

NGHIÊN CỨU ĐÁNH GIÁ THUẬT TOÁN ĐOM ĐÓM VÀ PHIÊN BẢN LẠI VỚI PSO TRONG TỐI ƯU KHỐI LƯỢNG KHUNG GIÀN THÉP KHÔNG GIAN

AN APPLICATION OF THE FIREFLY ALGORITHM AND ITS HYBRID
VARIANT WITH PSO FOR OPTIMISATION OF THE 3D TRUSS STRUCTURES

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 nhằm giới thiệu và đánh giá hiệu quả tính toán tối ưu bài toán khung giàn thép không gian của thuật toán mô phỏng nổi tiếng là thuật toán đom đóm (Firefly Algorithm - FA), thuật toán PSO (Particle Swarm Optimisation) và thuật toán lai tạo giữa PSO và FA, được ký hiệu là HFPSO. Việc lai tạo giữa hai thuật toán nhằm khai thác được sức mạnh ưu thế của cả hai thuật toán và giảm thiểu được nhược điểm của chúng từ đó tạo ra thuật toán mới với sức mạnh tính toán vượt trội hơn. Các thuật toán được giới thiệu trong nghiên cứu này sẽ trước hết được triển khai tính toán với một số các bài toán mẫu (mathematical benchmark) nổi tiếng nhằm mục đích kiểm chuẩn và đánh giá hiệu quả tính toán của các thuật toán. Và sau cùng các thuật toán sẽ được ứng dụng để tối ưu hóa trọng lượng của kết cấu giàn thép nhằm đánh giá và so sánh hiệu quả của các thuật toán.

Từ khóa: Tối ưu hóa, khung giàn thép, thuật toán lai, thuật toán đom đóm, thuật toán PSO.

Abstract

The purpose of this research is to present and compare the performance of well-known natural-inspired algorithms for solving 3D steel truss optimisations, including the Firefly Algorithm (FA), the Particle Swarm Optimisation (PSO), and the hybrid algorithm of PSO and FA, as HFPSO. Developing a new algorithm by combining two others aims to improve the original algorithms' computational efficiency while avoiding their drawbacks. To test and assess the computing efficiency of the methods presented in this work, they will first be applied to a large set of well-known mathematical benchmark problems. Finally, the algorithms will be applied to optimise the 3D steel truss structure's weight to provide the

final evaluation and comparison of their performance.

Keywords: Optimisation, 3D steel truss, hybrid algorithm, firefly algorithm, PSO.

1. Mở đầu

Trên 40 là con số thống kê của Iztok Fister Jr. và cộng sự về số lượng các thuật toán lấy cảm hứng từ tự nhiên được phát triển và giới thiệu cho đến năm 2013 [1]. Những năm gần đây sự phát triển đó còn trở lên nhanh chóng và mạnh mẽ hơn với hàng trăm các thuật toán mới, các phiên bản cải tiến và các phiên bản lai..., với những sức mạnh tính toán ngày càng mạnh hơn để phục vụ cho các vấn đề kỹ thuật ngày càng phức tạp và đa dạng hơn. Mục tiêu cải tiến và phát triển các phiên bản mới của các thuật toán nhằm gia tăng hiệu quả tính toán của các thuật toán từ đó cho phép tìm kiếm các kết quả của các bài toán tối ưu trong thực tế một cách chính xác và nhanh chóng hơn.

Đối với các thuật toán lấy cảm hứng từ tự nhiên, cơ bản có thể phân chia làm ba nhóm dựa trên nguyên tắc sao chép các hiện tượng tính chất trong tự nhiên, gồm: (1) nhóm các thuật toán dựa trên quy luật và tập quán của các nhóm quần thể sinh học gọi tắt là bầy đàn, ví dụ: thuật toán bầy đàn (PSO) [2], thuật toán đàn kiến (Ant Colony Optimisation - ACO) [3], và thuật toán bầy dơi (Bat Algorithm - BA) [4],...; (2) nhóm các thuật toán dựa trên các quy luật tiến hóa, như: thuật toán gen di truyền (Genetic Algorithm - GA) [5], thuật toán chiến lược tiến hóa (Evolution Strategy - ES) [6] và thuật toán lập trình gen di truyền (Genetic Programming - GP) [7],...; và (3) nhóm mô phỏng quy luật hiện tượng vật lý, như: thuật toán trọng lực (Gravitational Search Algorithm - GSA) [8] và thuật toán hố đen (Black Hole - BH) [9],... [10]. Bên cạnh các nhóm thuật toán gốc nêu trên, hiện nay có xu hướng nghiên cứu nhằm lai tạo các thuật toán với nhau tận dụng các điểm mạnh và giảm thiểu các nhược điểm của thuật toán gốc. Và kết quả là giúp gia tăng hiệu quả về thời gian tính toán và/hoặc là chất lượng của các phiên bản lai [11]. Cách thức lai tạo có thể

thực hiện một cách linh hoạt bằng cách kết hợp các thuật toán giữa các nhóm với nhau hoặc giữa các thuật toán trong cùng một nhóm.

Trong bài báo này, hai thuật toán nổi tiếng là thuật toán FA và PSO sẽ được nghiên cứu cùng với một phiên bản lai giữa chúng. Đối với phiên bản lai, kỳ vọng rằng khả năng hội tụ nhanh của thuật toán PSO [2] kết hợp với sức mạnh tìm kiếm của thuật toán đom đóm (FA) [12] sẽ được kết hợp [13]. Sức mạnh của các thuật toán được đề xuất nghiên cứu trong bài báo này sẽ trước hết được kiểm chuẩn thông qua các bài toán mẫu và sau đó sẽ được áp dụng để thực hiện tính toán tối ưu hoá khối lượng cho bài toán giàn thép không gian.

