

NGHIÊN CỨU ỨNG DỤNG THUẬT TOÁN MPA (MARINE PREDATORS ALGORITHM) CHO BÀI TOÁN TỐI ƯU ĐA MỤC TIÊU

AN APPLICATION OF THE MARINE PREDATORS ALGORITHM FOR MULTI-OBJECTIVE OPTIMISATION PROBLEMS

VŨ HỮU TRƯỜNG

Khoa Công trình, Trường Đại học Hàng hải Việt Nam

Email liên hệ: truongvh.ctt@vamaru.edu.vn

Tóm tắt

Nghiên cứu này giới thiệu một phương pháp tối ưu hiệu quả cao để giải quyết các bài toán tối ưu hóa đa mục tiêu. Phương pháp được phát triển dựa trên thuật toán Marine Predators Algorithm (MPA), được giới thiệu lần đầu bởi Faramarzi và cộng sự vào năm 2020. Thuật toán được mở rộng bằng cách tích hợp bộ lưu trữ Pareto (Archive) và cơ chế chọn lọc (Selection Mechanism) để tìm ra kẻ săn mồi ưu tú nhất từ bộ lưu. Phương pháp đề xuất được kiểm chứng qua tám bài toán kiểm chuẩn tối ưu đa mục tiêu và sau đó áp dụng để giải quyết các bài toán tối ưu hóa trong lĩnh vực kỹ thuật kết cấu. Toàn bộ các bài toán được lập trình và tính toán trên phần mềm MATLAB.

Từ khóa: Tối ưu hóa đa mục tiêu, thuật toán MPA, phân loại phi vượt trội, kết cấu công trình.

Abstract

This study introduces an effective optimization method for solving multi-objective optimization problems. The method is based on the Marine Predators Algorithm (MPA), initially developed by Faramarzi et al. in 2020. The MOMPA algorithm, proposed by Zhong and colleagues in 2021, extends MPA by incorporating two critical techniques: An Archive to store the Pareto solution set and a Selection Mechanism to identify the best predator from the Archive. In this research, the MOMPA algorithm is validated using eight benchmark multi-objective test problems and then applied to optimize structural engineering problems. All mathematical problems in this study were implemented and solved using MATLAB.

Keywords: Multi-objective optimisation (MOO), Marine Predators Algorithm (MPA), non-dominated, engineering structures.

1. Mở đầu

Tối ưu đa mục tiêu là một bài toán đặc biệt trong các vấn đề tối ưu hóa, khi đó nhiều mục tiêu cùng tồn

tại cần được giải quyết trong một vấn đề tối ưu hóa. Đồng thời các mục tiêu này có xu hướng mâu thuẫn và đối lập nhau. Các vấn đề tối ưu đa mục tiêu có thể gặp trong nhiều lĩnh vực khác nhau, như: Toán học, kỹ thuật, kinh tế và nhiều lĩnh vực khác [1]. Các bài toán tối ưu hóa đa mục tiêu luôn tồn tại nhiều kịch bản tối ưu khác nhau được gọi là tập Pareto và mặt Pareto [2]. Đây cũng là thách thức của bài toán tối ưu đa mục tiêu khi phải tối ưu tất cả các mục tiêu cùng lúc [3].

Trong những thập niên gần đây, các thuật toán lấy cảm hứng từ tự nhiên (metaheuristic) được phát triển vô cùng mạnh mẽ và được minh chứng là một phương thức tiếp cận mạnh mẽ để giải quyết các bài toán tối ưu. Các thuật toán metaheuristic thường có ưu thế với các cách tiếp cận cổ điển là có thể cung cấp lời giải chính xác trong thời gian ngắn và dễ triển khai cả khi xử lý các vấn đề tối ưu phức tạp. Cho đến nay đã có đến trăm thuật toán được phát triển và giới thiệu trong những thập niên vừa qua. Có thể kể đến một số thuật toán nổi bật được phát triển cho xử lý các bài toán tối ưu đa mục tiêu như: MOMVO - thuật toán tối ưu đa mục tiêu đa vũ trụ [4], MOMSA - thuật toán tối ưu đa mục tiêu bộ ngựa, MOALO - thuật toán đa mục tiêu kiến sư từ [5],... Tuy nhiên, cần lưu ý rằng mặc dù có rất nhiều thuật toán được giới thiệu cho đến nay nhưng điều đó không có nghĩa thuật toán được phát triển sau là hoàn hảo và mạnh mẽ hoàn toàn hơn so với những thuật toán được giới thiệu trước đó. Thực tế cho thấy, đối với mỗi một vấn đề tối ưu hóa luôn tồn tại một hoặc vài thuật toán có thể đáp ứng tốt chứ không phải tất cả.

Tại Việt Nam, vấn đề tối ưu hóa nói chung và tối ưu hóa đa mục tiêu nói riêng đã và đang nhận được sự quan tâm lớn từ cộng đồng nghiên cứu trong nhiều lĩnh vực như kỹ thuật, kinh tế, và công nghiệp. Các thuật toán tối ưu lấy cảm hứng từ tự nhiên (metaheuristic) như thuật toán di truyền, thuật toán bầy đàn, và các biến thể đã được nghiên cứu và áp dụng rộng rãi. Một số nghiên cứu đã đạt được kết quả khả quan khi áp dụng các thuật toán này vào tối ưu hóa chi phí, độ bền, và hiệu quả của các kết cấu xây dựng [6-8]. Tuy nhiên, các nghiên cứu tại Việt Nam vẫn đối mặt với nhiều thách thức, bao gồm việc cải

thiện hiệu suất tính toán, đảm bảo tính ổn định của các thuật toán, cũng như áp dụng chúng vào các bài toán thực tế có quy mô lớn và yêu cầu khắt khe.

