■Ngày nhận bài: 13/12/2024 ■Ngày sửa bài: 09/01/2025 ■Ngày chấp nhận đăng: 14/02/2025

Tích hợp phương pháp phần tử hữu hạn với các thuật toán tối ưu hóa đa mục tiêu để giải kết cấu dàn không gian

Integrating finite element method with multi-objective metaheuristic optimization algorithms for solving space truss structures

> KS NGUYỄN NGÔ CÔNG PHÁP, PGS.TS TRƯƠNG ĐÌNH NHẬT[2*], PGS. TS NGUYỄN HỮU ANH TUẤN[3], THS LÊ THỊ THÙY LINH[4]

[1]HVCH Ngành Kỹ thuật xây dựng, Trường ĐH Kiến trúc TP.HCM, Email: 22xd08@uah.edu.vn
[2]GV Khoa Xây dựng, Trường ĐH Kiến trúc TP.HCM,
[3]GV Khoa Xây dựng, Trường ĐH Kiến trúc TP. HCM, Email: tuan.nguyenhuuanh@uah.edu.vn
[4]GV Khoa Sư phạm Công nghiệp, ĐH Sư phạm Kỹ thuật, Đại học Đà Nẵng; Email: lttlinh@ute.udn.vn
[*]Corresponding author, Email: nhat.truongdinh@uah.edu.vn

TÓM TẮT

Tối ưu hóa kết cấu công trình luôn là đề tài được quan tâm trong quá trình thiết kế của các kỹ sư xây dựng. Nghiên cứu này trình bày các bước để thực hiện thông qua việc áp dụng phương pháp phần tử hữu hạn kết hợp với các thuật toán tối ưu hóa phổ biến hiện nay. Việc so sánh mức độ hiệu quả của các thuật toán NSGA-II, MOPSO, MOEA/D và MOJS sẽ được khảo sát trên các ví dụ công trình điển hình. Kết quả cho thấy thuật toán MOJS cho kết quả tính toán ổn định và nhanh hơn các thuật toán còn lại. Kết quả nghiên cứu sẽ đề xuất thêm cho các kỹ sư thiết kế nhiều sự lựa chọn và cân nhắc khi lựa chọn thuật toán tối ưu hóa kết cấu công trình

Từ khóa: Đa mục tiêu; tối ưu kết cấu; kết cấu thép; MOJS; Meta-heuristic.

ABSTRACT

Structural optimization has always been a topic of interest in the design process of construction engineers. This study outlines the steps to implement the process by applying the finite element method combined with currently popular optimization algorithms. The effectiveness of the NSGA-II, MOPSO, MOEA/D, and MOJS algorithms will be examined using representative structural examples. The results indicate that the MOJS algorithm provides more stable and faster computational performance than the other algorithms. The findings of this study offer engineers additional options and considerations when selecting optimization algorithms for structural design

Keywords: Multi objective; structural optimization; steel structure; MOJS; Meta-heuristic.

1. GIỚI THIỆU

Việc tối ưu hóa trong thiết kế công trình thép thu hút sự quan tâm ngày càng lớn khi tổng giá thành hay khối lượng của công trình thường được tối thiểu hóa dựa trên biến thiết kế là diện tích tiết diện điển hình của các cấu kiện [1]. Không chỉ dừng lại ở các mục tiêu đơn lẻ, các bài toán tối ưu hóa thực tế phải giải quyết nhiều vấn đề cùng lúc [2]. Trong nhiều trường hợp, việc tìm giá trị tối ưu của các yếu tố liên quan là rất khó khăn, vì vậy không thể sử dụng phương pháp chính xác mà phải áp dụng metaheuristic - một giải pháp cao cấp hơn cho bài toán tìm kiếm hoặc tối ưu hóa vấn đề.

Trong vài thập kỷ vừa qua, nhiều thuật toán metaheuristic đã được phát triển và chúng được lấy cảm hứng từ các quá trình tự nhiên hoặc con người. Những thuật toán này được phân loại theo mô hình lấy cảm hứng từ thiên nhiên tương ứng, ví dụ có thể kể đến như mô hình lấy cảm hứng cho các thuật toán tối ưu hóa bầy đàn (PSO - Particle swarm optimization), tối ưu hóa đàn kiến (ACO - Ant Colony Optimization), thuật toán đàn ong nhân tạo (ABC - Artificial Bee Colony), thuật toán tìm kiếm sứa (JS - Jellyfish Search).

Đi cùng với việc sáng tạo ra các thuật toán thì việc đối chiếu với các thuật toán khác để kiểm tra tính hiệu quả của từng thuật toán cũng là việc tất yếu. Trong quá trình so sánh, các tác giả cũng đề xuất những thuật toán nâng cấp để nâng cao hiệu quả tối ưu, đồng thời giảm thời gian tìm được kết quả. Năm 2016, I. Aydogdu, A. Akin và M. P. Saka [3] đề xuất thuật toán LFABC (Artificial bee colony algorithm with Levy flight distribution) khi so sánh với các thuật toán S-ABC, DHS và ACO, các tác giả nhận ra khi áp dụng cả bốn thuật toán trên mô hình khung thép không gian 8 tầng, 1024 thanh thì thuật toán LFABC cho kết quả nhẹ hơn 25,2% so với thiết kế tối ưu được tìm thấy bởi thuật toán ACO.