Cấu trúc của bài báo, gồm: Phần 2 giới thiệu thuật toán PSO. Phần 3 giới thiệu thuật toán đom đóm. Phần 4 giới thiệu phiên bản thuật toán lai. Phần 5 và 6 các bài toán mẫu sử dụng được sử dụng để kiểm chuẩn các thuật toán. Và phần 7 cung cấp các đánh giá và phân tích cho các kết quả áp dụng của thuật toán trong tối ưu khối lượng khung giàn thép không gian. Và cuối cùng, các kết luận và đề xuất được đưa ra trong Phần 8.

2. Cấu trúc thuật toán PSO

PSO là một kỹ thuật tối ưu hóa dựa trên tập tính sinh sống và thói quen săn mồi của một quần thể sinh học, như: các loài chim và cá. Thuật toán này được đề xuất bởi nhà tâm lý học xã hội người Mỹ James Kennedy và kỹ sư Russell C. Eberhart [2] vào năm 1995, xem Hình 1. Về mặt toán học, vị trí X_i và vận tốc V_i của từng cá thể khi thực hiện tìm kiếm săn mồi như sau:

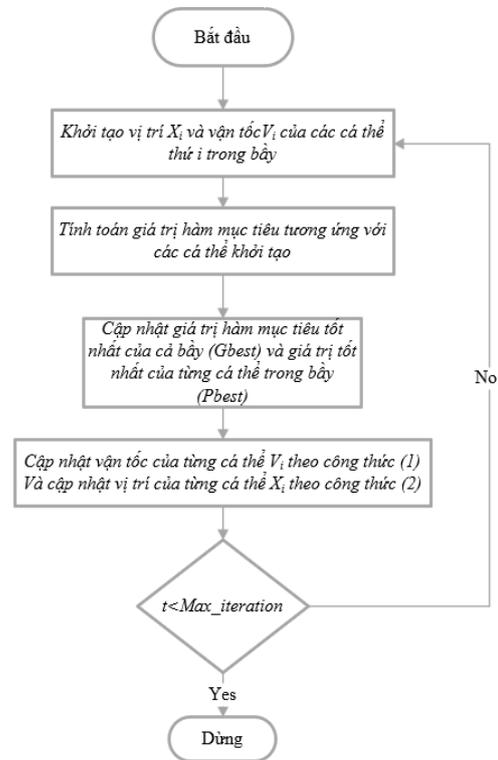
$$V_i(t+1) = wV_i(t) + c_1r_1(Pbest_i(t) - X_i(t)) + c_2r_2(Gbest_i(t) - X_i(t)) \quad (1)$$

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

Trong đó, c_1, c_2 là các gia số; r_1, r_2 là các số ngẫu nhiên trong khoảng $[0, 1]$; w là trọng số điều khiển vận tốc tìm kiếm của các cá thể - được đề xuất trong [14]; i là chỉ số của cá thể thứ i trong đàn; t là vòng lặp tính toán thứ t ; $Pbest$ là giá trị hàm mục tiêu tốt nhất đạt được của từng cá thể trong bầy ở vòng tính toán thứ t ; $Gbest$ là giá trị tốt nhất của hàm dạng của cả đàn ở vòng tính toán thứ t .

Theo như (1) và (2), chúng ta thấy rằng PSO có cơ chế đặc biệt trong việc sử dụng và trao đổi thông tin giữa các cá thể trong bầy phục vụ cho quá trình tìm kiếm (dựa trên thông tin về $Pbest$ và $Gbest$) điều này giúp PSO có thể giải quyết các vấn đề khác nhau với hiệu suất tốt và chi phí tính toán thấp [15]. Đây chính là một ưu thế, một cách thức tiếp cận gián tiếp việc xử

lý toán học nhằm tăng cường khả năng tính toán tối ưu. Điều này giúp PSO trở thành một trong số các thuật toán được ưa thích nhất do cần ít chi phí tính toán và dễ thực hiện.



Hình 1. Sơ đồ giải thuật PSO [2]

