



## MODEL UPDATING FOR A RAILWAY BRIDGE USING A HYBRID EVOLUTIONARY OPTIMIZATION ALGORITHM

Nguyen Manh Hai\*, Tran Ngoc Hoa, Bui Tien Thanh, Tran Viet Hung,  
Nguyen Duy Tien

University of Transport and Communications, No 3 Cau Giay Street, Hanoi, Vietnam

### ARTICLE INFO

TYPE: Research Article

Received: 31/07/2021

Revised: 07/09/2021

Accepted: 20/09/2021

Published online: 15/10/2021

<https://doi.org/10.47869/tcsj.72.8.2>

\* *Corresponding author*

Email: nmhai@utc.edu.vn

**Abstract.** Cuckoo Search (CS) is a common evolutionary algorithm that has been employed to deal with numerous optimization problems over the last decades. However, one of the fundamental disadvantages of the CS is that this algorithm fixes the velocities of the populations throughout the global search. This drawback restricts the flexibility in the movement of the populations. In addition, CS not only depends mainly on the quality of the original population but also does not have capabilities such as crossover or mutation operators to improve the quality of the new generations. To deal with this shortcoming of traditional CS, in this research, an improved CS (ICS) combined with a genetic algorithm (GA) is proposed. To demonstrate the effectiveness of the proposed algorithm (a hybrid evolutionary algorithm combined genetic algorithm with improved cuckoo search - HGAICS), a large-scale truss bridge (Do Len Bridge) is employed. HGAICS is used to minimize the deviation between the numerical and measured results, determine structural uncertain parameters. The obtained results show that HGAICS outperforms GA and CS in terms of accuracy and extremely reduces computational time compared with GA.

**Keywords:** Model updating, Truss bridge, CS, GA, Hybrid evolutionary algorithm.

© 2021 University of Transport and Communications



## CẬP NHẬT MÔ HÌNH CHO CẦU GIÀN THÉP ĐƯỜNG SẮT SỬ DỤNG THUẬT TOÁN TỐI ƯU TIẾN HOÁ LAI

Nguyễn Mạnh Hải\*, Trần Ngọc Hoà, Bùi Tiến Thành, Trần Việt Hưng,  
Nguyễn Duy Tiến

Trường Đại học Giao thông vận tải, Số 3 Cầu Giấy, Hà Nội, Việt Nam

### THÔNG TIN BÀI BÁO

CHUYÊN MỤC: Công trình khoa học

Ngày nhận bài: 31/07/2021

Ngày nhận bài sửa: 07/09/2021

Ngày chấp nhận đăng: 20/09/2021

Ngày xuất bản Online: 15/10/2021

<https://doi.org/10.47869/tcsj.72.8.2>

\* Tác giả liên hệ

Email: nmhai@utc.edu.vn

**Tóm tắt.** Thuật toán tối ưu hoá chim cú cu (Cuckoo Search – CS) là một thuật toán tiến hóa phổ biến đã được sử dụng để giải quyết nhiều vấn đề tối ưu hóa trong nhiều thập kỷ qua. Tuy nhiên, một trong những nhược điểm cơ bản của CS là thuật toán này cố định vận tốc của các quần thể trong toàn bộ quá trình tìm kiếm giải pháp tối ưu, vì vậy giảm sự linh hoạt trong quá trình di chuyển của các phần tử. Ngoài ra CS không chỉ phụ thuộc chủ yếu vào chất lượng của quần thể ban đầu mà còn không có các khả năng như giao chéo hay đột biến để cải thiện chất lượng của các thế hệ mới. Để khắc phục nhược điểm này của CS, trong nghiên cứu này, một CS cải tiến (ICS) kết hợp với thuật toán di truyền (GA) được đề xuất. Để chứng minh được hiệu quả của phương pháp được đề xuất (Thuật toán lai giữa GA và ICS - HGAICS), một kết cấu cầu giàn thép nhịp lớn đã được sử dụng. HGAICS được sử dụng để giảm sự khác biệt giữa kết quả mô hình số và mô hình thực nghiệm, xác định các tham số bất định của kết cấu. Kết quả thu được cho thấy HGAICS vượt trội hơn GA và CS về độ chính xác và giảm thời gian tính toán so với GA.