Tiếp theo, năm 2020 và năm 2022, hai tác giả là Jui-Sheng Chou và Trương Đình Nhật [4, 5] đã đề xuất hai thuật toán MOJS (Multi-objective Jellyfish Search) và MOFBI (Multi-objective forensic-based investigation) để đối chiếu với các thuật toán khác Các kết quả cho thấy thuật toán MOJS vượt trội ở hầu kết các chỉ số khi thực hiện kiểm tra 20 tiêu chí khác nhau trên nhiều ví dụ giàn thép, điều tương tự với thuật toán MOFBI.

Hiện nay trong nước chưa có nhiều các nghiên cứu về tối ưu hóa đa mục tiêu, nhất là các bài toán tối ưu hóa kết cấu công trình. Các nghiên cứu tối ưu hóa phổ biến trong nước hiện nay xoay quanh việc áp dụng các thuật toán tối ưu hóa đơn lẻ để tối ưu hóa từ đơn mục tiêu sang đa mục tiêu [6-8], tuy nhiên các so sánh thuật toán vẫn chỉ dừng ở các ví dụ đơn mục tiêu. Vì vậy nghiên cứu này sẽ cung cấp góc nhìn về việc đối chiếu các thuật toán tối ưu hóa phổ biến hiện nay thông qua các bài toán đa mục tiêu.

2. CƠ SỞ KHOA HỌC

2.1. Phương pháp phần tử hữu hạn

Phương pháp phần tử hữu hạn (FEM) [9] được áp dụng để tính toán lực dọc trục (hoặc ứng suất phần tử) của kết cấu dàn 3D, được trình bày như một phương trình độ cứng tổng thể như công thức (1-3):

$$\{F\} = [K]\{d\} \quad (1)$$

Định luật Hooke tính toán ứng suất trục của mỗi phần tử (σ_x) như sau

$$\sigma_x = E\varepsilon_x \quad (2)$$

Do đó, lực trục của mỗi phần tử (T) được tính bằng công thức sau:

$$T = A\sigma_x \quad (3)$$

Trong đó: {F} là lực nút tổng thể, [K] là ma trận độ cứng tổng thể, {d} là chuyển vị nút tổng thể, E là mô đun đàn hồi, ε là biến dạng dọc trục, A là diện tích mặt cắt ngang

2.2. Các lý thuyết tối ưu hóa điển hình

Bài báo sẽ thực hiện so sánh 4 thuật toán tối ưu hóa sau: NSGA-II, MOPSO, MOEA/D và MOJS

2.2.1. Thuật toán NSGA-II

Thuật toán tối ưu NSGA-II được Kalyanmoy Deb và các cộng sự đề xuất vào năm 2002 [10]. Thuật toán NSGA-II áp dụng phương pháp phân loại không bị chi phối để thực hiện gán trị thích nghi. Một cá thể được xem là chiếm ưu thế so với cá thể khác nếu: 1- giải pháp của nó không kém hơn giải pháp của cá thể kia ở mọi mục tiêu, 2 - giải pháp của nó vượt trội hơn giải pháp của cá thể kia ở ít nhất một mục tiêu. Tất cả các cá thể không bị bất kỳ cá thể nào khác thống trị được xếp vào nhóm một, lần lượt đến các nhóm 2, 3… [11]

2.2.2. Thuật toán MOPSO

Thuật toán tối ưu MOPSO được Carlos A. Coello Coello và Maximino Salazar Lechuga đề xuất vào năm 2002 [12] lấy cảm hứng từ hành vi bầy đàn trong tự nhiên. Thuật toán mô phỏng hành vi xã hội của các bầy đàn điển hình là loài chim, bằng cách kết hợp kinh nghiệm bản thân và xã hội thì bầy chim học cách tiếp cận mục tiêu thức ăn của chúng. Chúng liên tục điều chỉnh vị trí bản thân trong bầy đàn để tối ưu hóa vị trí toàn cục, từ đó tự tổ chức lại thành một cấu trúc tối ưu. [13]

2.2.3. Thuật toán MOEA/D

Thuật toán tối ưu MOEA/D được Qingfu Zhang và Hui Li đề xuất vào năm 2007 [14], thuật toán dựa trên quá trình phân rã. Mục tiêu của bài toán tối ưu hóa thông qua MOEA/D là phân rã một bài toán đa mục tiêu (MOP) thành một số bài toán tối ưu hóa đơn mục tiêu thông qua các hàm tổng hợp và sử dụng các thuật toán tiến hóa khác để tối ưu chúng cùng một lúc. Vì các giải pháp tối ưu của mỗi bài toán con đã được chứng minh là pareto tốt nhất đối với bài toán đa mục tiêu ban đầu nên tập hợp các giải pháp tối ưu của các bài toán con được xem là xấp xỉ tốt của pareto tốt nhất [15].