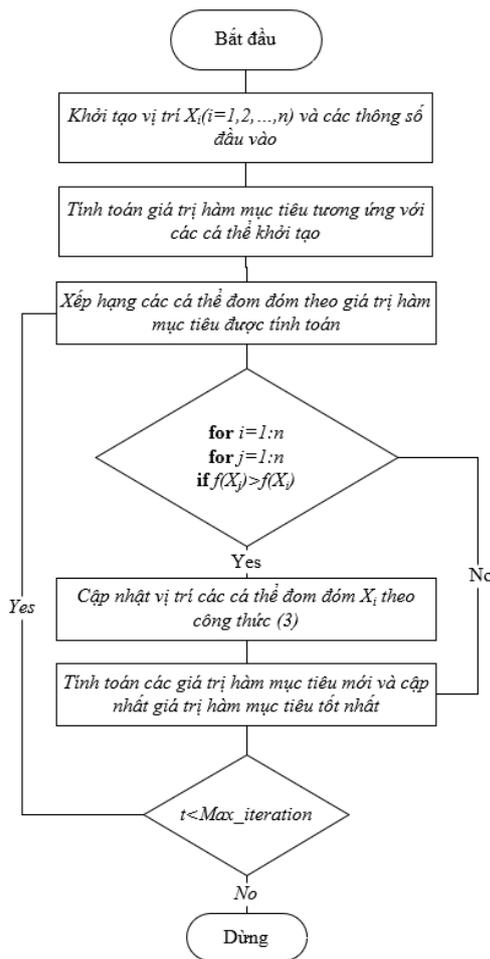
3. Cấu trúc thuật toán FA

Vào đầu năm 2008, thuật toán FA được lần đầu giới thiệu bởi Xin-She Yang và cho đến nay đã có hàng trăm ấn phẩm liên quan đến thuật toán này [15]. Đây là một thuật toán lấy cảm hứng từ thiên nhiên bắt nguồn từ tập quán của loài đom đóm trong tự nhiên. Thuật toán này mang một số ưu điểm so với PSO [16], là: (1) FA không sử dụng đơn nhất giá trị tối ưu toàn cục giúp ngăn ngừa việc mắc kẹt ở các vị trí tối ưu cục bộ dẫn tới hội tụ sớm và sai; (2) thêm nữa FA không sử dụng đặc tính vận tốc giúp cho các vấn đề liên quan đến bầy giá trị cục bộ do sự nhanh và chậm của vận tốc có thể được ngăn chặn [17].

Đom đóm sử dụng quá trình phát quang sinh học nhằm tạo ra ánh sáng nhấp nháy với chức năng để cảnh báo, giao tiếp và săn mồi [16]. Sự di chuyển của chúng bị dẫn dắt bởi các nguồn sáng xung quanh của các cá thể khác trong bầy và được tính toán như sau:

$$X_i(t+1) = X_i(t) + \beta_0 e^{-\gamma r_{ij}^2} (X_j(t) - X_i(t)) + \alpha \epsilon_i(t) \quad (3)$$

Trong đó, β_0 hệ số mức hấp dẫn của nguồn sáng



Hình 2. Sơ đồ giải thuật FA [15]

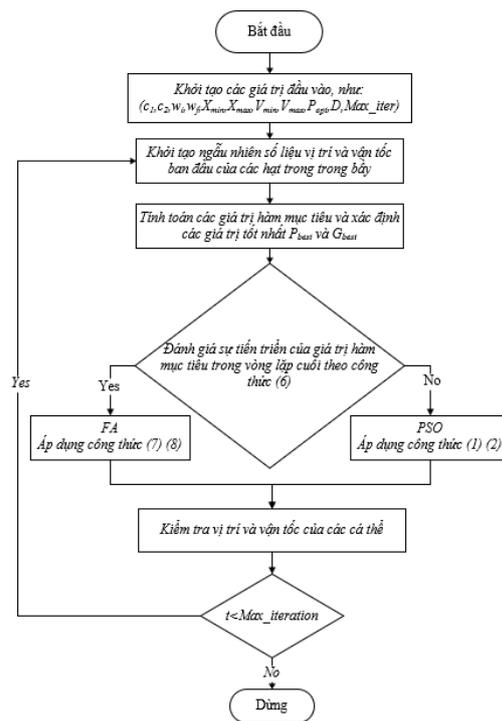
tại khoảng cách với nguồn sáng $r = 0$; α là hệ số ngẫu nhiên phân phối trong khoảng $[0,1]$; γ là hằng số hấp thụ ánh sáng và $\epsilon_i(t)$ là vecto ngẫu nhiên được phân phối bởi phân phối Gaussian. Sơ đồ giải thuật của FA được mô tả trong Hình 2. Khoảng cách giữa hai cá thể đom đóm được tính bởi công thức Euclidean (4):

$$r_{ij} = \|X_i - X_j\| = \sqrt{(X_i - X_j)^2} \quad (4)$$

4. Thuật toán lai HFPSO

Trong phần này sẽ giới thiệu một phiên bản lai của hai thuật toán FA và PSO - được ký hiệu là HFPSO, xem [13]. Cách thức lai này nhằm cân bằng giữa hai quy trình thăm dò và khai thác của thuật toán. Với phiên bản lai này kỳ vọng rằng thuật toán lai sẽ tận dụng được ưu điểm của PSO cho quy trình thăm dò do ưu điểm về tốc độ hội tụ nhanh và ưu điểm FA trong tìm kiếm mục tiêu cho quy trình khai thác [13].

Giải thuật của thuật toán HFPSO được trình bày trong Hình 3. Trước hết, các tham số đầu vào được sử dụng bởi cả hai thuật toán PSO và FA được khởi tạo.



Hình 3. Sơ đồ giải thuật HFPSO [13]

Tiếp theo, khởi tạo giá trị vận tốc và vị trí của các cá thể một cách ngẫu nhiên trong phạm vi tìm kiếm từ đó xác định các giá trị hàm mục tiêu tốt nhất toàn cục G_{best} và giá trị hàm mục tiêu tốt nhất của từng cá thể trong bầy P_{best} . Trong các vòng tính toán kế tiếp, nếu có sự cải thiện về giá trị hàm mục tiêu theo thì vị trí hiện tại được lưu trong một biến tạm thời X_{temp} và vị trí và vận tốc mới được tính theo CT. (7) và (8).

$$w = w_i - \left(\frac{w_i - w_f}{Max_{iter}}\right) \times iter \quad (5)$$

$$f(i, t) = \begin{cases} Yes, & f(X_i^t) \leq G_{best}^{t-1} \\ No, & f(X_i^t) > G_{best}^{t-1} \end{cases} \quad (6)$$

$$X_i(t+1) = X_i(t) + \beta_0 e^{-\gamma r_{ij}^2 (X_i(t) - G_{best}^{t-1})} + \alpha \epsilon_i \quad (7)$$

$$V_i(t+1) = X_i(t+1) - X_{temp} \quad (8)$$

Lúc này, nếu cá thể có giá trị hàm mục tiêu tốt hơn hoặc bằng giá trị toàn cục tốt nhất trước đó thì giả định rằng quá trình khai thác bắt đầu thì các cá thể sẽ được tính theo FA. Ngược lại, các cá thể sẽ được xử lý bởi PSO. Trong quy trình tiếp theo về đánh giá giá trị hàm mục tiêu và giới hạn phạm vi được thực hiện đối với tất cả các cá thể. Nếu đạt đến giới hạn lặp tối đa, thuật toán sẽ dừng và G_{best} sẽ được đưa ra làm kết quả của bài toán.