**Từ khóa:** Cập nhật mô hình, Cầu giàn thép, CS, GA, thuật toán tối ưu tiến hoá lai.

### 1. MỞ ĐẦU

Sự xuất hiện của các thuật toán tối ưu hóa tiến hóa (Evolutionary algorithm - EA) trong những thập kỷ gần đây đã nâng cao hiệu quả của giám sát sức khỏe công trình (Structural health monitoring – SHM) [1-3]. Các thuật toán tối ưu tiến hóa giúp giảm thiểu sự khác biệt giữa các kết quả được tính toán và đo lường. Sau đó, mô hình đã hiệu chỉnh có thể dự đoán ứng xử của các kết cấu được xem xét một cách chính xác hơn. Trong những thập kỷ gần đây, nhiều nhà nghiên cứu đã áp dụng thành công các thuật toán tối ưu tiến hóa cho SHM [4-9]. Ví

du, Na và cộng sự. [10] đã sử dụng thuật toán GA để xác định hư hỏng cho một kết cấu nhà quy mô lớn. Phương pháp đề xuất có thể xác định chính xác vị trí và mức độ hư hỏng của kết cấu được xem xét, ngay cả khi dữ liệu đo bị ảnh hưởng bởi nhiễu. Jung và cộng sự. [11] đã áp dụng GA để cập nhật mô hình cho một công trình cầu quy mô nhỏ. Hàm mục tiêu bao gồm biến dạng, và các đặc trưng động học (tần số dao động tự nhiên và hình thái dao động). Tuy nhiên so với các thuật toán tối ưu khác như thuật toán tối ưu bầy đàn (Particle Swarm Optimization – PSO), hay đặc biệt là CS, GA thường có độ hội tụ thấp và kém hiệu quả hơn khi giải quyết các bài toán tối ưu [12].

Nhiều nhà khoa học khác cũng thành công khi sử dụng các thuật toán tối ưu để giám sát sức khỏe kết cấu. Hòa và cộng sự [13] cũng đã áp dụng cả GA và PSO để cập nhật mô hình cho cầu đường sắt Nam Ô, từ đó xác định các tham số bất định. Nghiên cứu này sử dụng GA và PSO riêng rẽ để cập nhật mô hình và kết quả chứng minh rằng PSO cung cấp độ chính xác cao hơn GA. Tuy nhiên nghiên cứu chưa xét đến ảnh hưởng của các loại tải trọng, và vận tốc di chuyển khác nhau của đoàn tàu đến ứng xử động của kết cấu. Ngoài ra việc sử dụng GA hay PSO riêng rẽ cũng tồn tại những nhược điểm, làm giảm độ chính xác của kết quả thu được. Ngoài ra, độ chính xác của thuật toán PSO cũng thấp hơn nhiều so với thuật toán CS [12].

Trong các thuật toán tối ưu tiến hóa, thuật toán CS là thuật toán được dùng phổ biến trong những năm gần đây với nhiều ưu điểm vượt trội. Điển hình như, khả năng tìm kiếm trong không gian rộng lớn, có thể chọn lọc để giữ lại các quần thể có chất lượng tốt nhất cho thế hệ sau. Vì vậy thuật toán này đã được áp dụng thành công để giải quyết các vấn đề tối ưu hóa, đặc biệt là vấn đề giám sát sức khỏe kết cấu, ví dụ, Xu và cộng sự. [14] đã sử dụng CS để xác định vị trí và mức độ hư hỏng của kết cấu dầm và kết cấu giàn. Kết quả chứng minh rằng CS có thể phát hiện các hư hỏng trong các kết cấu được xem xét một cách chính xác ngay cả khi ảnh hưởng của nhiễu đã được xem xét. Ngoài ra nhiều nhà khoa học khác cũng áp dụng thành công thuật toán CS để giám sát sức khỏe kết cấu [15].

Tuy nhiên, CS cũng tồn tại một số nhược điểm cơ bản làm giảm hiệu quả và độ chính xác của nó trong các vấn đề tối ưu hóa. Một trong những nhược điểm chính của thuật toán CS truyền thống là thuật toán này cố định vận tốc của các quần thể trong toàn bộ quá trình tìm kiếm, vì vậy giảm sự linh hoạt trong quá trình di chuyển của các phần tử. Ngoài ra CS không chỉ phụ thuộc chủ yếu vào chất lượng của quần thể ban đầu mà còn không có các khả năng như giao chéo hay đột biến để cải thiện chất lượng của các thế hệ mới. Nếu vị trí của các quần thể ban đầu khác xa so với vị trí tốt nhất toàn cục, độ chính xác của kết quả thu được sẽ không cao (không tìm được các giải pháp tối ưu toàn cục).