2.2.4. Thuật toán MOJS

Thuật toán tối ưu MOJS được Jui-Sheng Chou và Trương Đình Nhật đề xuất vào năm 2020 [18]. Thuật toán lấy cảm hứng từ chuyển động của loài sứa biển - thứ bị chi phối bởi ngoại cảnh. Mặc dù có khả năng bơi nhưng sứa lại chủ yếu trôi trong nước, tùy thuộc vào dòng chảy và thủy triều, khi gặp điều kiện thuật lợi thì chúng kết thành đàn lớn và phát triển nhanh chóng. Vì là loài sinh vật bơi yếu nên hoạt động của chúng gắn liền với hải lưu, cùng với số lượng thức ăn tại các địa điểm sứa ghé thăm khác nhau, tất cả định hình được hướng di chuyển và tập trung tối ưu của loài sứa trên khắp đại dương [16].

2.3. Các phương pháp đánh giá

2.3.1. Chỉ số thể tích siêu hình (HV - Hypervolume index)

Thể tích của không gian mục tiêu được bao phủ bởi các thanh của tập hợp các giải pháp không bị chi phối (*A*) được gọi là chỉ số thể tích siêu hình (*HV*) [17]. Một hypercube v_i với một điểm tham chiếu được tạo ra về mặt toán học cho mỗi giải pháp $i \in A$. HV được tính bằng công thức (4):

$$HV = volum\left(\bigcup_{i=1}^{A} v_i\right) \quad (4)$$

HV cung cấp thông tin về cả sự hội tụ và sự đa dạng của tập A; một thuật toán tốt hơn có giá trị HV lớn hơn.

2.3.2. Chỉ số khoảng cách (SP - Spacing)

Chỉ số SP là phương sai của khoảng cách giữa các vector lân cận, như sau:

$$SP = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(\overline{d}' - d_i'\right)^2} \quad (5)$$

Trong đó:

$$d_i' = \underset{k \neq i}{k \in Ps} \min \sum_{m=1}^{M} \left|f_m^i - f_m^k\right| \quad (6)$$

SP biểu thị độ lệch chuẩn của khoảng cách giữa các giải pháp không bị chi phối i và k, Ps là tập hợp các giải pháp không bị chi phối, $\overline{d}'$ là giá trị trung bình của tất cả các d_i' và n là số lượng vector không bị chi phối đã được khám phá cho đến nay (công thức (5), (6)). Sự phân phối đồng đều của các giải pháp tương ứng với giá trị SP nhỏ hơn.

2.4. Áp dụng thuật toán tối ưu hóa để giải quyết vấn đề kết cấu

Các phương trình sau đây sử dụng để giải quyết bài toán tối ưu hóa đa mục tiêu của một kết cấu với nm phần tử (trong ng nhóm) và nm nút [18] (công thức (12-15)).

Tìm $A = \{A_1, A_2, \ldots A_{nm}\}$ sao cho

Tối thiểu hóa:

$$\begin{cases} f_1(A) = \sum_{i=1}^{nm} A_i \rho_i L_i \\ f_2(A) = \max\left(\left|\delta_j\right|\right) \end{cases} \quad (7)$$

Với điều kiện:

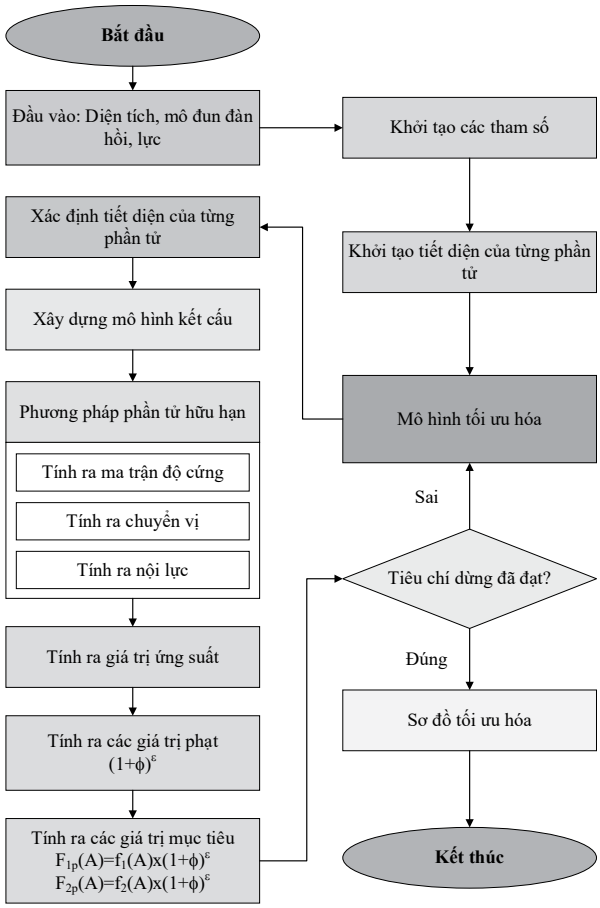
$$\sigma_{ci} \le \sigma_i \le \sigma_{ti}, \quad i = 1,2,\ldots nm \quad (8)$$