Bảng 1. Các hàm kiểm chuẩn toán học

	Hàm kiểm chuẩn toán học	Số biến tính	Phạm vi	Giá trị tối ưu
Sphere	$f_1(x) = \sum_{i=1}^n x_i^2$	30	[-100, 100]	0
Schwefel 2.22	$f_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	[-10, 10]	0
Schwefel 1.2	$f_3(x) = \sum_{i=0}^{n-1} \left(\sum_{j=0}^{i-1} x_j \right)^2$	30	[-100, 100]	0
Step	$f_4(x) = \sum_{i=1}^n (x_i + 0.5)^2$	30	[-100, 100]	0
QuarticWN	$f_5(x) = \sum_{i=1}^n ix_i^4 + random[0,1]$	30	[-1.28, 1.28]	0
Rastrigin	$f_6(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	[-5.12, 5.12]	0
Griewangk	$f_7(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[-600, 600]	0
Six-Hump Camel-Back	$f_8(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^2$	2	[-5, 5]	-1.0316

5. Hàm toán học kiểm chuẩn

Phân mục này của bài báo nhằm giới thiệu các bài toán kiểm chuẩn được lựa chọn từ tài liệu [18, 19]. Các bài toán kiểm chuẩn này được sử dụng nhằm đánh giá khả năng hội tụ, mức độ chính xác và hiệu suất tính toán của các thuật toán. Các bài toán kiểm chuẩn được lựa chọn và giới thiệu trong nghiên cứu này, gồm các hàm đơn trị, đa trị [18]. Các bài toán đơn trị chỉ sở hữu một nghiệm toàn cục và không có nghiệm cục bộ [18] nên thích hợp trong việc kiểm tra tốc độ hội tụ của thuật toán. Trong khi đó, các hàm đa trị tồn tại nhiều cực trị cục bộ thích hợp để đánh giá khả năng tránh và thoát khỏi bẫy lời giải cục bộ (các điểm cực trị cục bộ) của các thuật toán [18].

6. Kiểm chuẩn các thuật toán

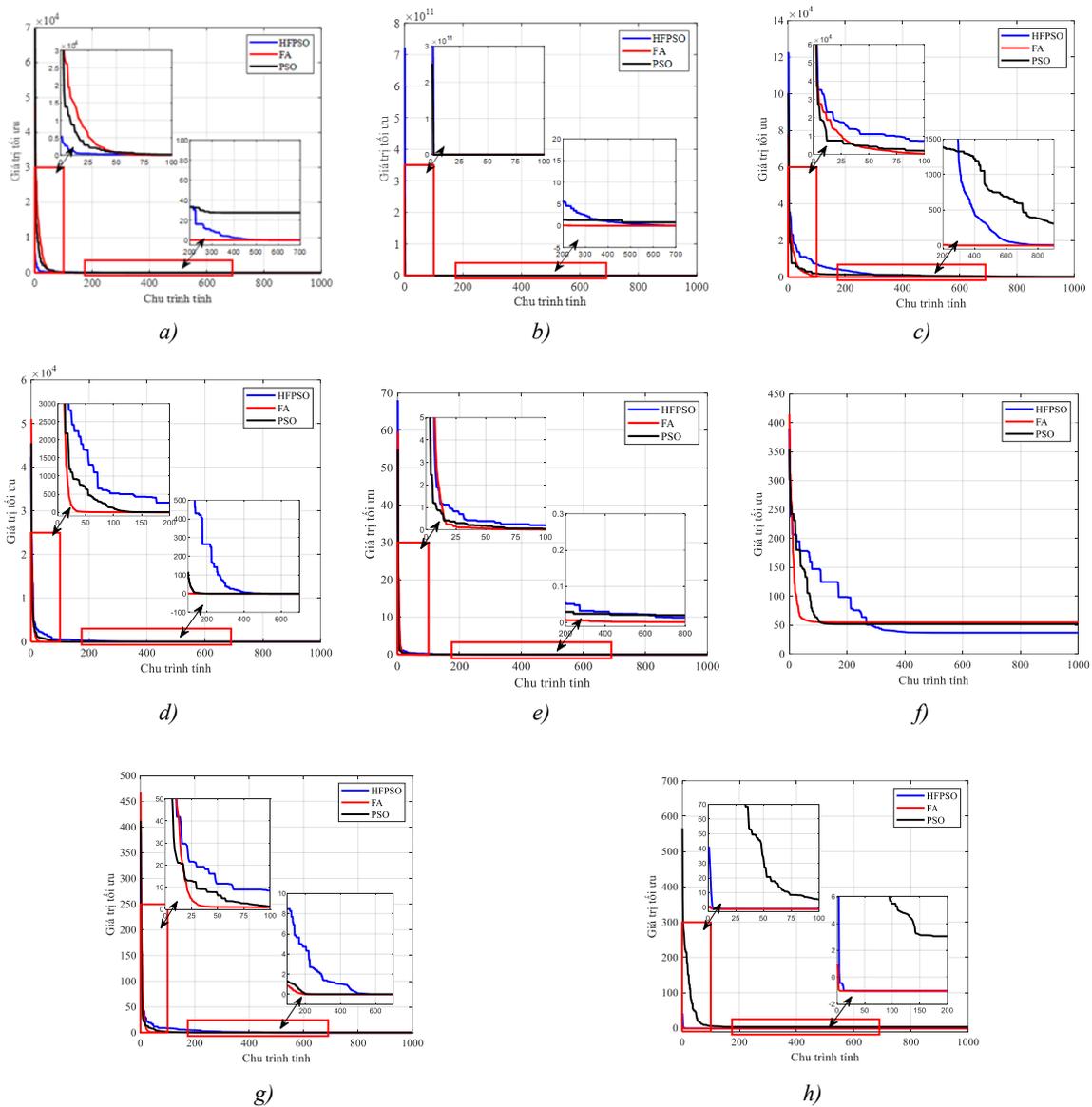
Trước hết để đánh giá hiệu quả tính toán của thuật toán FA, PSO và phiên bản lai HFPSO được giới thiệu trong bài báo này, các thuật toán được triển khai để tìm kiếm các giá trị tối ưu cho các hàm kiểm chuẩn. Các thông số đầu vào của các thuật toán được lấy như sau, Bảng 2:

Bảng 2. Các thông số đầu vào

Thuật toán	Thông số	Giá trị
FA	Số lượng cá thể	= Số biến tính
	β_0	2
	γ	1
	α	[0,1]
PSO	Số lượng cá thể	= Số biến tính
	r_1, r_2	[0,1]
	c_1, c_2	2
	w	0.9

Kết quả bước kiểm chuẩn này được thể hiện trong Hình 4. Đây là kết quả có được của ba thuật toán thực hiện với các hàm kiểm chuẩn đề xuất trong Bảng 1. Số lượng cá thể tính được sử dụng bằng với số lượng biến của bài toán kiểm chuẩn. Căn cứ theo kết quả được thể hiện trong biểu đồ của Hình 4, có thể thấy rằng thuật toán HFPSO đã tận dụng được ưu điểm cũng như khắc phục được nhược điểm của hai thuật toán gốc là PSO và FA. Thuật toán lai HFPSO đã có tốc độ hội tụ nhanh của PSO và có độ chính xác của thuật toán FA. Điều này được thấy rõ trong kết quả tính toán được mô tả trong Hình 4a), vốn là kết quả tính toán cho hàm kiểm chuẩn Sphere (bài toán hàm kiểm chuẩn đơn trị thích hợp để đánh giá hiệu suất về tốc độ hội tụ cũng như độ chính xác của phương pháp đề xuất). Cụ thể, trong kết quả này thuật toán HFPSO đã tiến nhanh vào vùng hội tụ chỉ sau khoảng 100 chu trình tính gần như tương đương với PSO. Thêm vào đó, kết quả của thuật toán HFPSO hoàn toàn vượt qua PSO và tiến tới kết quả chính xác của hàm kiểm chuẩn cũng như kết quả của thuật toán FA.

Đối với các kết quả còn lại thể hiện trong Hình 4 b-h), có thể thấy tốc độ hội tụ của thuật toán lai HFPSO không hoàn toàn vượt trội so với thuật toán PSO. Nguyên nhân của việc này là do các bài toán kiểm chuẩn tiếp theo không còn là các bài toán đơn trị mà là đa trị (tồn tại các điểm cực trị địa phương/ cục bộ) khiến cho thuật toán có thể bị bẫy ở các điểm lời giải cục bộ này. Tuy nhiên, có thể khẳng định rằng thuật toán lai HFPSO có hiệu suất tính toán là tương đương với thuật toán gốc FA và có xu hướng vượt qua hoàn toàn PSO. Minh chứng rõ ràng khi quan sát kết quả của Hình 4f). Như vậy, mặc dù thuật toán PSO được nhận định là có tốc độ hội tụ nhanh nhưng khi áp dụng với các hàm kiểm chuẩn đa trị ưu thế này không được rõ ràng. Trong



Hình 4. So sánh hiệu suất tính toán của các thuật toán

khi đó, thuật toán gốc FA cho thấy khả năng hoạt động rất hiệu quả cả về tốc độ hội tụ và độ chính xác đối với các hàm kiểm chuẩn đa trị.

Như vậy thông qua kết quả của bước kiểm chuẩn thứ nhất có thể nhận định rằng, thuật toán lai HFPSO có tận dụng được ưu điểm của hai thuật toán gốc, tuy nhiên cần lưu ý rằng, điều này không có nghĩa thuật toán lai hoàn toàn vượt trội so với hai thuật toán FA và PSO. Chính vì vậy, ở bước kiểm chuẩn kế tiếp, các thuật toán sẽ được triển khai tính toán với 30 lần tính độc lập cho mỗi bài toán kiểm chuẩn. Và để thực hiện việc đánh giá, các chỉ số thống kê (CSTK) sau được sử dụng: (1) Giá trị tốt nhất (Best): Giá trị tốt nhất mà tập kết quả có được; (2) Giá trị tệ nhất (Worst): Giá trị kém nhất mà bộ kết quả thu được; (3) Giá trị trung

bình (Mean): Là giá trị trung bình của tập kết quả; (4) Độ lệch chuẩn (Standard deviation): Phản ánh sự phân tán trong một tập dữ liệu, còn được gọi là giá trị kỳ vọng và (5) Giá trị xuất hiện thường xuyên nhất (Mode): Là thông số cho biết số xuất hiện phổ biến nhất trong tập dữ liệu kết quả.

Đây là bước kiểm chuẩn dựa trên phương thức thống kê nhằm nghiên cứu và đánh giá hiệu suất tính toán của các thuật toán có thực sự ổn định. Bước kiểm chuẩn này được yêu cầu thực hiện là bởi các thuật toán lấy cảm hứng từ tự nhiên luôn có tính ngẫu nhiên trong quy trình tính dẫn đến kết quả tính sau mỗi lần tính là có thể khác biệt thậm chí có sai lệch lớn. Một thuật toán lấy cảm hứng từ tự nhiên được coi là mạnh nếu các CSTK cho thấy sự đồng nhất thông qua các

kết quả của 30 lần tính. Kết quả của bước kiểm chuẩn này được thể hiện trong Bảng 3.

