

TỐI ƯU HÓA KHỐI LƯỢNG KHUNG GIÀN THÉP
BẰNG THUẬT TOÁN BẦY DƠI
AN ADVANCED VARIANT OF THE BAT ALGORITHM
TO OPTIMISE THE WEIGHT OF TRUSS STRUCTURES
ĐỖ THỊ MINH TRANG*, VŨ THỊ CHI, VŨ HỮU TRƯỜNG

Khoa Công trình, Trường Đại học Hàng hải Việt Nam

*Email liên hệ: minhtrang.ctt@vimaru.edu.vn

Tóm tắt

Bài báo này nhằm giới thiệu một phiên bản đã được cải tiến mạnh hơn của một thuật toán mô phỏng (metaheuristic) sao chép cơ chế săn mồi bầy đàn của loài dơi (bat). Đây là một phiên bản được cải tiến nhằm phục vụ cho các vấn đề kỹ thuật tối ưu hóa và thiết kế tối ưu hóa với hiệu suất cao hơn so với thuật toán bầy dơi cơ sở (BA). Cũng như những thuật toán mô phỏng khác, BA rất dễ triển khai để giải quyết nhiều vấn đề với tính linh hoạt cao. Tuy nhiên, sự thay đổi quá nhanh giữa hai quá trình gồm quá trình khai thác (exploitation) và quá trình thăm dò (exploration) sẽ dẫn đến xác suất thành công thấp trong việc tìm ra con mồi thực sự của bầy dơi. Do đó, kỹ thuật cải tiến được giới thiệu trong nghiên cứu này là một cách thức tạo tiếng vang (loudness) mới nhằm điều chỉnh mức di chuyển ngẫu nhiên (random walk) của các cá thể trong bầy dơi một cách hiệu quả hơn. Để kiểm chứng hiệu quả của đề xuất, các bài toán mẫu (mathematical benchmark) sẽ được sử dụng. Cuối cùng, BA với kỹ thuật cải tiến tiếng vang được áp dụng để thực hiện tối ưu hóa khối lượng của một kết cấu khung giàn.

Từ khóa: Thuật toán mô phỏng, tối ưu hóa, khung giàn, thuật toán bầy dơi.

Abstract

This paper introduces an advanced variant of a metaheuristic that replicates the bat-hunting mechanism. The proposed algorithm is applied to solve the optimisation engineering problems with superior performance compared to the original bat algorithm (BA). As with other metaheuristics, BA is easy to implement in solving many problems with great flexibility. However, changing too quickly between the two processes, exploitation and exploration, could cause low performance in finding the real prey. Therefore, this study provides a novel of the loudness function to

control the random walk of bats more efficiently. The proposed BA is tested to mathematical benchmarks and then applied to optimise the weight of the truss structure.

Keywords: Metaheuristic, optimisation, truss structure, bat algorithm.

1. Mở đầu

Những năm gần đây đã chứng kiến sự mở rộng một cách nhanh chóng về số lượng các thuật toán lấy cảm hứng từ tự nhiên nhằm phục vụ các vấn đề tối ưu hóa. Iztok Fister Jr. và cộng sự thống kê được hơn 40 thuật toán được phát triển và giới thiệu [1]. Các thuật toán này dựa trên việc sao chép các hiện tượng tự nhiên như các quy luật sinh học hoặc vật lý. Chúng được phân loại làm ba nhóm chính: (1) nhóm mô phỏng quy luật vật lý, (2) nhóm dựa trên quy luật tiến hóa và (3) nhóm dựa trên quy luật bầy đàn [2]. Nhóm thứ nhất dựa trên các quy luật vật lý, như: Thuật toán trọng lực (GSA) [3], thuật toán hố đen (BH) [4] và thuật toán không gian cong (CSO) [5],... Nhóm thứ hai, sử dụng cảm hứng từ các quy luật tiến hóa tự nhiên, như: Các thuật toán di truyền (GA) [6], thuật toán tiến hóa chiến lược (ES) [7] và thuật toán lập trình di truyền (GP) [8],... Nhóm cuối cùng dựa trên việc sao chép các hành vi xã hội của các hệ thống sinh học của bầy đàn động vật, ví dụ: Thuật toán bầy đàn (PSO) [9], thuật toán đàn kiến (ACO) [10], và thuật toán bầy dơi (BA) [11],... Trong đó, BA là một thuật toán được giới thiệu lần đầu vào năm 2010 bởi Xin-SheYang [11]. Nó dựa trên việc sao chép hành vi định vị bằng tiếng vang của loài dơi. Như đánh giá của chúng tôi cho thấy, BA có những lợi ích của một siêu thuật toán (metaheuristic) dựa trên trí tuệ bầy đàn.