$$\delta_{min} \le \delta_j \le \delta_{max}, \quad j = 1,2,\ldots nm \quad (9)$$

$$A_{min} \le A_k \le A_{max}, \quad k = 1,2,\ldots ng \quad (10)$$

$$\lambda_i \le \lambda_{igh}, \quad k = 1,2,\ldots nm \quad (11)$$

Trong đó A là vector các biến kích thước, $f_1(A)$ là trọng lượng của kết cấu; $f_2(A)$ là độ lệch tối đa của nút trong kết cấu. δ_j là độ lệch của nút thứ j; ρ_i là mật độ vật liệu của phần tử thứ i; A_i là diện tích mặt cắt ngang của phần tử thứ i, có thể thay đổi từ A_{min} đến A_{max}, và λ_i là độ mảnh của phần thử thứ i, được giới hạn bởi λ_{igh}.



Hình 1. Khung tối ưu hóa metaheuristic để giải quyết các vấn đề thiết kế kết cấu

Degertekin và cộng sự đã đề xuất một hàm phạt F_p để xử lý các điều kiện ràng buộc trong bài toán tối ưu hóa việc tối thiểu hóa hàm mục tiêu f(A).

$$F_p = f(A) \times (1+\phi)^{\varepsilon} \quad (12)$$

Trong đó: ϕ là tổng các số phạt, được định nghĩa như sau:

$$\phi = \sum_{i=1}^{q} \phi_i \quad (13)$$

$$\phi_i = \left|1 - \frac{p_i}{p_i^*}\right| \quad (14)$$

Trong đó ϕ_i là mức độ vi phạm ràng buộc với giới hạn là p_i^* và q là số lượng ràng buộc hoạt động. Số mũ trong hàm phạt ε là hàm số biểu thị số lần lặp và được cho bởi:

$$\varepsilon = \varepsilon_o \left(1 + \frac{t}{Max_{iter}}\right) \quad (15)$$

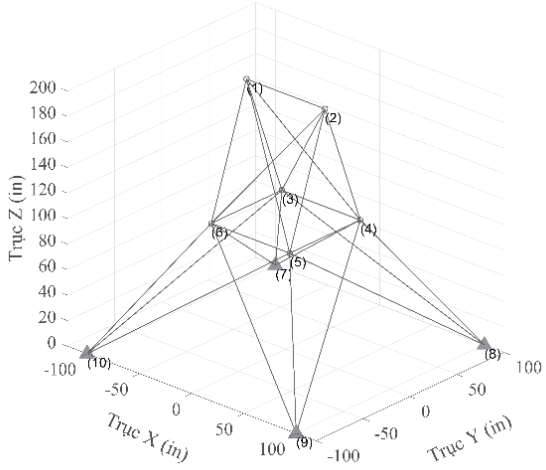
Giá trị ban đầu ε_0 có thể dược đặt từ 1,001 đến 10000 [19] và được đặt thành 2 trong trường hợp này.

Hình 1 mô tả quy trình thực hiện mô hình tối ưu hóa kết cấu từ lúc nhận được các tham số ban đầu đến lúc hoàn thành biểu đồ pareto tốt nhất. Trong nghiên cứu này, MOJS và các thuật toán khác sẽ được tính toán trên phần mềm MATLAB R2022b bằng máy tính với chip Intel(R) Core i7-7700HQ, tốc độ xung nhịp 2.80 GHz, RAM 16GB và Windows 10.

3. VÍ DỤ ĐIỂN HÌNH

3.1. Giàn 25 thanh

Bài toán tháp không gian 25 thanh được mô tả trong hình 2. Mật độ vật liệu, mô đun đàn hồi và giới hạn ứng suất của các thành phần trong kết cấu này tương ứng là 0.1 lb/in3, 10000 ksi và ±40000 psi. Tất cả các nút của kết cấu phải chịu giới hạn dịch chuyển là ±0.35 (in) theo các hướng X, Y và Z và tải trọng tại các nút được đặt thành P1x = 1 kips, P3x = 0.5 kips, P6x = 0.6 kips, P1y = P1z = P2y = P2z = -10 kips. Các thành phần của kết cấu được chia thành tám nhóm: (1) A1, (2) A2 - A5, (3) A6 - A9, (4) A10 - A11, (5) A12 - A13, (6) A14 - A17, (7) A18 - A21 và (8) A22 - A25. Các khu vực mặt cắt ngang rời rạc, bao gồm trong ví dụ với ví dụ tám biến tối ưu hóa, được chọn từ tập hợp D = {0.1, 0.2, 0.3… 2.3, 2.4, 2.6, 2.8, 3.0, 3.2, 3.4} (in^2) là diện tích cốt thép chịu kéo và x1 là diện tích tiết diện mặt cắt ngang) nên hai biến x6 và x9 sẽ bị loại ra trong khi xây dựng mô hình. [16]



Hình 2. Mô hình giàn 25 thanh

Bảng 1 thể hiện kết quả thống kê của giá trị HV. Qua đó cho thấy rằng thuật toán ngoại trừ MOEA/D cho kết quả trung bình HV thấp nhất thì 3 thuật toán còn lại đều có giá trị xấp xỉ nhau, tuy nhiên phương sai của MOJS là nhỏ nhất (std = 0.0061). Điều này cho thấy rằng MOJS có khả năng tìm ra đường pareto tốt nhất có phân bố cao nhất và ổn định nhất.