Chính vì vậy, việc tìm hiểu và ứng dụng các thuật toán hiện đại như MOMPA mở ra cơ hội nâng cao chất lượng và hiệu quả của các bài toán tối ưu hóa đa mục tiêu, góp phần thúc đẩy sự phát triển của lĩnh vực khoa học kỹ thuật tại Việt Nam. Thuật toán MOMPA được phát triển vào năm 2021 bởi Zhong và các đồng nghiệp [9], gọi là MOMPA. Thuật toán tối ưu hóa đa mục tiêu này lấy thuật toán nổi tiếng khác là thuật toán MPA làm nền tảng (được giới thiệu bởi Faramarzi cùng các đồng nghiệp vào năm 2020 [10]). Để xây dựng thuật toán MOMPA hai kỹ thuật quan trọng được sử dụng, gồm: (1) bộ lưu (archive) nơi lưu giữ tập nghiệm Pareto, (2) cơ chế chọn lọc (selection mechanism) giúp tìm kiếm kẻ săn mồi ưu tú nhất từ bộ lưu. Trong nghiên cứu này, trước khi triển khai thuật toán để tối ưu các bài toán kết cấu công trình, tám bài toán kiểm chuẩn được lựa chọn để thực hiện đánh giá hiệu quả của thuật toán.

Cấu trúc của bài báo gồm: Phần 2 giới thiệu bài toán tối ưu đa mục tiêu. Phần 3 giới thiệu thuật toán MPA. Phần 4 giới thiệu phiên bản tối ưu đa mục tiêu của thuật toán MPA. Phần 5 thực hiện kiểm chuẩn thuật toán và Phần 6 triển khai thuật toán MOMPA để tính toán tối ưu đa mục cho bài toán kết cấu công trình. Và cuối cùng, các kết luận và đề xuất được đưa ra trong Phần 7.

2. Tối ưu đa mục tiêu

Khác với bài toán tối ưu thông thường vốn chỉ giải quyết đơn nhất một hàm mục tiêu, các vấn đề tối ưu đa mục tiêu sẽ phải giải quyết các bài toán với nhiều mục tiêu xung đột với nhau. Đối với lời giải của bài toán tối ưu đa mục tiêu sẽ không phải là duy nhất mà sẽ là một tập hợp các lời giải đảm bảo sự cân bằng giữa các mục tiêu đối lập của bài toán. Điều này dẫn đến việc phải đánh giá các giải pháp để xác định vai trò giá trị giữa các mục tiêu. Về mặt toán học, một bài toán tối ưu đa mục tiêu được thể hiện như sau:

$$\begin{aligned} &\text{Tối thiểu} && F(x) = \{f_1(x), f_2(x), \dots, f_j(x)\} \\ &\text{hóa:} && \forall j = 1, 2, \dots, D \end{aligned} \quad (1)$$

$$\begin{aligned} &\text{với:} && g_k(x) \geq 0, k = 1, 2, \dots, K \\ &&& h_l(x) = 0, l = 1, 2, \dots, L \\ &&& L_i \leq x_i \leq U_i, i = 1, 2, \dots, M \end{aligned} \quad (2)$$

Trong đó: D là số hàm mục tiêu của $F(x)$, K, L là số điều kiện ràng buộc bất đẳng thức và đẳng thức, M là số biến tính; L_i và U_i lần lượt là giới hạn dưới

và trên của biến tính thứ i . Và $x = [x_1, x_2, \dots, x_M]$ là nghiệm tối ưu của bài toán đa mục tiêu. Điều này khiến khái niệm truyền thống về tối ưu cần được sửa đổi để phù hợp với sự đánh đổi giữa các mục tiêu đối lập. Khái niệm về "tối ưu đa mục tiêu" được Edgeworth giới thiệu vào năm 1881 và sau đó được Pareto củng cố vào năm 1896, được gọi là "tối ưu Pareto" và định nghĩa như sau:

- Khái niệm 1 (Sự vượt trội Pareto): Vector x vượt trội so với y (ký hiệu là $x < y$) khi và chỉ khi: $\forall i \in \{1, 2, \dots, M\}: f_i(x) \leq f_i(y)$ và $\exists j \in \{1, 2, \dots, M\}: f_j(x) < f_j(y)$.

- Khái niệm 2 (Tối ưu Pareto): Khi không có nghiệm vượt trội so với y , vector x được coi là nghiệm không vượt trội, còn được gọi là nghiệm Pareto (PS) được định nghĩa là $PS = \{x \in X | \nexists y \in X, y < x\}$.

- Khái niệm 3 (Mặt Pareto - PF) là biểu diễn của PS trong không gian mục tiêu: $PF = \{F(x) | x \in PS\}$.

3. Thuật toán MPA

Trước hết, MPA là một phương pháp dựa trên nguyên tắc bầy đàn, được lần đầu giới thiệu bởi Faramarzi cùng các đồng nghiệp vào năm 2020 [10]. Nhóm đã dựa trên thuyết sinh tồn của kẻ săn mồi mạnh nhất, nghĩa là lời giải của bài toán tối ưu chính là kẻ săn mồi ưu tú nhất. Về mặt toán học, thuật toán MPA sẽ thực hiện việc giám sát kẻ săn mồi bằng một ma trận được gọi là Elite:

$$Elite = \begin{bmatrix} X_{1,1}^l & X_{1,2}^l & \dots & X_{1,d}^l \\ X_{2,1}^l & X_{2,2}^l & \dots & X_{2,d}^l \\ \vdots & \vdots & \vdots & \vdots \\ X_{n,1}^l & X_{n,2}^l & \dots & X_{n,d}^l \end{bmatrix}_{n \times d} \quad (3)$$

