc

S-DE được xây dựng dựa trên cơ sở mở rộng các tính năng tìm kiếm ban đầu của DE trên miền liên tục sang miền rời rạc mà không làm thay đổi quy trình cơ bản của DE. Để thực hiện điều này, trong S-DE, các toán tử liên quan đến kỹ thuật đột biến và lai tạo được định nghĩa lại trên cơ sở sự vận hành của nó trên các tập con của tập phổ quát ban đầu. Các định nghĩa toán tử mới được quy định như sau:

- Định nghĩa 1: Hiệu của 2 tập cá thể \mathbf{X}_1 và \mathbf{X}_2

$$\hat{d}_1^j + \hat{d}_2^j = (e | e \in \hat{d}_1^j \text{ hoac } e \in \hat{d}_2^j), j = 1, \dots, D \tag{12}$$

- Định nghĩa 4: Tổng của một cá thể \mathbf{X} với \hat{d} :

$$x^j + \hat{d}^j = \begin{cases} x^j & \text{khi } \hat{d}^j = \emptyset \\ \hat{d}^j & \text{TH khác} \end{cases}, j = 1, \dots, D \tag{13}$$

3.3 Đề xuất thuật toán S-DE cải tiến cho biến thiết kế rời rạc

S-DE có ưu điểm là không làm thay đổi quy trình cơ bản của DE. Tuy nhiên, khả năng tìm kiếm của nó thường khá hạn chế do tốc độ hội tụ quá

b) Đột biến: Tương ứng với mỗi cá thể \mathbf{X}_i , một cá thể đột biến $\mathbf{V} = (v_1, v_2, \dots, v_D)$ được tạo ra dựa trên kỹ thuật đột biến DE. Một số kỹ thuật đột biến của DE thường được sử dụng là:

$$DE/rand/1: \mathbf{V} = \mathbf{X}_{r_1} + F \times (\mathbf{X}_{r_2} - \mathbf{X}_{r_3}) \tag{7}$$

$$DE/best/1: \mathbf{V} = \mathbf{X}_{best} + F \times (\mathbf{X}_{r_1} - \mathbf{X}_{r_2}) \tag{8}$$

trong đó: F - biên độ đột biến; \mathbf{X}_{best} - cá thể tốt nhất trong quần thể hiện tại; r_1, r_2 và r_3 - 3 số tự nhiên ngẫu nhiên được lựa chọn trong khoảng $[1, D]$ và thỏa mãn điều kiện $i \neq r_1 \neq r_2 \neq r_3$.

c) Lai tạo: Một cá thể mới $\mathbf{U} = (u_1, u_2, \dots, u_D)$ được tạo ra thông qua việc lai tạo giữa \mathbf{X}_i và \mathbf{V} thông qua cơ chế như sau:

được xác định như sau:

$$d^j = x_1^j - x_2^j = \{e | e \in x_1^j \text{ va } e \notin x_2^j\} \tag{10}$$

Trong trường hợp nếu A và B có cùng một phân tử giống nhau nào đó thì hiệu của 2 tập này sẽ là tập rỗng tương ứng với phân tử đó.

- Định nghĩa 2: Tích của biên độ đột biến và tập d được xác định như sau:

$$\hat{d}^j = F \times d^j = \begin{cases} d^j & \text{khi } r < F \\ \emptyset & \text{khi } r \geq F \end{cases} \tag{11}$$

trong đó: r - một giá trị ngẫu nhiên trong khoảng $[0, 1]$.

- Định nghĩa 3: Tổng của 2 tập đã được thu/phóng \hat{d}_1 và \hat{d}_2 được xác định như sau:

nhanh khiến cho kết quả tìm kiếm được thường là giá trị cục bộ địa phương. Để giải quyết vấn đề này, trong bài báo này tác giả đề xuất sử dụng kỹ thuật tìm kiếm địa phương vào S-DE và được đặt tên là ISDE (Improved Set-based Differential Evolution).