Trong nghiên cứu này, chúng tôi đề xuất một thuật toán để khắc phục những nhược điểm của phương pháp GA và CS truyền thống. Đầu tiên thuật toán cải tiến của CS (ICS) được đề xuất để tăng sự linh hoạt trong quá trình tìm kiếm của CS, sau đó ICS sẽ kết hợp với GA để cập nhật mô hình cho một cầu giàn thép thực tế. Thuật toán được đề xuất có thể kết hợp ưu điểm của 2 phương pháp GA (có khả năng lai chéo và đột biến để nâng cao chất lượng của các quần thể) và phương pháp ICS (khả năng tìm kiếm toàn cục ngẫu nhiên). Vì vậy, thuật toán đề xuất có thể khắc phục được nhược điểm của từng thuật toán GA, và CS riêng rẽ, cải thiện độ chính xác của kết quả thu được.

## 2. THUẬT TOÁN TỐI ƯU HGAICS

CS là thuật toán lấy cảm hứng từ quá trình sinh sản của chim cú cu được Xin-she Yang đề xuất năm 2009 [16]. Trong đó chim cú cu đẻ trứng vào tổ và để các con chim cú cu non

sống dựa vào nguồn thức ăn của các loài chim khác, vì vậy tăng khả năng sinh tồn của chúng trong quá trình tiến hoá. Thuật toán CS được hình thành dựa trên ba quy tắc lý tưởng hóa sau đây.

- Chim cú cu chọn một tổ ngẫu nhiên để đẻ trứng và mỗi lần chỉ đẻ một quả trứng trong một tổ.
- Những tổ có chất lượng trứng cao nhất sẽ được dùng cho các thế hệ tiếp theo.
- Số lượng tổ ngẫu nhiên mà chim cú cu chọn để đẻ trứng là cố định (đại diện cho tổng số lượng quần thể hay số giải pháp trong bài toán tối ưu). Xác suất chim chủ phát hiện ra chim cú cu đẻ trứng vào tổ của mình và vút trứng của chim cú cu là  $P_a$  (0-100%).

Các giải pháp tối ưu của CS qua từng bước lặp được xác định dựa vào công thức (1):

$$X^{l+1} = X^l + \text{levy}(\lambda) * \alpha \quad (1)$$

Ở đây

$$X^l = [x_1^l, x_2^l, \dots, x_m^l], \quad (2)$$

Và

$$x_j^l = [x_{1j}^l, x_{2j}^l, \dots, x_{nj}^l]^T, j = 1, 2, \dots, m. \quad (3)$$

Trong đó  $x_j^l$  và  $x_j^{l+1}$  là giải pháp của cú cu  $j$  tại bước lặp thứ  $l$  và  $l+1$ ; trong khi  $n$  cho biết số lượng quần thể và  $m$  là số lượng các tham số bất định cần được xác định.  $\lambda$  thể hiện độ dài bước của phân phối levy.  $\alpha$  là hệ số ngẫu nhiên của độ dài bước, đối với phương pháp CS truyền thống  $\alpha$  được chọn cố định ( $\alpha = 1$ ).  $\lambda$  được xác định từ phân phối levy sử dụng công thức (4)