Căn cứ theo kết quả trong Bảng 3, cho thấy thuật toán lai HFPSO đã một lần nữa minh chứng cho những nhận định đã đề cập ở trên về hiệu suất của phiên bản lai này. Thuật toán có hiệu suất tốt nhất là thuật toán FA. Để tiếp tục phân tích và đánh giá các thuật toán, bài kiểm tra tiếp theo của nghiên cứu này là so sánh chi phí tính toán về thời gian của các thuật toán. Kết quả của bước đánh giá này được thể hiện trong Bảng 4.

Căn cứ theo kết quả trong Bảng 4, cho thấy thời gian tính toán của thuật toán FA gấp rất nhiều lần so với hai thuật toán còn lại. Kể đến là PSO cũng có thời gian tính toán tương đối lớn so với thuật toán lai HFPSO.

7. Tối ưu khối lượng giàn thép

Giàn thép đầu tiên được triển khai tính toán trong nghiên cứu này là một giàn phẳng được cấu thành bởi 10 thanh, xem Hình 5, [19]. Để tính toán tối ưu cho khung giàn thép, một phương trình mục tiêu là hàm số của diện tích mặt cắt ngang của các thanh được thành lập như sau:

$$f(x) = \sum_{j=1}^{10} \rho A_j L_j. \quad (9)$$

Các thông số ràng buộc cho bài toán tối ưu khối lượng khung giàn phẳng được thiết lập như sau: (1) ứng suất cho phép của mỗi thanh, σ_{max} , là ± 25 ksi; (2) chuyển vị cho phép của mỗi nút giàn, d_{max} , là ± 2.0 in. theo hai phương x, y ; (3) khối lượng riêng, ρ , và mô đun đàn hồi, E , lần lượt là $0,11\text{lb/in.}^3$ và 10^7 psi; (4) tải trọng tác dụng tại hai nút 2 và 4 là 10kips.

Kết quả được thể hiện trong Bảng 5, là kết quả tốt nhất được lựa chọn sau 30 lần tính toán độc lập của mỗi thuật toán, với số lượng cá thể tính là 30. Căn cứ theo kết quả, có thể kết luận rằng giá trị tối ưu tính toán được bởi cả ba thuật toán là gần như không có sự sai khác mặc dù giá trị tối ưu tốt nhất thuộc về thuật toán FA (5060,9lb). Và kết quả này là tương đồng với các kết quả của các nghiên cứu khác, như: 5061,65lb của PBA [19], 5060,92 của HPSO [20], 5060,88 của ABC-AP [21],... Tuy nhiên, nếu xét trên thời gian tính toán, thuật toán HFPSO lại có thời gian tính toán tốt nhất trong ba thuật toán với chỉ 2,4s, xem Bảng 5. Như vậy, nếu xét đồng thời về tính chính xác và hiệu quả tính, thuật toán HFPSO trong trường hợp này là thuật toán tốt nhất.

Và cuối cùng, một giàn thép không gian sẽ được sử dụng để triển khai tính toán tối ưu trong nghiên cứu này. Đây là một cấu trúc giàn không gian gồm 25 thanh, xem Hình 6 [22]. Tuy nhiên, thay vì sử dụng 25

Bảng 3. Kết quả kiểm chuẩn thuật toán

f_i	CSTK	HFPSO	PSO	FA
f_1	(1)	7.889E-10	0.00E+00*	0.000E+00
	(2)	1.658E-08	3.95E-12	2.206E-16
	(3)	5.898E-09	1.32E-13	7.353E-18
	(4)	4.385E-09	7.21E-13	4.027E-17
	(5)	7.889E-10	0.00E+00	0.000E+00
f_2	(1)	3.864E-05	1.952E-63	1.824E-263
	(2)	1.752E-04	4.336E-03	6.016E-09
	(3)	8.271E-05	1.446E-04	2.005E-10
	(4)	3.339E-05	7.915E-04	1.098E-09
	(5)	3.864E-05	1.952E-63	1.824E-263
f_3	(1)	6.374E-01	1.952E-63	0.000E+00
	(2)	1.654E+01	4.336E-03	3.328E-09
	(3)	6.951E+00	1.446E-04	1.109E-10
	(4)	4.753E+00	7.915E-04	6.076E-10
	(5)	6.374E-01	1.952E-63	0.000E+00
f_4	(1)	0.000E+00	0.000E+00	0.000E+00
	(2)	0.000E+00	5.613E-12	1.558E-16
	(3)	0.000E+00	1.871E-13	5.193E-18
	(4)	0.000E+00	1.025E-12	2.844E-17
	(5)	0.000E+00	0.000E+00	0.000E+00
f_5	(1)	4.937E-03	6.305E-04	1.152E-04
	(2)	1.935E-02	1.901E-02	2.813E-03
	(3)	1.034E-02	2.607E-03	6.083E-04
	(4)	3.875E-03	3.661E-03	5.965E-04
	(5)	4.937E-03	1.066E-03	1.420E-04
f_6	(1)	2.686E+01	2.288E+01	3.980E+01
	(2)	1.094E+02	2.288E+01	3.980E+01
	(3)	5.539E+01	2.288E+01	3.980E+01
	(4)	2.317E+01	3.239E-11	3.613E-14
	(5)	2.686E+01	2.288E+01	3.980E+01
f_7	(1)	3.776E-10	3.935E-02	0.000E+00
	(2)	7.617E-02	3.935E-02	1.110E-16
	(3)	9.679E-03	3.935E-02	3.701E-18
	(4)	1.564E-02	7.826E-15	2.027E-17
	(5)	3.776E-10	3.935E-02	0.000E+00
f_8	(1)	-1.032E+00	-1.03E+00	-1.03E+00
	(2)	-2.155E-01	-1.03E+00	-1.03E+00
	(3)	-8.684E-01	-1.03E+00	-1.03E+00
	(4)	3.320E-01	6.77E-16	6.77E-16
	(5)	-1.032E+00	-1.03E+00	-1.03E+00