Trong đó: $X_{i,j}^l (\forall i = 1 \dots n; j = 1 \dots d)$ là vị trí của kẻ săn mồi ưu tú nhất, n là số lượng tác nhân tìm kiếm và d là số chiều của không gian tìm kiếm. Vào cuối mỗi vòng lặp, $Elite$ sẽ được cập nhật bởi kẻ săn mồi ưu tú hơn. Bên cạnh đó, MPA còn sử dụng thêm một ma trận khác có cùng kích thước với $Elite$ được gọi là $Prey$, như sau:

$$Prey = \begin{bmatrix} X_{1,1} & X_{1,1} & \dots & X_{1,d} \\ X_{2,1} & X_{2,2} & \dots & X_{2,d} \\ \vdots & \vdots & \vdots & \vdots \\ X_{n,1} & X_{n,2} & \dots & X_{n,d} \end{bmatrix}_{n \times d} \quad (4)$$

Trong đó: $X_{i,j} (\forall i = 1 \dots n; j = 1 \dots d)$ là vector vị trí của con mồi. Và bầy con mồi sẽ được khởi tạo một cách ngẫu nhiên như sau:

$$Prey = rand(n, d) \times (U_b - L_b) + L_b \quad (5)$$

Trong đó: U_b, L_b là các giá trị giới hạn lớn nhất

và nhỏ nhất của không gian tìm kiếm. Quy trình thuật toán MPA được chia thành ba giai đoạn, gồm:

Giai đoạn 1: Được gọi là giai đoạn vận tốc cao tức là tỷ lệ vận tốc giữa con mồi và kẻ săn mồi lớn hơn 10 ($v > 10$). Giai đoạn này xảy ra trong các vòng lặp tính toán ban đầu của quá trình tối ưu hóa (giai đoạn khám phá - exploration). Mô hình toán học của con mồi thứ i của vòng tính thứ $iter$ cho giai đoạn này như sau:

Khi $iter < \frac{1}{3}Max_iter$

$$step_i = r_B \otimes (Elite_i - r_B \otimes Prey_i) \forall i = 1 \dots n \quad (6)$$

$$Prey_i = Prey_i + P \cdot R \otimes step_i$$

Trong đó: r_B là một vector số ngẫu nhiên cho bởi phân phối Brown. Ký hiệu \otimes là phép nhân từng phần tử tương ứng giữa hai ma trận cùng kích thước. $P = 0,5$ là một số hằng số và R là một vector số ngẫu nhiên phân phối trong $[0,1]$. $step_i$ là vector biểu diễn kích thước bước đi tiếp theo của kẻ săn mồi thứ i . $iter$ là vòng lặp tính toán hiện tại trong khi Max_iter là số vòng lặp tính toán tối đa.

Giai đoạn 2: Được gọi là giai đoạn vận tốc đơn vị tức là tỷ lệ vận tốc v giữa kẻ săn mồi và con mồi là tương đương. Đây là giai đoạn cả con mồi và kẻ săn mồi đều đang tìm kiếm thức ăn của chúng. Trong giai đoạn này, quá trình khám phá chuyển dần sang quá trình khai thác (exploitation). Mô hình toán học của giai đoạn này như sau

Khi $\frac{1}{3}Max_iter < iter < \frac{2}{3}Max_iter$ bước di

chuyển cho nửa đàn làm nhiệm vụ khám phá, được tính bởi:

$$step_i = r_L \otimes (Elite_i - r_L \otimes Prey_i) \forall i = 1 \dots n/2 \quad (7)$$

$$Prey_i = Prey_i + P \cdot r_B \otimes step_i$$

Trong đó: r_L là một vector số ngẫu nhiên dựa trên phân phối Lévy. Và bước di chuyển cho nửa đàn còn lại là nhiệm vụ khai thác được tính bởi:

$$step_i = r_B \otimes (r_B \otimes Elite_i - Prey_i) \forall i = \frac{n}{2} + 1 \dots n \quad (8)$$

$$Prey_i = Elite_i + P \cdot CF \otimes step_i$$

Trong đó: $CF = \left(1 - \frac{iter}{Max_iter}\right)^{2 \frac{iter}{Max_iter}}$ được coi là một tham số thích ứng để kiểm soát kích thước bước cho chuyển động của kẻ săn mồi.

Giai đoạn 3: Đây là giai đoạn vận tốc thấp ($v = 0.1$)

khi kẻ săn mồi di chuyển nhanh hơn con mồi. Kích bản này xảy ra trong giai đoạn cuối của quá trình tối ưu hóa, chủ yếu liên quan đến khả năng khai thác cao:

Khi $iter > \frac{2}{3}Max_iter$

$$step_i = r_L \otimes (r_L \otimes Elite_i - Prey_i) \forall i = 1 \dots n \quad (9)$$

$$Prey_i = Elite_i + P \cdot CF \otimes step_i$$

Trong thực tế, các dòng xoáy trên biển hoặc các thiết bị dụ cá (Fish Aggregating Devices - FAD) cũng có thể ảnh hưởng đến hành vi của các loài săn mồi dưới biển. Trong MPA, FAD được coi là bẫy cục bộ cho thuật toán và cần sử dụng các bước nhảy dài hơn giúp tránh tình các bẫy này. Tác động của FAD được biểu diễn về mặt toán học như sau:

$$Prey_i = \begin{cases} Prey_i + CF[X_{min} + R \otimes (X_{max} - X_{min})] \otimes U \\ Prey_i + [FDA(1 - r) + r](Prey_{r_1} - Prey_{r_2}) \end{cases} \quad (10)$$

Trong đó: $FAD=0.2$ hằng số xác suất, U là vector nhị phân gồm số không và số một, r là số ngẫu nhiên trong $[0,1]$. X_{min} và X_{max} là các vector chứa giới hạn dưới và trên của các chiều, r_1 và r_2 là các con mồi ngẫu nhiên thứ r_1 , r_2 trong bẫy.

Tiếp đến là khả năng ghi nhớ nơi có khả năng tìm kiếm được con mồi của các loài săn mồi biển, được mô phỏng bằng cách tạo bộ nhớ lưu trong MPA. Sau khi cập nhật $Prey$ và triển khai hiệu ứng FAD, ma trận này được đánh giá về độ phù hợp để cập nhật $Elite$. Giải thuật MPA được trình bày như sau:

Khởi tạo bầy con mồi $Prey$ theo Ct. (5).