Bảng 1. Giá trị HV giải kết cấu 25 thanh

	Max	Min	Avg.	Std.
NSGA-II	0.9980	0.7853	0.7970	0.0380
MOPSO	0.9416	0.7777	0.7987	0.0377
MOJS	0.7877	0.7672	0.7804	0.0061
MOEA/D	0.9852	0.2103	0.4882	0.2072

Tiếp theo, bảng 2 thể hiện kết quả thống kê của giá trị SP. Kết quả phân tích cho thấy rằng MOJS là thuật toán duy nhất không ra kết quả SP = 0, đồng thời giá trị trung bình và phương sai đều nhỏ, vì thế phân phối kết quả các giá trị tìm ra của thuật toán MOJS là tốt nhất.

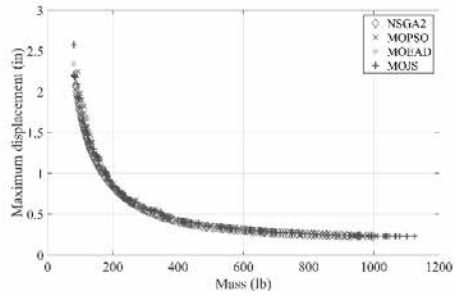
Bảng 2. Giá trị SP giải kết cấu 25 thanh

	Max	Min	Avg.	Std.
NSGA-II	3.5710	0.0000	2.7287	0.5825
MOPSO	132.8134	0.0000	9.1176	23.3957
MOJS	4.2625	2.0436	2.8884	0.5492
MOEA/D	21.8276	0.0000	2.4571	4.9749

Bảng 3. Thời gian trung bình bài kết cấu 25 thanh (giây)

NSGA-II	MOPSO	MOJS	MOEA/D
108.12	**30.20**	30.81	126.69

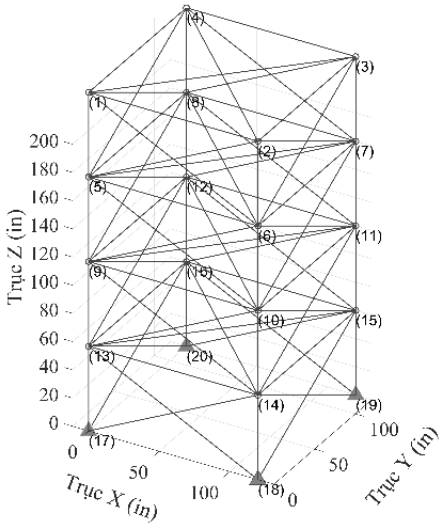
Kết quả pareto front tốt nhất được thể hiện trong hình 3, bao gồm pareto tốt nhất của 4 thuật toán. Trong bài toán 25 thanh thì thuật toán MOJS cho biểu đồ tìm kiếm có độ phủ rộng nhất (HV tốt nhất) và mật độ các giá trị thấp nhất (SP tốt nhất). Đồng thời kết quả trong bảng 3 cho thấy thời gian xử lý của MOJS (30.81s) thấp hơn nhiều so với NSGA-II (108.12s) và MOEA/D (126.69s). Điều này minh chứng MOJS là thuật toán thực hành tốt để xử lý bài toán này.



Hình 3. Kết quả tối ưu Pareto tốt nhất

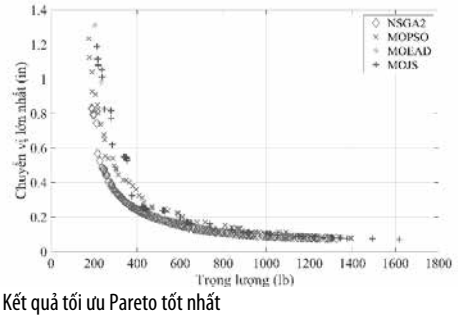
3.2. Giàn 72 thanh

Tiếp theo là bài toán kết cấu giàn không gian 72 thanh được mô tả như trong hình 4. Mật độ vật liệu là 0.1 lb/in^3 và và mô đun đàn hồi là 10000 ksi. Các phần tử phải chịu ứng suất trong phạm vi ±25 ksi. Các phần tử của kết cấu được thành 16 nhóm: (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) $A_{67} – A_{70}$, (16) $A_{71} – A_{72}$. Tập tiết diện là D = {0.1, 0.2, 0.3…2.5} (in^2). Tải trọng trên kết cấu trường hợp 1: $F_{1x} = F_{1y} = 5$ kips và $F_{1z} = -5$ kips, và tải trọng trường hợp 2: $F_{1z} = F_{2z} = F_{3z} = F_{4z} = -5$ kips. [9]