biến tính toán diện tích độc lập cho các thanh, các thanh giàn được phân thành 8 nhóm biến tính x_i (x_1, x_2, \dots, x_8), với diện tích của các thanh trong cùng nhóm là giống nhau, lần lượt là A_j (A_1, A_2, \dots, A_8). Hàm mục tiêu phục vụ tính toán tối ưu của bài toán là hàm của tổng khối lượng khung giàn thép là tương tự như bài toán giàn phẳng 10 thanh, như sau:

$$f(x) = \sum_{j=1}^{25} \rho A_j L_j \quad (10)$$

Toạ độ của các nút giàn và các trường hợp tải trọng được cho trong Bảng 6 và Bảng 7.

Bảng 5. Kết quả tối ưu tiết diện khung giàn thép [19]

x_i (in. ²)	HFPSO	FA	PSO
1	30.79	30.55	29.95
2	0.10	0.10	0.10
3	22.90	23.20	23.71
4	15.14	15.19	15.33
5	0.10	0.10	0.10
6	0.55	0.55	0.57
7	21.11	21.06	20.92
8	7.50	7.46	7.38
9	0.10	0.10	0.10
10	21.50	21.52	21.70
Khối lượng (lb ^a)	5061.1	5060.9	5061.8
Thời gian tính (s)	2.385	46.643	4.644

Bảng 6. Toạ độ nút giàn [19]

Nút	x	y	z
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

Khung giàn chịu hai trường hợp tải trọng (xem Bảng 7), các giới hạn ứng suất cho phép đối với từng nhóm thanh giàn nêu trong Bảng 8 và chuyển vị cho phép của các nút là 0,35in. theo các phương x, y và z [22]. Phạm vi lựa chọn tiết diện mặt cắt ngang của thanh là [0,01, 3,40] in.².

Kết quả tối ưu trong Bảng 9 cho thấy, thuật toán FA vẫn cung cấp kết quả tối ưu tốt nhất là 557,00lb mặc dù sự sai lệch giữa ba kết quả là không đáng kể, xem Bảng 9. Bên cạnh đó, cũng như bài toán tối ưu

giàn phẳng nếu xét trên cả phương diện chính xác và hiệu suất tính thì phiên bản lai HFPSO vẫn là thuật toán tốt nhất do thời gian tính là ít nhất và kết quả không có quá nhiều sai khác. Tuy nhiên, nếu so sánh kết quả của cả ba thuật toán trong nghiên cứu này với kết quả của một số nghiên cứu khác như kết quả của ACO là 545,53lb [23], HPSO là 545,19 [20] và TLBO là 545,175lb [22],..., có thể thấy rằng mức độ tối ưu của ba thuật toán vẫn cần được cải thiện hơn nữa (chênh lệch 2%). Nguyên nhân của kết quả này là do, ba phiên bản của các thuật toán được sử dụng trong nghiên cứu này là các phiên bản cơ sơ chưa có nhiều sự cải tiến như các phiên bản sau này.

Bảng 7. Các tổ hợp tải trọng [19]

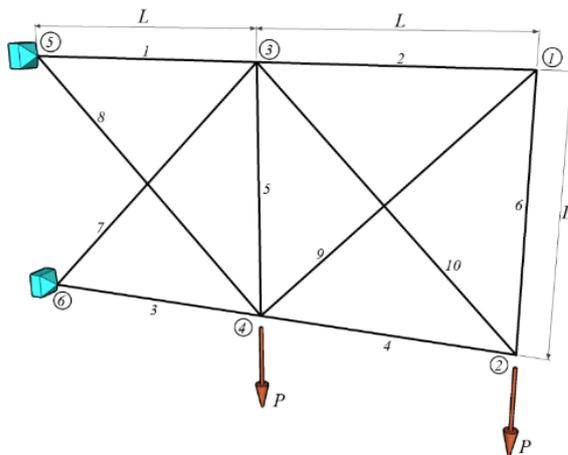
Tải trọng	Nút	P_x (kips)	P_y (kips)	P_z (kips)
1	1	1	-10	-10
	2	0	-10	-10
	3	0.5	0	0
2	6	0.6	0	0
	1	0	20	-5
	2	0	-20	-5

Bảng 8. Ứng suất cho phép trong các thanh của giàn [19]

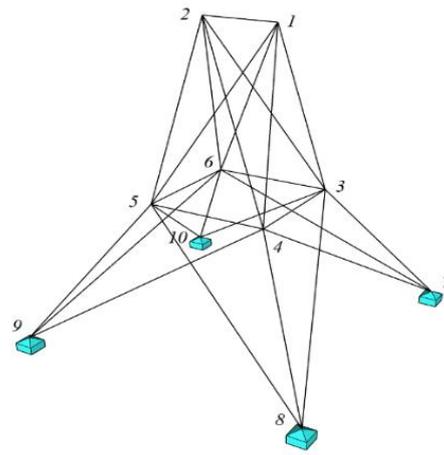
Nhóm	Phần tử	Ứng suất nén (ksi)	Ứng suất kéo (ksi)
x_1	1	35.092	35
x_2	2-5	11.590	35
x_3	6-9	17.305	35
x_4	10,11	35.092	35
x_5	12,13	35.092	35
x_6	14-17	6.7590	35
x_7	18-21	6.9590	35
x_8	22-25	11.082	35