Tính toán giá trị hàm mục tiêu, xây dựng ma trận $Elite$ và khởi tạo bộ nhớ theo Ct. (3) và (4).

While $iter < Max_iter$

If $iter \leq Max_iter/3$

Cập nhật $Prey$ theo Ct. (6).

Elseif $Max_iter/3 < iter \leq 2Max_iter/3$

Cập nhật $Prey$ bằng Ct. (7) với $i = 1, \dots, \frac{n}{2}$

Và cập nhật $Prey$ bằng Ct. (8) với $i = \frac{n}{2} + 1, \dots, n$

Elseif $iter > 2Max_iter/3$

Cập nhật $Prey$ bằng Ct. (9).

Endif

Tính toán lại giá trị hàm mục tiêu và thực hiện lưu bộ nhớ và cập nhật $Elite$

Triển khai FAD và cập nhật theo Ct. (10).

End while

4. Thuật toán MOMPA

Thuật toán MOMPA, lấy thuật toán MPA làm nền tảng kết hợp với hai kỹ thuật, gồm: (1) Bộ lưu (archive), được sử dụng để lưu trữ các lời giải Pareto, (2) Cơ chế chọn lọc (selection mechanism) để lựa chọn kẻ săn mồi ưu tú nhất từ bộ lưu.

4.1. Bộ lưu

Kỹ thuật này tạo ra một bộ lưu trữ này được sử dụng để lưu các lời giải không bị vượt trội. Kích thước của nó được ấn định, thường bằng một nửa kích thước của quần thể. Kho lưu trữ này luôn thu thập các giải pháp từ các quần thể hiện có và cập nhật kho lưu trữ bằng cách so sánh chúng với các giải pháp mới được tạo ra. Nó được cập nhật như sau:

- Khi một lời giải mới được tìm ra bị vượt trội bởi ít nhất một lời giải có trong bộ lưu trữ, thì lời giải mới không được chấp nhận cập nhật vào bộ lưu trữ.

- Nếu lời giải pháp mới vượt trội so với ít nhất một hoặc nhiều lời giải có trong bộ lưu, thì lời giải bị vượt trội đó sẽ bị xóa và thay thế bởi lời giải mới vượt trội hơn.

- Nếu lời giải mới không bị vượt trội trong tất cả các giải pháp trong bộ lưu, thì lời giải mới được thêm vào bộ lưu trữ với tư cách là thành viên mới.

Nếu bộ lưu bị đầy và lời giải mới đáp ứng các tiêu chí để được cập nhật, nó sẽ bị mất. Theo tình huống này, cần phải xóa bớt một lời giải khỏi bộ lưu. Việc lựa chọn lời giải để xóa khỏi bộ lưu được lựa chọn dựa theo phương pháp "Roulette" để tính toán xác suất loại bỏ của lời giải thứ i trong bộ lưu:

$$P_i = \frac{N_i}{C} \quad (11)$$

Trong đó: C là hằng số lớn hơn 1 và N_i là số lượng các giải pháp lân cận của giải pháp thứ i trong bộ lưu. Và là số giải pháp nằm trong kích thước d được tính bởi:

$$d = \frac{Fmax - Fmin}{Archivesize} \quad (12)$$

với, $Fmax$, $Fmin$ là các giá trị lớn nhất và nhỏ nhất của các hàm mục tiêu thu được sau mỗi vòng tính; $Archivesize$ là kích thước lớn nhất cho phép của bộ lưu.

4.2. Cơ chế chọn lọc

Đây là cơ chế phân loại và lựa chọn kẻ săn mồi ưu tú nhất. Nó có tác dụng định hướng di chuyển cho những cá thể khác trong quần thể và khả năng hội tụ bị chi phối bởi nó. Mỗi cá thể săn mồi được chỉ định một giá trị hàm mục tiêu có thể được so sánh trực tiếp, được tính bởi:

$$f_i = N_i \quad (13)$$

Như vậy, giá trị hàm mục tiêu mới của mỗi cá thể tương ứng với số lượng giải pháp lân cận. Khi đó, tất cả các cá thể có giá trị hàm mục tiêu thấp nhất được lựa chọn làm kẻ săn mồi ưu tú nhất. Và để thực hiện việc so sánh và lựa chọn kẻ săn mồi ưu tú nhất, các khái niệm của Pareto được sử dụng. Thuật toán MOMPA cho phép lựa chọn không chỉ một mà là nhiều kẻ săn mồi ưu tú. Dựa trên cơ chế này ma trận *Elite* được cập nhật và hình thành:

$$Elite = \begin{bmatrix} X_{1,1}^1 & X_{1,2}^1 & \dots & X_{1,d}^1 \\ X_{2,1}^2 & X_{2,2}^2 & \dots & X_{2,d}^2 \\ \vdots & \vdots & \vdots & \vdots \\ X_{k,1}^k & X_{k,2}^k & \dots & X_{k,d}^k \\ X_{k+1,1}^1 & X_{k+1,2}^1 & \dots & X_{k+1,d}^1 \\ \vdots & \vdots & \vdots & \vdots \\ X_{n,1}^m & X_{n,2}^m & \dots & X_{n,d}^m \end{bmatrix}_{n \times d} \quad (14)$$

Trong đó: $X^1, X^2, \dots, X^m, \dots, X^k$ là kẻ săn mồi ưu tú nhất được chọn bằng cơ chế chọn lọc, với $1 < m < k$ ($k \leq$ kích thước bộ lưu). Giải thuật của MOMPA được cho như sau:

Khởi tạo bầy con mồi *Prey* theo Ct. (5).

While $iter < Max_iter$

Tính toán giá trị hàm mục tiêu, lọc tìm và cập nhật lời giải không bị vượt trội

If Bộ lưu đầy

Loại bỏ bớt lời giải trong bộ lưu và cập nhật

End if

Chọn lựa kẻ săn mồi ưu tú nhất theo cơ chế lọc.