$$\text{Levy} \sim u = t^{-\lambda}; (1 \leq \lambda \leq 3) \quad (4)$$

Thuật toán CS phát triển các bước lặp để xác định mức độ phù hợp của từng quần thể dựa trên hàm mục tiêu. Khi hàm mục tiêu là tối thiểu, vị trí tốt nhất toàn cục sẽ đạt được. Tuy nhiên, một nhược điểm chính của thuật toán CS truyền thống là việc cố định vận tốc của các quần thể trong toàn bộ quá trình tìm kiếm. Việc cố định giá trị hệ số ngẫu nhiên của độ dài bước  $\alpha$  bé sẽ dẫn đến sự hội tụ chậm và gia tăng lớn về chi phí tính toán. Ngược lại, nếu cố định giá trị của  $\alpha$  lớn, giải pháp tối ưu có thể bị bỏ qua. Vì vậy  $\alpha$  nên được chọn linh hoạt, cụ thể khi các phần tử di chuyển đến gần vị trí tối ưu (sai số hàm mục tiêu bé), chúng sẽ sử dụng các bước di chuyển bé (giá trị  $\alpha$  nên chọn bé). Nếu vị trí của các phần tử ở xa phương án tối ưu (sai số hàm mục tiêu bé), chúng sẽ di chuyển với bước lớn hơn (giá trị  $\alpha$  nên chọn lớn). Ngoài ra, độ chính xác của kết quả thu được của CS thường thấp nếu vị trí xuất phát ban đầu của các quần thể nằm xa vị trí tối ưu toàn cục và thuật toán này cũng không có khả năng lai chéo hay đột biến để cải thiện chất lượng của các thế hệ mới.

Vì vậy, trong bài báo này, chúng tôi đề xuất một thuật toán để khắc phục những nhược điểm của phương pháp GA và CS truyền thống. Đầu tiên ICS được đề xuất để tăng sự linh hoạt trong quá trình tìm kiếm của CS, sau đó ICS sẽ kết hợp với GA để khắc phục những nhược điểm và cải thiện độ chính xác của mỗi phương pháp truyền thống. Hai nguyên tắc nổi bật của phương pháp được đề xuất sẽ được mô tả ngắn gọn như sau: (1) do sở hữu các toán tử giao chéo và đột biến, đầu tiên GA được áp dụng để tạo ra các quần thể ưu tú ban đầu và (2)

các quần thể đó sau đó được sử dụng để tìm kiếm giải pháp tốt nhất dựa trên khả năng tìm kiếm toàn cục và linh hoạt của ICS.

Để nâng cao độ chính xác và tốc độ hội tụ của thuật toán, ICS áp dụng tham số  $\alpha$  một cách linh hoạt. Cụ thể, nếu các phần tử di chuyển đến gần vị trí tối ưu (sai số hàm mục tiêu bé), chúng sẽ sử dụng các bước di chuyển bé để tránh bỏ lỡ giải pháp tốt nhất. Nếu vị trí của các phần tử ở xa phương án tối ưu (sai số hàm mục tiêu lớn), chúng sẽ di chuyển với bước lớn hơn để tiến nhanh đến giải pháp tối ưu và giảm thời gian tính toán. Giá trị của  $\alpha$  được xác định theo công thức (5) và (6)

$$\alpha_l = (\alpha_{max})^{l * c} \quad (5)$$

$$c = \frac{1}{N} * Ln\left(\frac{\alpha_{min}}{\alpha_{max}}\right) \quad (6)$$

Trong đó  $l$  đại diện cho số lần lặp hiện tại,  $l \in [0, N]$ ,  $N$  là tổng số bước lặp,  $\alpha_{max}$  và  $\alpha_{min}$  giá trị lớn nhất và nhỏ nhất của độ dài bước.

Nguyên lý làm việc của HGAICS được trình bày cụ thể như các bước dưới đây:

**Bước 1:** Tạo vị trí ban đầu của các quần thể

$$X^0 = [x_1^0, x_2^0, \dots, x_m^0], \quad (7)$$

$$x_j^0 = [x_{1j}^0, x_{2j}^0, \dots, x_{nj}^0]^T, \quad j = 1, 2, \dots, m. \quad (8)$$

**Bước 2:** Lựa chọn hàm mục tiêu  $\phi(x)$

$$\phi(x) = \sum_{z=1}^t \frac{(\tilde{f}_z - f_z)^2}{(\tilde{f}_z)^2} + \sum_{z=1}^t 1 - \frac{(\check{\epsilon}_z^T * \epsilon_z)^2}{(\epsilon_z^T * \epsilon_z) * (\check{\epsilon}_z^T * \check{\epsilon}_z)} \quad (9)$$

$(f_z, \epsilon_z)$  và  $(\tilde{f}_z, \check{\epsilon}_z)$  lần lượt là tần số dao động tự nhiên và hình thái dao động của mô hình số và mô hình thí nghiệm. “z” là thứ tự mode và  $t$  là số lượng mode.