Bảng 9. Kết quả tối ưu tiết diện khung giàn thép

x_i (in. ²)	HFPSO	FA	PSO
1	1.59	1.60	1.58
2	2.14	2.08	2.20
3	2.94	2.98	2.91
4	0.01	0.01	0.01
5	0.01	0.01	0.01
6	0.68	0.68	0.66
7	1.58	1.60	1.55
8	2.70	2.69	2.74
Khối lượng (lb)	557.06	557.00	557.26
Thời gian tính (s)	6.711	47.046	12.008



Hình 5. Giàn thép phẳng [19]



Hình 6. Giàn thép không gian [19]

8. Kết luận và đề xuất

Như vậy, nghiên cứu này đã trình bày một cách chi tiết các bài toán phân tích và đánh giá về hiệu suất, hiệu quả của ba kỹ thuật tính toán tối ưu lấy cảm hứng từ các hiện tượng tự nhiên, cụ thể là thuật toán FA, PSO và phiên bản lai giữa chúng HFPSO. Căn cứ theo kết quả đánh giá có thể thấy rằng thuật toán lai HFPSO đã thực sự tận dụng được ưu điểm của hai thuật toán gốc là PSO và FA. Tính chính xác của thuật toán HFPSO là đảm bảo tiệm cận với thuật toán FA nhưng tiết kiệm được thời gian tính toán hơn rất nhiều so với thuật toán FA. Nghiên cứu này là cơ sở trong việc lựa chọn và áp dụng các thuật toán lấy cảm hứng từ tự nhiên trong việc tối ưu hóa các bài toán giàn thép.

Lời cảm ơn

Nghiên cứu này được tài trợ bởi Trường Đại học Hàng hải Việt Nam trong đề tài mã số: **DT23-24.76**.

TÀI LIỆU THAM KHẢO

- [1] Fister, I. (2013), *A comprehensive review of bat algorithms and their hybridization* (Doctoral dissertation, Master's thesis, University of Maribor, Slovenia).
- [2] Kennedy, J. and R. Eberhart (1995). *Particle swarm optimization*. in Proceedings of ICNN'95-international conference on neural networks. IEEE.
- [3] Dorigo, M., V. Maniezzo, and A. Colomi (1996), *Ant system: optimization by a colony of cooperating agents*. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), Vol.26(1): pp.29-41.
- [4] 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.
- [5] Holland, J.H.(1992), *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.
- [6] Rechenberg, I. (1978), *Evolutionsstrategien*, in Simulationsmethoden in der Medizin und Biologie., Springer. pp. 83-114.
- [7] Koza, J.R (1992). *Evolution of subsumption using genetic programming*. in Proceedings of the first European conference on artificial life. MIT Press Cambridge, MA.
- [8] Rashedi, E., H. Nezamabadi-Pour, and S. Saryazdi (2009), *GSA: a gravitational search algorithm*. Information sciences, Vol.179(13): pp.2232-2248.
- [9] Hatamlou, A. (2013), *Black hole: A new heuristic optimization approach for data clustering*. Information sciences, Vol.222: pp.175-184.
- [10] Mirjalili, S. and A. Lewis (2016), *The whale optimization algorithm*. Advances in engineering software, Vol.95: pp.51-67.
- [11] Raidl, G.R. and J.J.H.m. Puchinger (2008), *Combining (integer) linear programming techniques and metaheuristics for combinatorial optimization.*: pp.31-62.
- [12] Yang, X.-S.J.a.p.a. (2010), *Firefly algorithm, stochastic test functions and design optimisation*.

- [13] Aydilek, I.B.J.A.S.C. (2018), *A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems*. Vol.66: pp.232-249.
- [14] Shi, Y. and R. Eberhart (1998). *A modified particle swarm optimizer*. in 1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360). 1998. IEEE.
- [15] Shi, Y., et al. (2011), *Cellular particle swarm optimization*. Vol.181(20): pp.4460-4493.
- [16] Yang, X.-S (2009). *Firefly algorithms for multimodal optimization*. in *International symposium on stochastic algorithms*. Springer.
- [17] Fister, I., et al. (2013), *A comprehensive review of firefly algorithms*. Vol.13: pp.34-46.
- [18] Mirjalili, S., S.M. Mirjalili, and X.-S. Yang (2014), *Binary bat algorithm*. *Neural Computing and Applications*, Vol.25(3): pp.663-681.
- [19] Vu-Huu, T., et al. (2023), *An improved bat algorithms for optimization design of truss structures*. *Structures*, Vol.47: pp.2240-2258.
- [20] Li, L.-J., et al. (2007), *A heuristic particle swarm optimizer for optimization of pin connected structures*. Vol.85(7-8): pp.340-349.
- [21] Sonmez, M. (2011), *Artificial Bee Colony algorithm for optimization of truss structures*. *Applied Soft Computing*, Vol.11(2): pp.2406-2418.
- [22] Camp, C.V. and M. Farshchin (2014), *Design of space trusses using modified teaching-learning based optimization*. *Engineering Structures*. Vol.62: pp.87-97.
- [23] Camp, C.V. and B.J.J.J.o.s.e. Bichon (2004), *Design of space trusses using ant colony optimization*. Vol.130(5): pp.741-751.

Ngày nhận bài:	30/10/2023
Ngày nhận bản sửa:	10/11/2023
Ngày duyệt đăng:	16/11/2023