Nội dung kỹ thuật này như sau:

Sau khi cá thể đột biến $\mathbf{V} = (V_1, V_2, \dots, V_D)$ được tạo ra dựa trên kỹ thuật cơ bản của S-DE, cá thể đột biến $\mathbf{V}' = (V_1', V_2', \dots, V_D')$ được tạo ra như sau:

$$V_i' = \begin{cases} V_i + \text{rand} \cdot \text{int}[0,3] & \text{khi } \text{rand} > 0.5 \\ V_i - \text{rand} \cdot \text{int}[0,3] & \text{khi } \text{rand} \leq 0.5 \end{cases} \quad (14)$$

Công thức (14) được giải thích như sau: tương ứng với mỗi thành phần V_i của \mathbf{V} , V_i' được tạo ra bằng cách dịch chuyển vị trí của V_i trong miền tìm kiếm lên trước hoặc về sau một số bước là số tự nhiên ngẫu nhiên trong đoạn $[0,3]$. Điều này giúp cho khả năng tìm kiếm địa phương của thuật toán được tăng lên, các cá thể tốt xung quanh cá thể đột biến có cơ hội được xét đến trong quá trình tối ưu. Qua đó, kết quả tối ưu tìm được sẽ tốt hơn và vượt qua được hội tụ cục bộ địa phương của S-DE. Lưu ý rằng, người dùng hoàn toàn có thể sử dụng giá trị khác thay cho 3 trong đoạn $[0,3]$ ở trên. Khi giá trị này tăng lên sẽ giúp cho khả năng tìm kiếm cục bộ của chương trình tối ưu tốt lên. Tuy nhiên, điều này lại khiến cho tốc độ hội tụ bị giảm xuống và sẽ tiêu tốn nhiều thời gian hơn cho việc chạy chương trình tối ưu. Trong một số bài toán tối ưu, khi hạn chế về nguồn lực tính toán thì giá trị lựa chọn này nên nhỏ nhằm thúc đẩy nhanh quá trình hội tụ của chương trình. Trong bài báo này, tác giả lựa chọn bằng 3 dựa trên sự khuyến nghị của tác giả Degertekin [23] và kinh nghiệm của bản thân.

4. Trường hợp nghiên cứu

Một giàn thép không gian 72 thanh được lựa chọn nghiên cứu để đánh giá hiệu quả của phương pháp tối ưu giàn thép sử dụng thuật toán S-DE và phân tích trực tiếp. Kích thước hình học của giàn thép được thể hiện trong hình 1. Thanh được làm từ thép A992 với khối lượng riêng là $7850 \text{ (kg/m}^3\text{)}$, mô-đun đàn hồi $E=200 \text{ (Gpa)}$ và cường độ chảy của vật liệu $F_y=345 \text{ (Mpa)}$. Tải trọng gió (W) bằng 150 (kN) được đưa về các nút giàn theo phương trục X như trên hình 1. Tải trọng bản thân (D) bằng 200 (kN) và hoạt tải (L) bằng 120 (kN) được đưa về tải trọng thẳng đứng đặt tại các nút giàn. Diện tích tiết diện các thanh được chia thành 16 biến thiết kế như sau: Nhóm 1: Tiết diện thanh số 1-4; nhóm 2: 5-12; nhóm 3: 13-16; nhóm 4: 17-18; nhóm 5: 19-22; nhóm 6: 23-30; nhóm 7: 31-34; nhóm 8: 35-36; nhóm 9: 37-

40; nhóm 10: 41-48; nhóm 11: 49-52; nhóm 12: 53-54; nhóm 13: 55-58; nhóm 14: 59-66; nhóm 15: 67-70; nhóm 16: 71-72. Diện tích tiết diện các thanh được lựa chọn từ tập rời rạc sau: List(42)=[1.62, 1.80, 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, 11.5, 13.5, 13.9, 14.2, 15.5, 16.0, 16.9, 18.8, 19.9, 22.0, 22.9, 26.5, 30.0, 33.5] (in^2). Các tổ hợp tải trọng được xem xét là: 2 tổ hợp cường độ: (1.2D+1.6L, 1.2D+0.5L+1.7W) và 1 tổ hợp sử dụng với giới hạn chuyển vị lệch tầng bằng H/400: (1.0D+0.5L+0.7W).