Thêm vào đó, BA sử dụng khả năng phát xung (r) và tiếng vang (A) nhằm tự động thu phóng (auto zooming) vào nơi tìm kiếm [12]. Việc thu phóng này đi kèm với quá trình chuyển đổi từ hoạt động thăm dò sang khai thác cục bộ cường độ cao để đạt được kết quả tối ưu hóa. Nó là một lợi ích bổ sung của BA so với các thuật toán khác. Chúng là chìa khóa để BA có

thể hoạt động hiệu quả với sự hội tụ sớm hơn các thuật toán khác [19]. Tuy nhiên, thực tế là BA vẫn có thể bị kẹt lại ở hoạt động thăm dò dẫn tới sai sót trong việc tìm kiếm kết quả tối ưu toàn cục. Nguyên nhân dẫn tới điều này là do việc thực hiện chuyển đổi giữa hai quá trình thăm dò và khai thác được thực hiện quá sớm. Do đó, rất nhiều phương pháp đã được đề xuất để giải quyết vấn đề này, và hình thành một số biến thể BA với hiệu suất được cải thiện tốt hơn [11]. Ví dụ, thuật toán cải tiến bởi Jamil và cộng sự sử dụng kỹ thuật Lévy (IBA) [13]. Huang và cộng sự đã giới thiệu một phiên bản cải tiến của thuật toán bầy dơi bằng cách kết hợp BA với phương pháp Latinh trực giao (MBA). Nakamura và cộng sự [14] và Mirjalili và cộng sự [15] đã giới thiệu một biến thể của kỹ thuật rời rạc được gọi là thuật toán bầy dơi nhị phân (BBA). Bạn đọc có thể tham khảo thêm các biến thể của BA cải tiến từ rất nhiều nguồn tài liệu trên thế giới.

BA đã được ứng dụng trong hầu hết các vấn đề tối ưu hóa, xử lý hình ảnh, phân loại, khai thác dữ liệu, lựa chọn tính năng,... Như đã khẳng định, mặc dù thực tế là BA hiệu quả, nhưng vẫn cần có những cải tiến nhằm tăng hiệu suất của của thuật toán [11]. Do đó, nghiên cứu này đề xuất một cải tiến nhằm tác động vào quá trình thăm dò và khai thác từ đó cải thiện hiệu suất tìm kiếm của BA trong việc tìm ra lời giải tối ưu toàn cục một cách hiệu quả hơn.

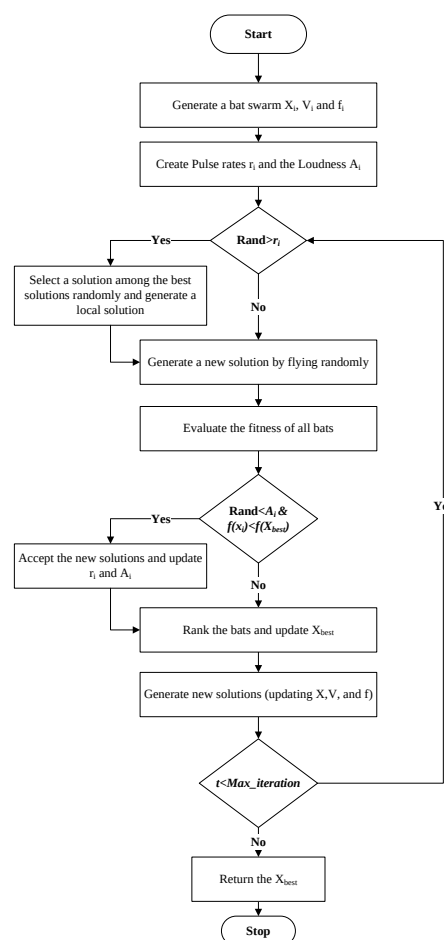
Bài báo có cấu trúc như sau: Phần 2 giới thiệu thuật toán BA cơ sở. Phần 3 giới thiệu đề xuất trong kiểm soát tiếng vang A. Phần 4 và 5 tiến hành kiểm chuẩn thuật toán được đề xuất với các bài toán mẫu. Và phần 6 cung cấp những kết quả áp dụng của thuật toán trong tìm kiếm tối ưu khối lượng khung giàn thép. Các kết luận được cung cấp trong Phần 7.