Hình thành ma trận *Elite* theo Ct. (14).

If $iter \leq Max_iter/3$

Cập nhật *Prey* theo Ct. (6).

Elseif $Max_iter/3 < iter \leq 2Max_iter/3$

Cập nhật *Prey* bằng Ct. (7) với $i = 1, \dots, \frac{n}{2}$

Và cập nhật *Prey* bằng Ct. (8) với $i = \frac{n}{2} +$

$1, \dots, n$

Elseif $iter > 2Max_iter/3$

Cập nhật *Prey* bằng Ct. (9).

End (if)

Thực hiện lưu bộ lưu

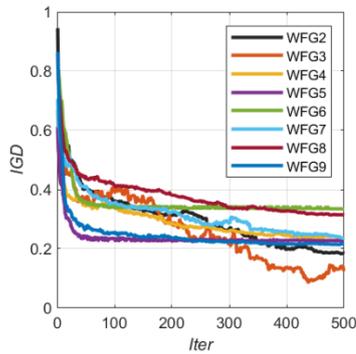
Triển khai FAD và cập nhật theo Ct. (10).

End while

5. Kiểm chuẩn

Bảng 1. Kết quả kiểm chuẩn thuật toán

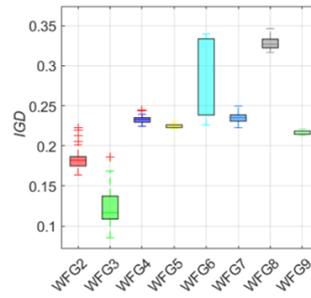
Hàm kiểm chuẩn	CSTK	MOMPA
WFG2	(1)	0.1816
	(2)	0.0111
WFG3	(1)	0.1242
	(2)	0.0202
WFG4	(1)	0.2321
	(2)	0.0044
WFG5	(1)	0.2252
	(2)	0.0033
WFG6	(1)	0.3027
	(2)	0.0475
WFG7	(1)	0.2363
	(2)	0.0061
WFG8	(1)	0.3261
	(2)	0.0065
WFG9	(1)	0.2171
	(2)	0.0028



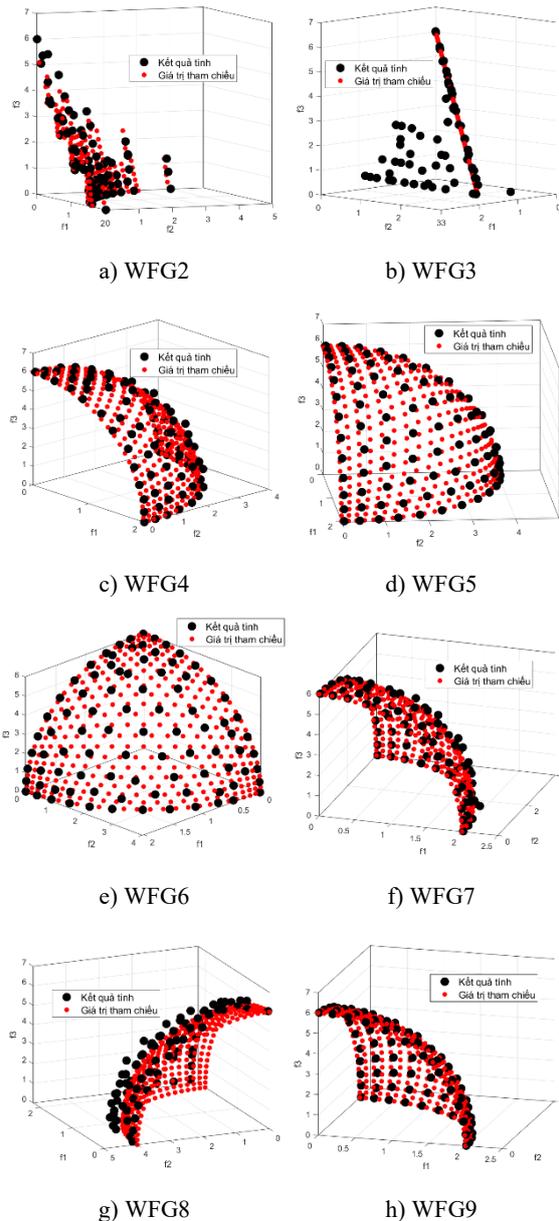
Hình 1. Giá trị IGD qua các vòng tính

Để kiểm chuẩn hiệu quả tính toán của thuật toán MOMPA được giới thiệu trong bài báo này, các hàm kiểm chuẩn được triển khai, như: WFG2, WFG3, WFG4, WFG5, WFG6, WFG7, WFG8 và WFG9 [11]. Các bài toán kiểm chuẩn này bao gồm 3 phương trình mục tiêu cần tối ưu là $F(f) = (f_1(x), f_2(x), f_3(x))$, chi tiết tham khảo tài liệu [11]. Và để đánh giá được kết quả thu được từ thuật toán khi giải các hàm kiểm chuẩn, thông số IGD được đề xuất sử dụng. Đây là chỉ số IGD cho phép đánh giá chi tiết về mức độ chính xác của thuật toán [12, 13].

$$IGD = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{n_{PF}} \quad (17)$$



Hình 2. Biểu đồ hộp của IGD

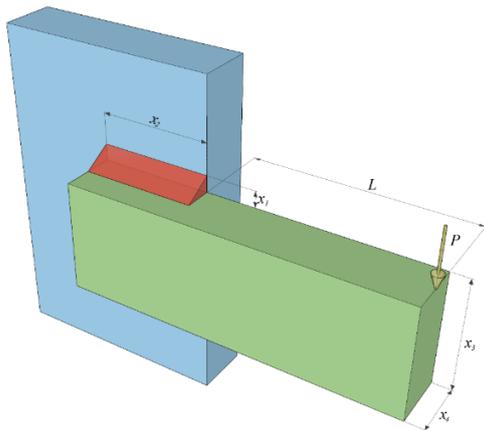


Hình 3. Kết quả tính so với nghiệm tham chiếu của các hàm kiểm chuẩn

Trong đó: n_{PF} là số lời giải không bị vượt trội và d_i là khoảng cách Euclid giữa tập nghiệm tham chiếu và tập lời giải Pareto tìm được. Như vậy, IGD đo đặc sai cách tối thiểu trung bình từ tập tham chiếu đến điểm gần nhất trong tập nghiệm giải được [12]. Chính vì vậy, giá trị kỳ vọng để minh chứng cho thuật toán có hiệu quả tính toán mạnh là những giá trị IGD thấp tiệm cận đến giá trị 0 [12].