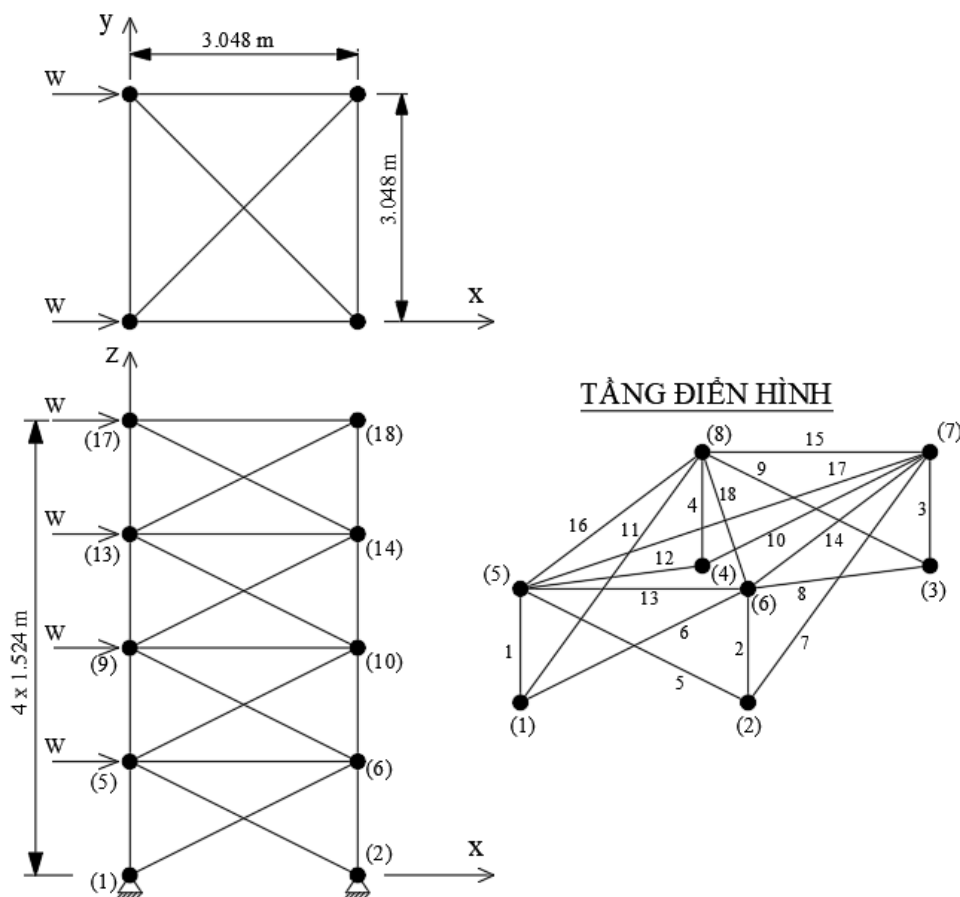
Các thông số sử dụng trong chương trình tối ưu là: Số các thể trong quần thể: DEpop = 25, số vòng tiến hóa: Maxltr = 1000, F = 0,7, CR = 0,6. Tập hợp 42 diện tích thanh giàn trong List được đánh số thứ tự từ 1 đến 42. Các giá trị của biến tối ưu trong chương trình sẽ nhận các giá trị tự nhiên từ 1 đến 42. 3 kỹ thuật tối ưu dựa trên nền tảng DE được nghiên cứu và so sánh là: (1) ISDE được xây dựng trong bài báo này; (2) S-DE; (3) các cá thể mới đầu tiên vẫn được xác định dựa theo các kỹ thuật trong DE thông thường như là một biến liên tục nhưng sẽ được chuyển đổi sang miền rời rạc trước khi hàm mục tiêu được tính toán, đặt tên là D-DE (Discrete DE). D-DE có thể được giải thích cụ thể hơn như sau: Đầu tiên, thành phần v_i của \mathbf{V} vẫn được xác định một cách thông thường dựa trên các kỹ thuật đột biến và lai ghép của thuật toán DE. Nhưng nhiên giá trị v_i hầu như là một số thực không nằm trong tập dữ liệu rời rạc giành cho v_i . Do vậy, trong bước thứ hai, v_i sẽ được nhận giá trị nằm trong tập dữ liệu rời rạc cho trước gần với giá trị hiện tại của v_i nhất.