Hình 4. Mô hình giàn 72 thanh

Kết quả HV được thể hiện trong bảng 4, giá trị trung bình của thuật toán MOEA/D vẫn thấp hơn so với giá trị xấp xỉ nhau của 3 thuật toán còn lại. Về phương sai, thuật toán MOJS có phương sai nhỏ nhất, chứng tỏ các giá trị biểu đồ trong nhiều trường hợp của thuật toán MOJS luôn ổn định, các giá trị của NSGA-II dù cho kết quả bao phủ tốt nhưng vẫn xuất hiện nột vài trường hợp bị lệch nhiều so với vùng xuất hiện chính của các trường hợp. Dù vậy ta vẫn có thể xem NSGA-II và MOJS là 2 thuật toán cho kết quả tốt với ví dụ này.



Hình 5. Kết quả tối ưu Pareto tốt nhất

Bảng 4. Giá trị HV giải kết cấu 72 thanh

	Max	Min	Avg.	Std.
NSGA-II	1.0000	0.8042	0.9672	0.0618
MOPSO	0.9199	0.8058	0.8297	0.0235
MOJS	0.8303	0.7744	0.8047	0.0141
MOEA/D	0.9641	0.1017	0.4524	0.2432

Nhìn vào bảng 5, dễ nhận thấy giá trị SP của MOEA/D trải từ cận nhỏ nhất bằng 0 đến cận lớn nhất bằng 1, không những vậy các giá trị của thuật toán này cũng gần như đều bằng 0. Cả thuật toán NSGA-II và thuật toán MOPSO cũng xuất hiện các giá trị bằng 0 trong một vài ví dụ, như vậy mức độ ổn định của MOJS là cao hơn so với các thuật toán còn lại trong ví dụ này. Không những vậy, giá trị SP của MOJS nhìn chung cũng nhỏ nhất so với các thuật toán còn lại nếu không tính đến giá trị lý tưởng bằng 0 của các thuật toán còn lại.

Trong bài toán giàn 72 thanh, thuật toán MOJS tỏ ra tốt hơn khi xét đến giá trị SP của mỗi thuật toán, giá trị HV cũng thuộc nhóm thuật toán tốt. Không những vậy, khi xét đến giá trị thời gian thì MOJS (40.26s) cũng tỏ ra tốt hơn so với NSGA-II (135.60s) và MOEA/D (119.96s). Như vậy, MOJS vẫn tốt nhất trong 4 thuật toán để xử lý bài toán này.

Bảng 5. Giá trị SP giải kết cấu 72 thanh

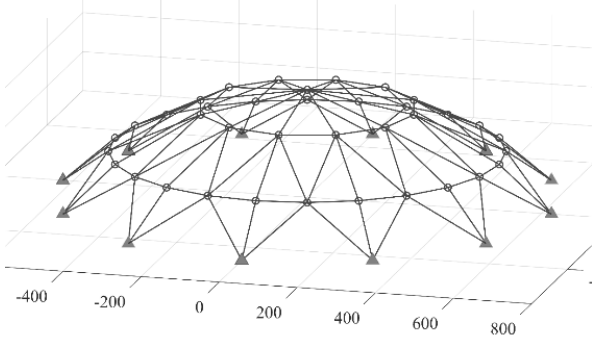
	Max	Min	Avg.	Std.
NSGA-II	21.2950	0.0000	1.0069	3.9383
MOPSO	20.1218	0.0000	13.3529	3.8931
MOJS	18.2302	1.8723	7.7773	4.3633
MOEA/D	25.9638	0.0000	3.0443	7.1609

Bảng 6. Thời gian trung bình bài kết cấu 72 thanh (giây)

NSGA-II	MOPSO	MOJS	MOEA/D
135.60	45.60	**40.26**	119.96

3.3. Mái vòm 120 thanh

Bài toán giảm thiểu trọng lượng của mái vòm 120 thanh ban đầu được nghiên cứu bởi hai nhà khoa học Soh và Yang vào năm 1996 (thể hiện trong hình 6). Hai mục tiêu chính của bài toán này là giảm thiểu tổng trọng lượng kết cấu và giảm thiểu sự chuyển vị tối đa của tất cả các nút theo mọi hướng. Mái vòm chịu tải trọng thẳng đứng -13.49 kips (60kN) tại nút 1, -6.744 kips (30kN) tại nút 2 đến nút 14 và -2.248 kips (10kN) tại các nút còn lại. Giới hạn dưới và trên diện tích mặt cắt ngang của các thanh lần lượt là 0.775 in^2 và 20.0 in^2. Các thành phần kết cấu được chia làm 7 nhóm (1) $A_1 – A_{12}$, (2) $A_{13} – A_{24}$, (3) $A_{25} – A_{48}$, (4) $A_{49} – A_{60}$, (5) $A_{61} – A_{84}$, (6) $A_{85} – A_{96}$, (7) $A_{97} – A_{120}$. Mô đun đàn hồi, ứng suất chảy và mật độ vật liệu tương ứng là E = 30450 ksi, F_y = 58 ksi và ρ = 0.288 lb/in^3. Ứng suất kéo và nén và độ mảnh cho phép được ước tính dựa trên các thông số kỹ thuật AISC-ASD [4]