Bên cạnh đó, để đánh giá được hiệu suất trong tính toán của thuật toán, các bài toán kiểm chuẩn sẽ được triển khai tính toán với 30 lần tính độc lập. Và hai chỉ số thống kê (CSTK) quan trọng sau được sử dụng, gồm: (1) Giá trị trung bình - Mean và (2) Độ lệch chuẩn - Standard Deviation (SD). Các thông số ban đầu của thuật toán được lấy như sau: Số lượng cá thể sẵn mỗi là 100, số vòng lặp tính tối đa là 500, hệ số ảnh hưởng FDA là 0,2. Kết quả kiểm chuẩn thuật toán được thể hiện chi tiết trong Bảng 1, Hình 1 - Hình 3. Trong đó, Bảng 1 cho thấy giá trị của các thông số thống kê cho chỉ số IGD thông qua 30 lần tính toán độc lập đối với mỗi bài toán kiểm chuẩn. Căn cứ vào giá trị thu được có thể khẳng định rằng hiệu suất tính toán của thuật toán được giới thiệu trong nghiên cứu này (MOMPA) là tương đối tốt. Các giá trị IGD thu được là thấp và chỉ số độ lệch chuẩn (SD thấp) cũng cho thấy thuật toán có hiệu suất làm việc cao.

Thêm vào đó, căn cứ theo Hình 1 có thể thấy thuật toán có xu hướng đạt giá trị hội tụ tương đối sớm (trước 100 vòng tính). Điều này sẽ giúp giảm thiểu thời gian tính toán cho những bài toán tối ưu đa mục tiêu phức tạp trong thực tế.



Hình 4. Dầm công-xôn [11]

Và cuối cùng, trong Hình 2 nơi biểu diễn biểu đồ hộp (biểu đồ biểu diễn sự phân phối của tập dữ liệu) cho thấy thuật toán MOMPA có hiệu suất làm việc tương đối ổn định (các khối hộp tương đối nhỏ hẹp, như WFG2,

WFG3, WFG4, WFG5, WFG7, WFG8 và WFG9).

6. Tối ưu đa mục tiêu kết cấu

Đối với việc tối ưu đa mục tiêu cho các kết cấu công trình, trong nghiên cứu này nhóm tác giả triển khai thuật toán MOMPA để tính toán tối ưu cho một số bài toán như: (1) Kết cấu dầm công-xôn, xem Hình 4 [14]; (2) Kết cấu giàn vòm thép, xem Hình 5 [15].

6.1. Tối ưu đa mục tiêu dầm công-xôn

Dầm công-xôn được sử dụng trong tính toán này là một dầm được liên kết bằng đường hàn có kết cấu cho như sau Hình 4. Đối với bài toán tối ưu đa mục tiêu cho kết cấu này, hàm tối ưu gồm hai mục tiêu: (1) $f_1(x)$ hàm tổng chi phí và (2) $f_2(x)$ hàm chuyển vị đầu dầm:

Cực tiểu hóa:

$$\begin{cases} f_1(x) = 1.10471 \times x_1^2 \times x_2 \\ \quad + 0.04811 \times x_3 \times x_4 \times (14 + x_2) \\ f_2(x) = \frac{65856000}{30.10^6 \times x_4 \times x_3^3} \end{cases} \quad (18)$$

Với các điều kiện ràng buộc được gồm có:

$$\begin{cases} g_1(x) = \tau - 13600 \leq 0 \\ g_2(x) = \sigma - 30000 \leq 0 \\ g_3(x) = x_1 - x_4 \leq 0 \\ g_4(x) = P - 6000 \leq 0 \\ g_5(x) = 0.125 - x_1 \leq 0 \end{cases}$$

$$\text{và: } \begin{cases} 0.125 \leq x_1 \leq 5 \\ 0.1 \leq x_2, x_3 \leq 10 \\ 0.125 \leq x_4 \leq 5 \end{cases}$$

Trong đó:

$$\begin{cases} \tau = \sqrt{(\tau')^2 + \frac{2\tau'\tau''}{2R} + \tau''} \\ R = \sqrt{\frac{x_2^2}{4} + \frac{x_1+x_3^2}{4}} \\ \tau' = \frac{P}{\sqrt{2}x_1x_2}; \tau'' = \frac{MR}{J}; M = P \left(L + \frac{x_2}{2} \right) \\ J = 2 \times \left(x_1 + x_2\sqrt{2} \times \left(\frac{x_2}{12} + \frac{(x_1+x_3)^2}{4} \right) \right) \\ \sigma = \frac{6PL}{x_4x_3^2} \end{cases} \quad (19)$$

Với: τ, τ', τ'' là các ứng suất tiếp trong dầm, σ là ứng suất pháp, R bán kính hiệu dụng, M là mô-men lớn nhất trong dầm, J là mô-men quán tính của tiết diện dầm, P, L, x_1, x_2, x_3, x_4 là tải trọng, chiều dài và các kích thước cơ bản của dầm, xem Hình 4.