Kết quả tối ưu được trình bày trong bảng 1. Mỗi thuật toán được chạy độc lập 10 lần để giảm thiểu tác động của tính chất ngẫu nhiên đến kết quả tối ưu tìm được. Dựa trên kết quả trình bày trong bảng 1, ta có thể thấy ISDE luôn tìm được thiết kế tối ưu tốt nhất trong tất cả các lần chạy. Thiết kế tối ưu tốt nhất này tương ứng với khối lượng giàn là 2325,9 (kg). Trong khi đó, D-DE và S-DE không thể tìm được kết quả tối ưu tốt nhất này qua 10 lần chạy khác nhau. Khối lượng tối ưu tốt nhất mà D-DE tìm được là 2384,4 (kg) và giá trị kém nhất tìm được là 2421,3 (kg). Kết quả tối ưu sử dụng thuật toán S-DE

KẾT CẤU - CÔNG NGHỆ XÂY DỰNG

tỏ ra khá kém với khối lượng tốt nhất là 2857,5 (kg) và khối lượng kém nhất là 3207,5 (kg). Kết quả này

cho thấy ISDE tỏ ra mạnh mẽ và ổn định hơn hẳn D-DE và S-DE.



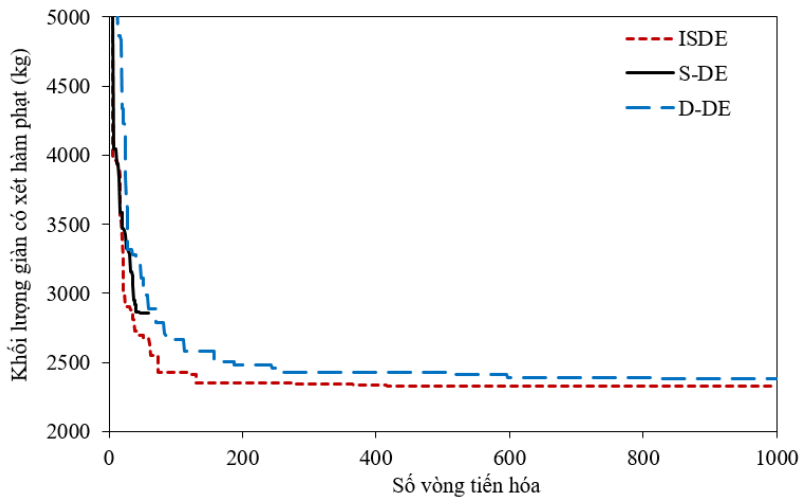
Hình 1. Giàn thép không gian 72 thanh

Bảng 1. Kết quả chạy tối ưu giàn 72 thanh

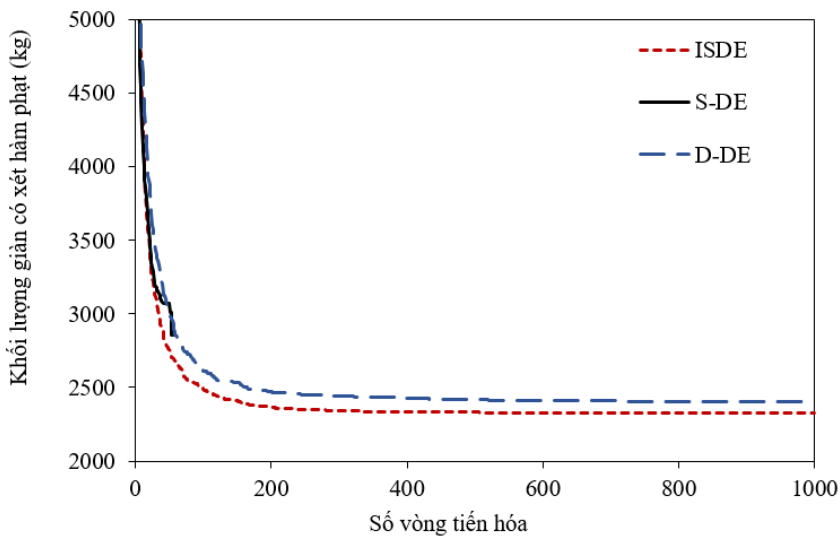
	ISDE	S-DE	D-DE
Khối lượng giàn tối ưu tốt nhất (kg)	2325,90	2857,50	2384,40
Khối lượng giàn tối ưu kém nhất (kg)	2325,90	3207,50	2421,30
Khối lượng trung bình các kết quả tối ưu (kg)	2325,90	2857,50	2402,60
Độ lệch chuẩn khối lượng các kết quả tối ưu (kg)	0,00	100,27	11,55
Tiết diện các thanh giàn của thiết kế tối ưu tốt nhất (in ²)	11,5; 1,80; 1,62; 1,62; 7,22; 1,62; 1,62; 1,62; 3,09; 1,62; 1,62; 1,62; 1,62; 1,62; 1,62; 1,62	11,5; 2,13; 2,13; 3,38; 7,22; 1,62; 3,38; 2,62; 3,09; 2,13; 2,62; 2,93; 1,62; 1,80; 1,80; 1,62	11,5; 1,80; 1,62; 1,62; 7,22; 1,62; 1,62; 1,80; 3,13; 1,62; 1,80; 1,80; 1,80; 1,80; 1,62; 1,62
Đánh giá vi phạm các điều kiện ràng buộc	Không vi phạm	Không vi phạm	Không vi phạm