2. Cấu trúc thuật toán BA

Dơi là loài vật rất phổ biến ở hầu hết mọi nơi trên Trái đất và chúng là loài động vật có vú duy nhất có thể bay được. Trong tự nhiên, dơi sử dụng định vị bằng tiếng vang làm phương tiện chính để định hướng bản thân, mặc dù không phải tất cả chúng đều bị mù.

Cơ chế này cũng hỗ trợ trong việc phát hiện con mồi và phân biệt giữa các loài côn trùng khác nhau. Đây chính là nguồn cảm hứng chính trong việc thiết kế và phát triển thuật toán BA với tính đơn giản và linh hoạt cao của Xin-She Yang vào năm 2010 [16]. Như đã biết, BA có khả năng điều chỉnh tần số và định vị tiếng vang đáp ứng với các tần số khác nhau [11]. Chức năng này mang lại lợi thế cho BA so với các thuật toán dựa trên trí thông minh bầy đàn khác, như là PSO, SA và HS [17]. Một tiến bộ khác biệt khác

của BA là khả năng tự động thu phóng làm cho quá trình khai thác trở nên mạnh hơn trong việc tìm kiếm các lời giải toàn cục [20]. Ngoài ra, thuật toán BA thực hiện việc điều khiển tham số để điều chỉnh tốc độ phát xung và tiếng vang của các bước lặp thay vì sử dụng các tham số xác định trước [12]. Đây là chìa khóa giúp bầy dơi chuyển đổi từ thăm dò sang khai thác khi tiếp cận gần lời giải tối ưu. Một cách tổng quát, thuật toán BA được mô tả như sau [18].



Hình 1. Sơ đồ giải thuật BA

Đầu tiên, hãy xem xét một con dơi nhân tạo trong BA có vector vị trí và vector vận tốc được cập nhật trong sau mỗi lần lặp lại, như:

$$V_i(t+1) = (X_i(t) - X_{best})F_i + V_i(t) \quad (1)$$

$$X_i(t+1) = V_i(t+1) + X_i(t) \quad (2)$$

trong đó, X_{best} là lời giải tối ưu đạt được ở lần lặp thứ t và F_i là tần số của lần thứ lặp i , được điều chỉnh sau mỗi chu kỳ tính toán như sau:

$$F_i = F_{min} + (F_{max} - F_{min})\beta \quad (3)$$

trong đó, β là một số có giá trị thay đổi trong phạm vi $[0,1]$.

Giải thuật 1: Quy trình mã giải thuật của BA [15]

Đầu vào: Khởi tạo bầy dơi $X_i (i = 1, 2, \dots, n)$ và V_i .