Hình 6. Mô hình giàn 120 thanh

Kết quả HV được thể hiện trong bảng 7, tương tự như 2 ví dụ trên thì giá trị trung bình của thuật toán MOEA/D lần nữa chỉ tìm ra các miền giá trị thấp hơn so với giá trị xấp xỉ nhau của 3 thuật toán

còn lại. Về phương sai, với giá trị chỉ rơi vào 0.0012 thì NSGA-II tạo được vùng giá trị rất gần nhau giữa các lần xử lý với nhau, không những vậy, giá trị lớn nhất và nhỏ nhất cũng là tốt nhất. Ta có thể kết luận là HV của NSGA-II khi xét đến bài toán 120 thanh này cho giá trị tốt nhất.

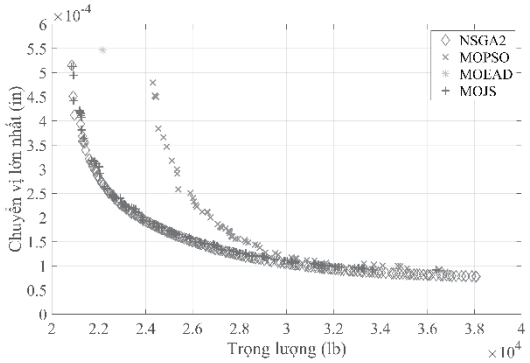
Bảng 7. Giá trị HV giải kết cấu 120 thanh

	Max	Min	Avg.	Std.
NSGA-II	0.4776	0.4721	0.4750	0.0012
MOPSO	0.4268	0.3084	0.3832	0.0267
MOJS	0.4687	0.3810	0.4451	0.0172
MOEA/D	0.3511	0.0512	0.1406	0.0918

Bảng 8. Giá trị SP giải kết cấu 120 thanh

	Max	Min	Avg.	Std.
NSGA-II	96.7592	34.6822	48.4876	14.4835
MOPSO	833.0958	21.4837	316.2612	177.6477
MOJS	602.4524	0.8584	82.0493	121.4704
MOEA/D	125.7937	0.0000	9.1911	28.7990

Thông số trong bảng 8 cho thấy giá trị SP của MOPSO trải từ cận nhỏ gần bằng 0 đến cận lớn nhất bằng 1, còn thuật toán MOEA/D gần như chỉ cho ra các giá trị SP bằng không. Thuật toán NSGA-II và MOJS nhìn chung cho kết quả ổn định hơn và thấp. Nếu so đến phương sai Std thì NSGA-II (14.48) cho kết quả phương sai tốt hơn MOJS (121.47) nên về giá trị SP thì NSGA-II vẫn tỏ ra ổn định.



Hình 7. Kết quả tối ưu Pareto tốt nhất

Bảng 9. Thời gian trung bình bài kết cấu 120 thanh (giây)

NSGA-II	MOPSO	MOJS	MOEA/D
176.54	**67.41**	76.23	103.44

Trong bài toán mái vòm 120 thanh, thuật toán NSGA-II tỏ ra tốt hơn. Kết quả pareto tốt nhất trong hình 7 dễ dàng nhận thấy độ phủ HV và khoảng cách giữa các điểm (giá trị SP) của NSGA-II là tốt nhất. Tuy nhiên, khi xem xét đến thời gian thực thi tối ưu hóa (bảng 9) thì thuật toán MOJS (76.23s) lại tỏ ra tốt hơn hẳn so với thời gian xử lý của thuật toán NSGA-II (176.54s) và thuật toán MOEA/D (103.44s). Xét đến tính ứng dụng hiệu quả về mặt thời gian, thì ta vẫn xem thuật toán MOJS (giá trị tốt thứ hai trong ví dụ này) cho kết quả hiệu quả nhất.

4. KẾT LUẬN

Nghiên cứu này xây dựng một khung giải tối ưu hóa kết cấu đa mục tiêu với điều kiện ứng suất, chuyển vị và độ mảnh của công trình dàn thép không gian. Nghiên cứu đã thực hiện so sánh 4 thuật toán tối ưu hóa đa mục tiêu cho những ví dụ cụ thể. Ứng với mỗi ví dụ, sẽ cho ra các đường pareto tốt nhất ứng với mỗi thuật toán, đồng thời kiểm tra ứng suất các thanh ứng với điểm nguy hiểm nhất của mỗi thuật toán tìm được.