Hình 2 và hình 3 thể hiện đường cong hội tụ của lần chạy tối ưu tốt nhất và giá trị trung bình các lần chạy của các thuật toán. Ta có thể nhận thấy rằng S-DE hội tụ quá sớm (khoảng 50 vòng tiến hóa/1000 vòng được thiết lập). Điều này lý giải cho việc kết quả tối ưu tìm được khi sử dụng thuật toán S-DE khá

kém. Trong khi đó, ISDE có tốc độ hội tụ tốt nhất trong 3 thuật toán. Đồng thời ISDE cho phép tìm được kết quả tối ưu tốt hơn so với 2 thuật toán còn lại. Thuật toán D-DE có tốc độ hội tụ kém nhất trong 3 thuật toán. Tuy nhiên, D-DE có khả năng xử lý hội tụ địa phương khá tốt, đặc biệt là so với S-DE.



Hình 2. Đường cong hội tụ của lần chạy tối ưu tốt nhất



Hình 3. Đường cong hội tụ trung bình của tất cả các lần chạy tối ưu

5. Kết luận

Bài báo đề xuất một thuật toán tối ưu dựa trên thuật toán tiến hóa vi phân và nguyên lý bộ giá trị (Set-based Differential Evolution, S-DE) nhằm tối ưu hóa giàn thép phi tuyến có biến thiết kế rời rạc. Trong thuật toán được đề xuất (ISDE), kỹ thuật tìm kiếm địa phương được tích hợp trong S-DE giúp cho ISDE có khả năng tìm kiếm tốt hơn và tránh được hạn chế hội tụ quá nhanh của S-DE. Ứng xử phi tuyến hình học và vật liệu của giàn thép được xét đến thông qua sử dụng phân tích trực tiếp phi tuyến tính phi đàn hồi. Kết quả nghiên cứu giàn không gian 72 thanh với 16 biến thiết kế rời rạc cho thấy rằng ISDE tỏ ra mạnh mẽ vượt trội so với S-DE và D-DE. Điều này thể hiện qua việc ISDE luôn tìm được kết quả tối ưu tốt nhất trong mọi lần chạy trong khi S-DE và D-DE thì không. Tốc độ hội tụ của ISDE cũng tốt hơn so với S-DE và D-DE. Các hướng nghiên cứu tiếp theo là: (1) Đánh giá hiệu

quả của ISDE trên nhiều dạng kết cấu giàn khác nhau cùng với các điều kiện về dạng biến thiết kế rời rạc khác nhau; (2) Nghiên cứu mở rộng hiệu quả của ISDE đối với các bài toán tối ưu khác.

TÀI LIỆU THAM KHẢO

1. M.H. Ha, Q.A. Vu, V.H. Truong (2018). Optimum Design of Stay Cables of Steel Cable-stayed Bridges Using Nonlinear Inelastic Analysis and Genetic Algorithm. *Structures*; 16: 288-302.
2. V.H. Truong, S.E. Kim (2018). A robust method for optimization of semi-rigid steel frames subject to seismic loading. *Journal of Constructional Steel Research*; 145C: 184-195.
3. H.A. Pham, D.X. Nguyen, V.H. Truong (2021). An efficient differential-evolution-based moving compensation optimization approach for controlling differential column shortening in tall buildings. *Expert Systems with Applications* 169, 114531.