Xác định tần số xung F_i

Khởi tạo mức xung r_i và tiếng vang A_i

Đầu ra: Lời giải tối ưu X_{best} và giá trị tối ưu tương ứng $f(X_{best})$.

while ($t < \text{số lần lặp tối đa}$)

Thiết lập lời giải mới bằng cách điều chỉnh tần số, tốc độ và vị trí của các cá thể dơi [ct. (1)-(3)]

if ($\text{rand} > r_i$)

Lựa chọn kết quả tốt nhất trong các kết quả tối ưu một cách ngẫu nhiên

Thiết lập kết quả cục bộ xung quanh các kết quả tối ưu đã chọn

end if

Tạo lời giải mới bằng kỹ thuật di chuyển ngẫu nhiên

if ($\text{rand} < A_i \ \& \ f(X_i) < f(X_{best})$)

Chấp nhận lời giải mới X_{best}

Tăng r_i và giảm A_i

end if

Xếp hạng các cá thể dơi và tìm giá trị tối ưu $f(X_{best})$

end while

Bên cạnh đó, BA áp dụng phương pháp di chuyển ngẫu nhiên để thực hiện quy trình khai thác như sau:

$$X_{new} = \varepsilon A_i(t) + X_{old} \quad (4)$$

Trong đó ε là một số ngẫu nhiên thay đổi trong phạm vi $[-1, 1]$, và khi A lớn bầy dơi sử dụng độ lớn của âm thanh phát ra để thăm dò thay vì khai thác. Ngoài ra, do tập tính săn mồi của dơi, chúng sẽ tăng

tần số âm thanh siêu âm và giảm độ ồn khi săn đuổi con mồi. Do đó, các điều chỉnh sau được thực hiện cho hai yếu tố này:

$$A_i(t+1) = \alpha A_i(t) \quad (5)$$

$$r_i(t+1) = [1 - e^{-\gamma t}]r_i(0) \quad (6)$$

trong đó α và γ là các hằng số. Cuối cùng, A_i tiến tới 0, trong khi r_i đạt đến giá trị cuối cùng của nó là $r(0)$. Lưu ý rằng khi các giải pháp mới được đưa ra, cả tốc độ và âm lượng đều được sửa đổi để đảm bảo rằng những bầy dơi đang tiến dần tới điểm tối ưu tốt nhất. Như vậy, mức độ phát xung (r) và tiếng vang (A) chi phối sự cân bằng của phương pháp này.

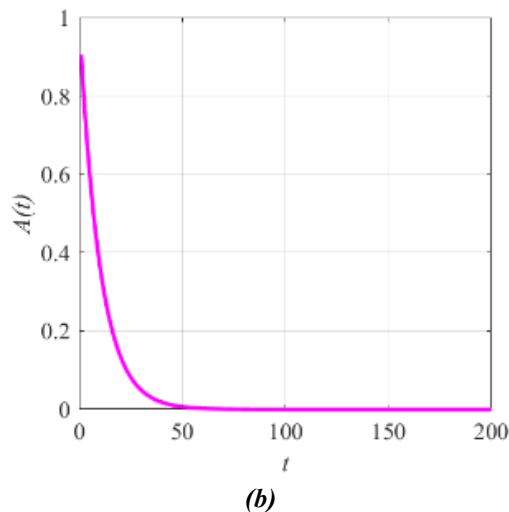
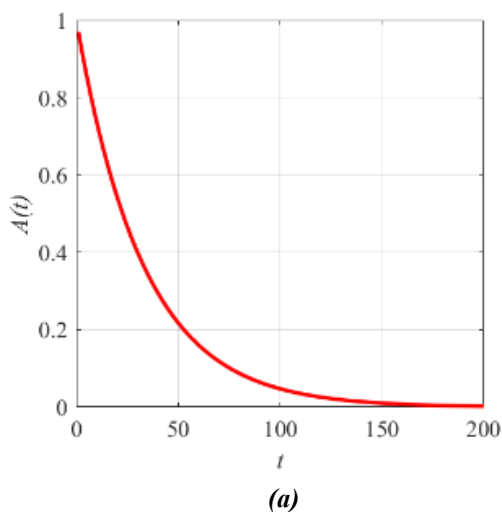
Quy trình mã giải thuật và lưu đồ giải thuật của thuật toán BA lần lượt được thể hiện trong Giải thuật 1 và Hình 1. Trong trường hợp này, rand là một số ngẫu nhiên phân phối từ 0 đến 1. Ngoài ra, bằng cách điều chỉnh tốc độ phát xung và tiếng vang, BA có thể tự động chuyển đổi từ các động thái thăm dò sang hoạt động khai thác tìm kiếm cục bộ, tập trung vào các vị trí tối ưu tiềm năng.