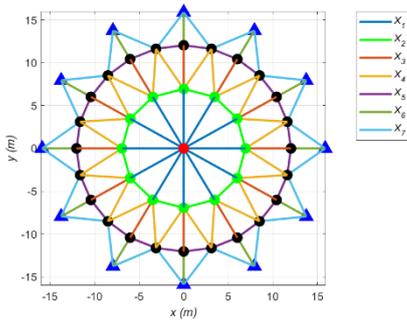
6.2. Tối ưu đa mục tiêu giàn vòm thép không gian

Giàn vòm thép không gian được sử dụng trong nghiên cứu này được cấu tạo bởi 120 thanh, xem Hình

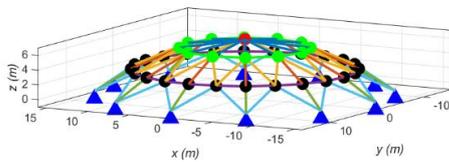
5. Đối với kết cấu này, hàm tối ưu của bài toán có hai mục tiêu, gồm: (1) $f_1(x)$ là hàm tổng khối lượng của giàn và $f_2(x)$ là hàm biến dạng của giàn.

$$\text{Cực tiểu hóa: } \begin{cases} f_1(x) = \sum_{j=1}^N \rho x_j L_j \\ f_2(x) = \mathbf{u}^T \mathbf{F} \end{cases} \quad (20)$$

Với điều kiện ràng buộc về ứng suất trong các thanh giàn cho bởi: $\sigma_{max} - \sigma_a \leq 0$. Trong đó, $N = 120$ thanh, $\rho = 7850 \text{ kg/m}^3$, x_j và L_j là tiết diện mặt cắt ngang và chiều dài của thanh thứ j ; σ_{max} là ứng suất của các thanh giàn và $\sigma_a = 400 \text{ MPa}$ là ứng suất cho phép. Ngoài ra, \mathbf{u} là chuyển vị của các đầu thanh, \mathbf{F} là các tải trọng tác dụng lên giàn.



a) Hình chiếu bằng giàn vòm thép



b) Hình chiếu không gian giàn vòm thép

Hình 5. Giàn thép vòm không gian [12]

Trong bài toán tối ưu kết cấu giàn vòm thép cần lưu ý rằng toàn bộ khung giàn có 120 thanh, tuy nhiên điều đó không có nghĩa tổng số biến tính của bài toán là 120 biến. Thực tế trong tính toán 120 thanh sẽ được phân thành các nhóm có tiết diện giống nhau có vai trò tương tự (chịu ứng suất tương đương). Trong bài toán thiết kế tối ưu này 120 thanh sẽ được phân làm 7 nhóm thanh. Điều đó có nghĩa rằng bài toán sẽ có 7 biến tính $0 \leq X_i (i = 1: 7) \leq 1$, xem Hình 5. Các tải trọng tác dụng lên giàn là các tải trọng thẳng đứng được đặt tại các nút giàn lần lượt như sau: Các nút màu đen ($P = 500 \text{ kN}$), từ nút xanh lá cây ($P =$

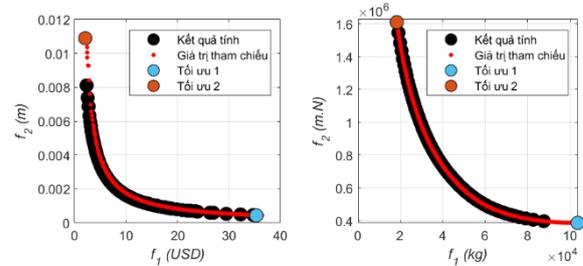
1500 kN) và nút màu đỏ ($P = 3000 \text{ kN}$).

6.3. Kết quả tính toán

Bảng 2. Kết quả kiểm chuẩn thuật toán

Kết cấu tối ưu	CSTK	IGD
Dầm công-xôn	(1)	0.3919
	(2)	0.0734
Giàn vòm thép	(1)	0.1229
	(2)	0.0183

Để phục vụ tính toán cho hai bài toán kết cấu công trình nêu trên, các thông số tính toán ban đầu được thiết lập như sau: Tổng số cá thể sản mỗi là 100; đồng thời thuật toán MOMPA vẫn sẽ được triển khai tính toán 30 lần độc lập; số vòng lặp tính toán lớn nhất của mỗi lần tính toán là 500 vòng.



a) Dầm công-xôn

b) Giàn vòm thép

Hình 6. Kết quả tính so với tham chiếu

Kết quả tính toán của hai bài toán kết cấu công trình trước hết được thể hiện thông qua thông số IGD trong Bảng 2 và Hình 6. Căn cứ theo giá trị SD của trị số IGD được thể hiện trong Bảng 2, có thể thấy rằng thuật toán MOMPA có hiệu suất tính toán vẫn rất tốt (SD nhỏ). Thêm vào đó, theo kết quả so sánh thể hiện trong Hình 6, chúng ta có thể khẳng định rằng kết quả thu được từ thuật toán MOMPA là trùng khớp với kết quả tham chiếu (cả hai bài toán tối ưu kết cấu). Thậm chí rằng kết quả của thuật toán MOMPA đối với bài toán dầm công-xôn có xu hướng vượt qua so với kết quả tham chiếu.

Bảng 3. Kết quả tối ưu dầm công-xôn

Kết quả	$x_1(m)$	$x_2(m)$	$x_3(m)$	$x_4(m)$	$f_1(USD)$	$f_2(m)$
Tối ưu 1	0.125	6.044	10.000	0.201	2.05	0.01
Tối ưu 2	1.305	0.390	10.000	5.000	35.35	0.0004

Bảng 4. Kết quả tối ưu giàn vòm thép

Kết quả	$x_1(m^2)$	$x_2(m^2)$	$x_3(m^2)$	$x_4(m^2)$	$x_5(m^2)$	$x_6(m^2)$	$x_7(m^2)$	$f_1(kg)$	$f_2(m.N)$
Tối ưu 1	0.146	0.359	0.249	0.059	0.001	0.314	0.105	1.83E+04	1.61E+06
Tối ưu 2	0.989	0.999	0.980	0.977	1.000	1.000	1.000	1.04E+05	3.87E+05

Bên cạnh đó, cần lưu ý rằng khác với bài toán tối ưu đơn mục tiêu luôn chỉ có một kết quả tối ưu duy nhất. Đối với bài toán tối ưu đa mục tiêu kết quả tối ưu của bài toán là một tập nghiệm (được gọi là tập nghiệm Pareto). Các nghiệm trong tập này sẽ cho thấy sự đối lập của các mục tiêu trong bài toán tối ưu đa mục tiêu. Điều này được thể hiện thông qua hai kịch bản tối ưu được thu thập trong Bảng 3 và Bảng 4. Khi mục tiêu thứ nhất f_1 đạt giá trị tối thiểu thì mục tiêu thứ hai f_2 sẽ đạt giá trị cực đại và ngược lại.