Kết quả phân tích đã chỉ ra rằng không phải thuật toán tối ưu hóa nào cũng có thể xử lý các bài toán kết cấu một cách hiệu quả. Các thuật toán cũng đôi khi tạo ra sai số lớn về vị trí điểm tìm được trong những lần thực hiện lại. Thuật toán MOJS được phát triển sau này đã tỏ rõ tính ưu việt hơn so với các thuật toán tiền nhiệm khi đã có thể: 1- Luôn xử lý số liệu trong thời gian nhanh nhất, 2- Nhanh chóng tìm được các điểm tối ưu của các ví dụ, 3- Tính đồng đều, ổn định của các lần thực hiện trong các ví dụ là cao hơn hẳn các thuật toán còn lại.

TÀI LIỆU THAM KHẢO

[1] N. T. T. Thúy, C. T. Sơn, and T. V. Hùng, "Tối ưu khối lượng khung thép sử dụng phân tích trực tiếp và thuật toán tiến hóa vi phân tự thích ứng," *Tạp chí Xây dựng,* pp. 162-167, 2023.

[2] D. Zouache, Y. Ould Arby, F. Nouioua, and F. Ben Abdelaziz, "Multi-objective chicken swarm optimization: A novel algorithm for solving multi-objective optimization problems," *Computers & Industrial Engineering,* vol. 129, pp. 377-391, 2019.

[3] İ. Aydoğdu, A. Akın, and M. P. Saka, "Design optimization of real world steel space frames using artificial bee colony algorithm with Levy flight distribution," *Advances in engineering software,* vol. 92, pp. 1-14, 2016.

[4] J.-S. Chou and D.-N. Truong, "Multiobjective forensic-based investigation algorithm for solving structural design problems," *Automation in Construction,* vol. 134, p. 104084, 2022.

[5] J.-S. Chou and D.-N. Truong, "Multiobjective optimization inspired by behavior of jellyfish for solving structural design problems," *Chaos, Solitons & Fractals,* vol. 135, p. 109738, 2020.

[6] M. S. Hùng, "Nghiên cứu bài toán tối ưu đa mục tiêu khung thép sử dụng phân tích trực tiếp," *Tạp chí xây dựng,* vol. 13, no. 3, pp. 86-89, 2023.

[7] T. A. Bảo and B. Đ. Năng, "Sử dụng thuật toán tối ưu bầy đàn thiết kế tối ưu trọng lượng dầm liên hợp thép-bê tông theo tiêu chuẩn Eurocode 4," *Tạp chí Vật liệu và Xây dựng-Bộ Xây dựng,* vol. 13, no. 03, pp. 77-81, 2023.

[8] N. H. Kỳ and L. M. Hoàng, "Tối ưu kết cấu dàn thép sử dụng thuật toán bầy sói xám," *Tạp chí Vật liệu và Xây dựng-Bộ Xây dựng,* vol. 12, no. 02, pp. 27-32, 2022.

[9] T. D. Nhat and J.-S. Chou, "Integrating enhanced optimization with finite element analysis for designing steel structure weight under multiple constraints," *Journal of Civil Engineering and Management,* vol. 29, no. 8, pp. 757-786-757-786, 2023.

[10] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE transactions on evolutionary computation,* vol. 6, no. 2, pp. 182-197, 2002.

[11] S. Kannan, S. Baskar, J. D. McCalley, and P. Murugan, "Application of NSGA-II algorithm to generation expansion planning," *IEEE Transactions on Power systems,* vol. 24, no. 1, pp. 454-461, 2008.

[12] C. C. Coello and M. S. Lechuga, "MOPSO: A proposal for multiple objective particle swarm optimization," in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, 2002, vol. 2: IEEE, pp. 1051-1056.

[13] M. Jain, V. Saihjpal, N. Singh, and S. B. Singh, "An overview of variants and advancements of PSO algorithm," *Applied Sciences,* vol. 12, no. 17, p. 8392, 2022.

[14] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on evolutionary computation,* vol. 11, no. 6, pp. 712-731, 2007.

[15] W. Zheng, Y. Tan, L. Meng, and H. Zhang, "An improved MOEA/D design for many-objective optimization problems," *Applied Intelligence,* vol. 48, pp. 3839-3861, 2018.

[16] J.-S. Chou and D.-N. Truong, "A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean," *Applied Mathematics and Computation,* vol. 389, pp. 125-535, 2021.

[17] J. J. Durillo and A. J. Nebro, "jMetal: A Java framework for multi-objective optimization," *Advances in Engineering Software,* vol. 42, no. 10, pp. 760-771, 2011.

[18] S. O. Degertekin, L. Lamberti, and I. B. Ugur, "Discrete sizing/layout/topology optimization of truss structures with an advanced Jaya algorithm," *Applied Soft Computing,* vol. 79, pp. 363-390, 2019.

[19] J.-S. Chou and N.-M. Nguyen, "FBI inspired meta-optimization," *Applied Soft Computing,* vol. 93, p. 106339, 2020.