3. Di chuyển ngẫu nhiên

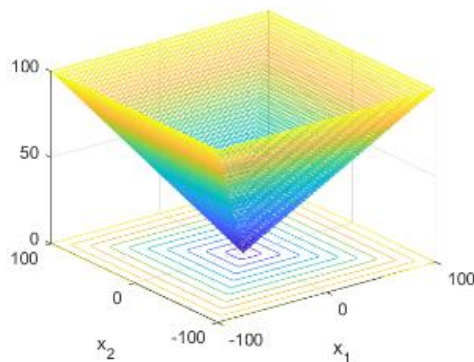
Di chuyển ngẫu nhiên (random walk) là đặc trưng của các thuật toán tìm kiếm tối ưu hiện nay. Trong nghiên cứu này, các tác giả giới thiệu một cách thức thay thế để điều chỉnh bước di chuyển ngẫu nhiên một cách hiệu quả, như sau:

$$A_i(t+1) = \frac{A_i(0)}{e^{\gamma t}} \quad (7)$$

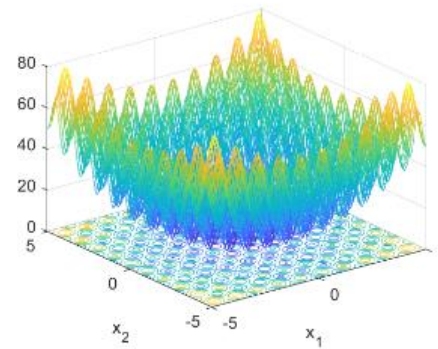
Đặc biệt, phương trình (7) được đề xuất cho phép kiểm soát quy trình chuyển mạch mạnh hơn bằng cách tăng phạm vi A và r .



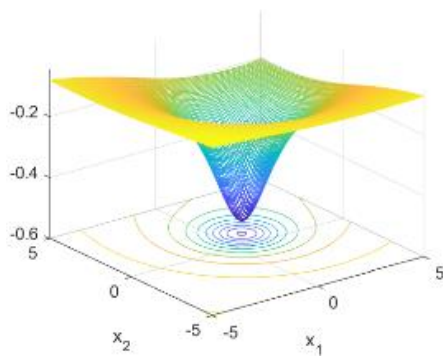
Hình 2. (a) Hàm kiểm soát tiếng vang cơ bản; (b) Hàm kiểm soát tiếng vang đề xuất



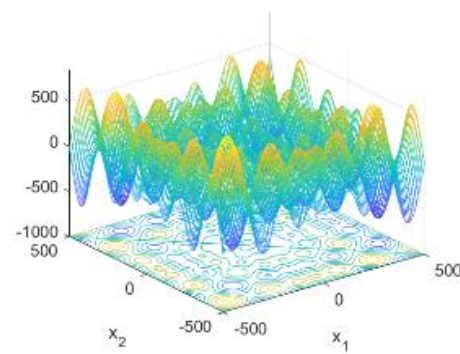
(a) Rosenbrock



(b) Rastrigin



(c) Shekel-10



(d) Schwefel

Hình 3. Các hàm kiểm chuẩn toán học

4. Các bài toán kiểm chuẩn

Trong phần này bốn hàm kiểm chuẩn toán học được lựa chọn sẽ được giới thiệu. Các hàm này được triển khai để đánh giá độ hội tụ, tỷ lệ chính xác, độ mạnh mẽ và hiệu suất của các thuật toán BA được cải tiến. Các hàm kiểm chuẩn này được lựa chọn từ tài liệu Tham khảo [34, 35]:

- Rosenbrock: $f_1(x) = \sum_{i=1}^{n-1} |100(x_{i+1} - x_i^2) + (x_i - 1)^2|$; phạm vi $[-100; 100]$; cực trị 0.
- Rastrigin: $f_2(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(\pi x_i) + 10]$; phạm vi $[-5; 5]$; cực trị 0;
- Shekel-10: $f_3(x) = -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$; kích thước 4; cực trị tại -10.5363;
- Schwefel: $f_4(x) = \sum_{i=1}^n -x_i \sin(\sqrt{|x_i|})$; phạm vi $[-500, 500]$; cực trị -418.9828 $\times n$.