7. Kết luận

Nghiên cứu đã áp dụng thuật toán MOMPA để giải quyết hai bài toán tối ưu hóa đa mục tiêu trong lĩnh vực kết cấu công trình, bao gồm kết cấu dầm công-xôn và kết cấu giàn vòm thép không gian. Kết quả tính toán cho thấy thuật toán MOMPA đạt được hiệu suất cao, với các giá trị chỉ số IGD thấp và độ lệch chuẩn nhỏ, minh chứng cho khả năng hội tụ ổn định và độ chính xác cao của thuật toán.

Cụ thể, đối với bài toán dầm công-xôn, thuật toán đã đưa ra các giải pháp tối ưu với sự cân bằng rõ ràng giữa hai mục tiêu chi phí và chuyển vị đầu dầm, đáp ứng yêu cầu kỹ thuật. Đối với bài toán giàn vòm thép không gian, MOMPA đã tối ưu hóa thành công khối lượng tổng thể của giàn trong khi vẫn đảm bảo giới hạn về ứng suất và biến dạng, đồng thời phân bố hợp lý các tiết diện thanh giàn. Ngoài ra, kết quả phân tích nghiệm Pareto chỉ ra rằng thuật toán có khả năng khai thác tốt các giải pháp tối ưu, đảm bảo sự cân bằng giữa các mục tiêu xung đột. MOMPA đã chứng minh là một công cụ hiệu quả và mạnh mẽ để giải quyết các bài toán tối ưu hóa kết cấu phức tạp, với khả năng ứng dụng rộng rãi trong thực tiễn.

Lời cảm ơn

Nghiên cứu này được tài trợ bởi Trường Đại học Hàng hải Việt Nam trong đề tài mã số: **DT24-25.88**.

TÀI LIỆU THAM KHẢO

[1] Gunantara, N. (2018), *A review of multi-objective optimization: Methods and its applications*. Cogent Engineering, Vol.5(1): p.1502242.
 [2] Abouhawwash, M., M. Jameel, and K. Deb (2020), *A smooth proximity measure for optimality*

in multi-objective optimization using Benson's method. Computers & Operations Research, Vol.117, p.104900.

[3] Jameel, M. and M. Abouhawwash (2024), *Multi-objective Mantis Search Algorithm (MOMSA): A novel approach for engineering design problems and validation*. Computer Methods in Applied Mechanics and Engineering, Vol.422, p.116840.
 [4] Mirjalili, S., et al. (2017), *Optimization of problems with multiple objectives using the multi-verse optimization algorithm*. Knowledge-Based Systems, Vol.134, pp.50-71.
 [5] Mirjalili, S., P. Jangir, and S. Saremi (2017), *Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems*. Applied Intelligence, Vol.46(1), pp.79-95.
 [6] Tiên, N.T.C., et al. (2022), *Ứng dụng tối ưu đa mục tiêu cho bài toán tối ưu tổ hợp với hàm mục tiêu nhân tính*. Tạp chí Khoa học Đại học Cần Thơ. Số 58(6), tr.20-29.
 [7] Học, T.Đ., et al. (2016), *Nghiên cứu ứng dụng thuật toán tiến hóa vi phân đa mục tiêu trong tối ưu tiến độ và chi phí cho dự án*. Tạp chí Khoa học và Công nghệ - Đại học Đà Nẵng. Số 1, tr.98-102.
 [8] Hùng, M.S.J.T.c.V.l.v.X.d.-B.X.d. (2023), *Tối ưu đa mục tiêu giàn thép với biến thiết kế rời rạc sử dụng phân tích trực tiếp*. Tạp Chí Vật liệu Và Xây dựng - Bộ Xây dựng. Số 13(01), tr.63-66.
 [9] Zhong, K., et al. (2021), *MOMPA: Multi-objective marine predator algorithm*. Computer Methods in Applied Mechanics and Engineering, Vol.385, p.114029.
 [10] Faramarzi, A., et al. (2020), *Marine Predators Algorithm: A nature-inspired metaheuristic*. Expert Systems with Applications, Vol.152, p.113377.
 [11] Chand, S. and M. Wagner (2015), *Evolutionary many-objective optimization: A quick-start guide*. Surveys in Operations Research and Management Science, Vol.20(2), pp.35-42.

- [12] Audet, C., et al. (2021), *Performance indicators in multiobjective optimization*. European Journal of Operational Research, Vol.292(2), pp.397-422.
- [13] Coello, C.A.C. and N.C. Cortés (2005), *Solving Multiobjective Optimization Problems Using an Artificial Immune System*. Genetic Programming and Evolvable Machines, Vol.6(2), pp.163-190.
- [14] Ray, T. and K.M. Liew, A Swarm Metaphor for Multiobjective Design Optimization. Engineering Optimization, 2002. 34(2): pp.141-153.
- [15] Srinivas, N. and K. Deb (1994), *Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms*. Evolutionary Computation, Vol.2(3), pp.221-248.

Ngày nhận bài:	04/11/2024
Ngày nhận bản sửa:	24/11/2024
Ngày duyệt đăng:	09/12/2024