4. M.H. Ha, Q.V. Vu, V.H. Truong (2020). Optimization of nonlinear inelastic steel frames considering panel zones. *Advances in Engineering Software*; 142: 102771.
5. V. H. Truong, S.E. Kim (2017). An efficient method for reliability-based design optimization of nonlinear inelastic steel space frames. *Struct Multidisc Optim*; 56: 331-351.
6. R. M. Storn and K. V. Price (1997). Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim*, vol. 11, pp. 341–359.
7. A. K. Qin, V. L. Huang and P. N. Suganthan (2009). Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans. Evol. Comput.*, vol. 13, pp. 398-417.
8. Y. Wang, Z. Cai, Q. Zhang (2011). Differential evolution with composite trial vector generation strategies and control parameters. *IEEE Trans. Evol. Comput.*, vol. 15, pp. 55-66.
9. R. Storn and K. V. Price (1996). Minimizing the real functions of the ICEC'96 contest by differential evolution. In *Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 842-844.
10. K. V. Price (1997). Differential evolution versus the functions of the 2nd ICEO. In *Proceedings of IEEE International Conference on Evolutionary Computation*, Apr.
11. J. Zhang, Viswanath Avasarala, Arthur C. Sanderson and Tracy Mülle (2008). Differential Evolution for Discrete Optimization: An Experimental Study on Combinatorial Auction Problems. In *Proceedings of IEEE Conference on Evolutionary Computation*.
12. N. Damak, B.Jarboui, P. Siarryb and T. Louki (2009). Differential evolution for solving multi-mode resource-constrained project scheduling problems. *Computers & Operations Research*, vol. 36, pp. 2653-2659.
13. Q.-K. Pan, M. Fatih Tasgetiren and Yun-Chia Liang (2007). A Discrete Differential Evolution Algorithm for the Permutation Flowshop Scheduling Problem. *Presented at the Genetic Evol. Comput*.
14. M. F. Tasgetiren, P.N. Suganthan and Quan-Ke Pan (2010). An ensemble of discrete differential evolution algorithms for solving the generalized traveling salesman problem. *Applied Mathematics and Computation*, vol. 215, pp. 3356–3368.
15. Y. Liu, W. N. Chen, Z.H. Zhan, Y. Lin, Y.J. Gong, J. Zhang (2013). A set-based discrete differential evolution algorithm. *2013 IEEE International Conference on Systems, Man, and Cybernetics* 1347-1352.
16. V.H. Truong, Q.V. Vu, H.T. Thai, M.H. Ha (2020). A robust method for safety evaluation of steel trusses using Gradient Tree Boosting algorithm. *Advances in Engineering Software* 147, 102825.
17. H.M. Hùng, T.V. Hùng, Đ.V. Thuật, V.Q. Việt (2020). Phương pháp xử lý hiện tượng mô hình quá khớp trong xây dựng mô hình học sâu để ước lượng khả năng chịu tải của giàn phi tuyến. *Tạp chí Khoa học Công nghệ Xây dựng (KHCNXD)-ĐHXD* 14 (1V), 12-20.
18. V.H. Truong, S.E. Kim (2018). Reliability-based design optimization of nonlinear inelastic trusses using improved differential evolution algorithm. *Advances in Engineering Software*; 121: 59-74.
19. V.H. Truong, Q.V. Vu, V.T. Dinh (2019). A deep learning-based procedure for estimation of ultimate load carrying of steel trusses using advanced analysis. *Journal of Science and Technology in Civil Engineering (STCE)-NUCE*; 13(3): 113-123.
20. S.E. Kim, V.H. Truong (2020). Reliability Evaluation of Semirigid Steel Frames Using Advanced Analysis. *Journal of Structural Engineering*; 146(5): 04020064.
21. H.M. Hùng, T.V. Hùng, Đ.V. Thuật (2020). Bài toán tối ưu kết cấu dàn phẳng sử dụng phân tích trực tiếp có xét đến điều kiện ràng buộc về tần số dao động riêng. *Tạp chí Khoa học Công nghệ Việt Nam - Bộ Khoa học và Công nghệ*, 62(6) 6.2020, 24-28.
22. AISC-LRFD (1999), "Manual of steel construction – load and resistance factor design", *Chicago (IL): American Institute of Steel Construction*.
23. Degertekin SO (2008), "Optimum design of steel frames using harmony search algorithm": *Struct Multidiscip Optim* 36:393–401.

Ngày nhận bài: 20/4/2021.

Ngày nhận bài sửa: 28/5/2021.

Ngày chấp nhận đăng: 30/5/2021.