Các hàm toán học kiểm chuẩn được chọn bao gồm các hàm đơn hoặc đa chiều [15, 19]. Trong đó các hàm đơn chiều chỉ sở hữu một nghiệm toàn cục và không có nghiệm cục bộ [15]. Do đó, chúng thường hữu ích trong việc kiểm tra tốc độ hội tụ của các thuật toán. Trong khi đó, các hàm đa chiều tồn tại nhiều nghiệm cục bộ thích hợp để đánh giá khả năng thoát nghiệm cục bộ của các thuật toán [15].

5. Kết quả kiểm chuẩn

Để tiến hành kiểm chuẩn, kết quả giải bài toán sẽ được thu thập thông qua 30 lần chạy độc lập với các thông số ban đầu được thiết lập như sau: Tiếng vang ban đầu $A = 1$, tốc độ xung $r = 1$, tần số $Q_i^t \in [0; 2]$, hằng số $\gamma = 0.1$, số cá thể bơi trong các lần tính là 30 con. Đồng thời, để đánh giá hiệu quả của thuật toán được đề xuất, năm chỉ số thống kê quan trọng được sử dụng, gồm: (1) Giá trị tốt nhất (Best): Giá trị tốt nhất của tập kết quả; (2) Giá trị tệ nhất (Worst): Giá trị kém nhất của tập kết quả; (3) Giá trị trung bình (Mean): Là giá trị trung bình của một tập dữ liệu cho thấy xu hướng hướng tâm của tập dữ liệu; (4) Độ lệch chuẩn (Standard deviation): phản ánh sự phân tán trong một tập dữ liệu, còn được gọi là giá trị kỳ vọng. Do đó, các thuật toán có độ lệch chuẩn càng nhỏ độ tin cậy càng cao; và (5) Giá trị xuất hiện thường xuyên nhất (Mode): Là thông số cho biết kết quả xuất hiện thường xuyên nhất trong tập dữ liệu. Nói cách khác, nó là giá trị xuất hiện thường xuyên nhất trong các kết quả tính của thuật toán. Nếu không có số nào trong bộ dữ liệu được lặp lại thì không tồn tại chỉ số này.

Bảng 1. Kết quả kiểm chuẩn thuật toán với các bài toán mẫu

f_i	Chỉ số thống kê	BA đề xuất	BA cơ sở
f_1	(1)	7.1484E+00	5.7126E+01
	(2)	4.2020E+05	3.1227E+03
	(3)	1.6938E+04	6.9615E+02
	(4)	7.6420E+04	8.3896E+02
	(5)	7.1484E+00	5.7126E+01
f_2	(1)	1.1939E+01	3.8628E+01
	(2)	8.6561E+01	8.1456E+01
	(3)	4.3612E+01	5.7846E+01
	(4)	1.9856E+01	1.0319E+01
	(5)	1.1939E+01	3.8628E+01
f_3	(1)	-1.0536E+01	-9.1443E+00
	(2)	-1.8595E+00	-1.5344E+00
	(3)	-4.5938E+00	-3.9496E+00
	(4)	3.3858E+00	2.3305E+00
	(5)	-1.0536E+01	-9.1443E+00
f_4	(1)	-2.2726E+03	-9.7337E+04
	(2)	-1.0487E+03	-1.4040E+03
	(3)	-1.6027E+03	-1.5308E+04
	(4)	2.7554E+02	1.9485E+04
	(5)	-1.7593E+03	-9.7337E+04

Kết quả của quy trình kiểm chuẩn thuật toán được thể hiện trong Bảng 1. Đồng thời kết quả của thuật toán còn được so sánh với kết quả của thuật toán BA cơ sở, xem Bảng 1. Kết quả cho thấy mặc dù không hoàn toàn vượt trội so với thuật toán cơ sở nhưng với phương thức đề xuất mới đã cung cấp các kết quả tối ưu tốt hơn. Dựa trên kết quả kiểm chuẩn này, bài toán tối ưu khung giàn thép được tiến hành ở phần tiếp theo của nghiên cứu.

6. Tối ưu khối lượng khung giàn thép

Khung giàn thép được sử dụng để triển khai thuật toán tối ưu trong nghiên cứu này là một cấu trúc giàn không gian gồm 25 thanh, xem Hình 4 [20]. Hệ 25 thanh của giàn được phân thành 8 nhóm x_i ($x_1, x_2, \dots, x_8$) với diện tích mặt cắt ngang giống nhau cho mỗi nhóm A_j ($A_1, A_2, \dots, A_8$). Như vậy, hàm mục tiêu của bài toán này là hàm của tổng khối lượng khung giàn thép:

$$f(x) = \sum_{j=1}^{25} \rho A_j L_j. \quad (8)$$

* 1lp=0.454kg; 1in=2.54m; 1psi=1lp/in²=6.8948kPa

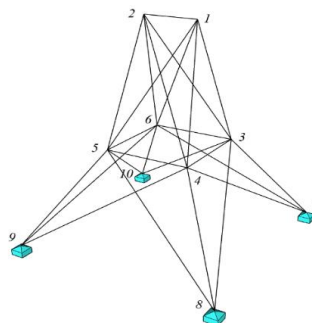
Toạ độ của các nút phần tử của giàn được cho trong Bảng 2. Khối lượng riêng của vật liệu cấu tạo khung giàn thép là $\rho = 0,11\text{lb/in}^3$ và mô đun đàn hồi là $E = 107\text{psi}^*$. Khung giàn chịu hai trường hợp tải trọng (xem Bảng 3), các giới hạn ứng suất cho phép đối với từng nhóm thanh giàn nêu trong Bảng 4 và chuyển vị cho phép của các nút là 0,35in theo các phương x , y và z [20]. Nội lực và các kết quả chuyển vị của khung giàn trong nghiên cứu này được tính toán bởi phương pháp phần tử hữu hạn và cũng được thực hiện mã hoá giải thuật bằng ngôn

Bảng 2. Toạ độ nút khung giàn

Nút	x (in)	y (in)	z (in)
1	-37.5	0	200
2	37.5	0	200
3	-37.5	37.5	100
4	37.5	37.5	100
5	37.5	-37.5	100
6	-37.5	-37.5	100
7	-100	100	0
8	100	100	0
9	100	-100	0
10	-100	-100	0

Bảng 3. Các tải trọng tác dụng lên giàn

Tải trọng	Nút	P_x (kip)*	P_y (kip)	P_z (kip)
1	1	1	-10	-10
	2	0	-10	-10
	3	0.5	0	0
	6	0.6	0	0
2	1	0	20	-5
	2	0	-20	-5



Hình 4. Giàn không gian 25 thanh [20]

ngữ lập trình trên phần mềm MATLAB. Phạm vi lựa chọn tiết diện mặt cắt ngang của thanh là $[0,01, 3,40] \text{in}^2$.

Kết quả tính toán tối ưu bằng thuật toán BA của nghiên cứu này cho thấy, khối lượng thiết kế tối ưu của giàn thép là 545,21lb, xem Bảng 5. Kết quả này hoàn toàn tương thích so với kết quả của thuật toán gen di truyền (GA) với kết quả đối sánh là 545,8lb. Đồng thời các kết quả về tiết diện tối ưu của các thanh giàn cũng tương ứng sát với kết quả của thuật GA.

Bảng 5. Kết quả tối ưu tiết diện khung giàn thép

x_i	PBA (in^2)	GA (in^2) [21]
1	0,013	0,010
2	2,007	2,012
3	2,969	2,949
4	0,010	0,010
5	0,010	0,030
6	0,671	0,684
7	1,672	1,680
8	2,687	2,676
Khối lượng	545,21lb	545,8lb

7. Kết luận và đề xuất

Như vậy, bài báo này đã giới thiệu thành công một biến thể của kỹ thuật của bước di chuyển ngẫu nhiên để tăng cường hiệu quả của thuật toán BA trong việc tối ưu khối lượng khung giàn thép. Bên cạnh đó, dựa theo kết quả của nghiên cứu này chúng ta có thể áp dụng thuật toán cho các lĩnh vực tối ưu hóa khác nhau của các bài toán kỹ thuật.

Lời cảm ơn

Nghiên cứu này được tài trợ bởi Trường Đại học Hàng hải Việt Nam trong đề tài mã số: **DT22-23.75**.

TÀI LIỆU THAM KHẢO

[1] Fister, I., *A comprehensive review of bat algorithms and their hybridisation* (Doctoral dissertation, Master's thesis, University of Maribor, Slovenia). 2013.

[2] Mirjalili, S. and A. Lewis (2016), *The whale optimisation algorithm*, Advances in engineering software, Vol.95: pp.51-67.

[3] Rashedi, E., H. Nezamabadi-Pour, and S. Saryazdi, GSA (2009): *A gravitational search algorithm*, Information sciences, Vol.179(13): pp.2232-2248.

[4] Hatamlou, A., Black hole (2013): *A new heuristic optimisation approach for data clustering*, Information sciences, Vol.222: pp.175-184.

[5] Moghaddam, F.F., R.F. Moghaddam, and M. Cheriet (2012), *Curved space optimisation: A random search based on general relativity theory*. arXiv preprint arXiv:1208.2214.

[6] Holland, JH (1992), *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.

[7] Rechenberg, I. (1978), *Evolutionsstrategien, in Simulationsmethoden in der Medizin und Biologie*, Springer, pp.83-114.

[8] Koza, J.R (1992). *Evolution of subsumption using genetic programming*. in Proceedings of the first European conference on artificial life. MIT Press Cambridge, MA.

[9] Kennedy, J. and R. Eberhart (1995). *Particle swarm optimisation*. in Proceedings of ICNN'95-international conference on neural networks. IEEE.

[10] Dorigo, M., V. Maniezzo, and A. Colnani (1996), *Ant system: optimisation by a colony of cooperating agents*. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), Vol.26(1): pp.29-41.

[11] Yang, X.-S. and X. He (2013), *Bat algorithm: literature review and applications*, International Journal of Bio-inspired computation, Vol.5(3), pp. 141-149.

[12] Yang, X.-S., *Nature-inspired optimization algorithms*. 2020: Academic Press.

[13] Jamil, M., H. Zepernic, and XJASC Yang (2013), *Improved bat algorithm for global optimisation*.

[14] Nakamura, R.Y., et al (2012), *BBA: a binary bat algorithm for feature selection*. in 2012 25th SIBGRAPI conference on graphics, patterns and images, IEEE.

[15] Mirjalili, S., S.M. Mirjalili, and X.-S. Yang (2014), *Binary bat algorithm*. Neural Computing and Applications, Vol.25(3), pp.663-681.

[16] Yang, X.-S., *A new metaheuristic bat-inspired algorithm*, in Nature inspired cooperative strategies for optimisation (NICSO 2010). 2010, Springer. pp.65-74.

- [17] Geem, Z.W., J.H. Kim, and G.V.J.s (2001). Loganathan, A new heuristic optimisation algorithm: harmony search. Vol.76(2), pp.60-68.
- [18] Beyer, H.-G. and H.-P. Schwefel (2002), Evolution strategies-a comprehensive introduction. Natural computing, Vol.1(1): pp.3-52.
- [19] Seyyedabbasi, A. and F.J.E.w.C. Kiani (2021), *I-GWO and Ex-GWO: improved algorithms of the Grey Wolf Optimizer to solve global optimisation problems*. Vol.37(1), pp.509-532.
- [20] Camp, CV and M. Farshchin (2014), *Design of space trusses using modified teaching-learning based optimisation*, Engineering Structures, Vol.62, pp.87-97.
- [21] Cao, G. (1996), *Optimised design of framed structures using a genetic algorithm*, The University of Memphis.

Ngày nhận bài:	20/10/2022
Ngày nhận bản sửa:	02/11/2022
Ngày duyệt đăng:	07/11/2022