**Bước 3:** Xác định giá trị của hàm mục tiêu  $\phi(x)$  của các quần thể ban đầu

$$\phi(x^0) = \begin{bmatrix} \phi(x_1^0) \\ \phi(x_2^0) \\ \phi(x_3^0) \\ \vdots \\ \phi(x_n^0) \end{bmatrix}_{n \times 1} \quad (10)$$

**Bước 4.** Sắp xếp giá trị tốt nhất của mỗi quần thể theo thứ tự tăng dần dựa trên giá trị của hàm mục tiêu  $\phi(x)$ .

$$\mathbf{p}_{best}^0 = \begin{bmatrix} p_p^0 \\ p_u^0 \\ \cdot \\ \vdots \\ p_v^0 \end{bmatrix}_{n*1} \quad (11)$$

$$\mathbf{p}_p^0 = [x_{p1}^0 \ x_{p2}^0 \ \dots \ x_{pm}^0]_{1*m} \quad (12)$$

$$\mathbf{p}_u^0 = [x_{u1}^0 \ x_{u2}^0 \ \dots \ x_{um}^0]_{1*m} \quad (13)$$

$$\mathbf{p}_v^0 = [x_{v1}^0 \ x_{v2}^0 \ \dots \ x_{vm}^0]_{1*m} \quad (14)$$

$$\phi(x_p^0) < \phi(x_u^0) < \phi(x_v^0) \quad (15)$$

$p, u, \dots, v$  lần lượt là quần thể thứ  $p^{th}, u^{th}$ , and  $v^{th}$ .

**Bước 5.** Quá trình lai chéo và đột biến

Các cặp bố mẹ với chất lượng tốt nhất xác định từ bước 4 được chọn để lai chéo và đột biến, số lượng cặp bố mẹ được lựa chọn dựa trên tỷ lệ phần trăm lai chéo và đột biến được tính như trong công thức (16) và (17)

$$n_c = p_c * n \quad (16)$$

$$n_m = p_m * n \quad (17)$$

$p_c$  là phần trăm lai chéo và  $p_m$  là phần trăm đột biến

**Bước 6.** Tính toán lại giá trị của hàm mục tiêu  $\phi(x)$

**Bước 7.** Chọn các quần thể tốt nhất sau quá trình lai chéo và đột biến dựa trên các giá trị của hàm mục tiêu  $\phi(x)$ .

$$\mathbf{p}_{best}^i = \begin{bmatrix} p_p^i \\ p_u^i \\ \cdot \\ \vdots \\ p_v^i \end{bmatrix}_{n*1} \quad (18)$$

$$\mathbf{p}_p^i = [x_{p1}^i \ x_{p2}^i \ \dots \ x_{pm}^i]_{1*m} \quad (19)$$

$$\mathbf{p}_u^i = [x_{u1}^i \ x_{u2}^i \ \dots \ x_{um}^i]_{1*m} \quad (20)$$

$$\mathbf{p}_v^i = [x_{v1}^i \ x_{v2}^i \ \dots \ x_{vm}^i]_{1*m} \quad (21)$$

$$\phi(x_p^i) < \phi(x_u^i) < \phi(x_v^i) \quad (22)$$

**Bước 8.** Lặp lại bước 5 đến bước 7 cho đến khi hết số bước lặp của GA.

**Bước 9.** Quần thể ban đầu của ICS được xác định dựa trên kết quả có được từ GA và chiến thuật của Levy flights [16].

$$X^{t+1} = p_{best}^t + levy(\lambda) * \alpha_l \quad (23)$$

Ở đây

$$p_{best}^t = [x_1^t, x_2^t, \dots, x_m^t], \quad (24)$$

Và

$$x_j^t = [x_{1j}^t, x_{2j}^t, \dots, x_{nj}^t]^T, j = 1, 2, \dots, m. \quad (25)$$

$t$  là số bước lặp sử dụng của thuật toán GA và  $p_{best}^t$  là giá trị tốt nhất của các quần thể tại bước lặp thứ  $t$  của GA.

**Bước 10:** Độ dài bước của phân phối levy được xác định như công thức (26)

$$levy(\lambda) = \frac{1}{\delta^\beta} \quad (26)$$

Tham số ( $\delta$ ) được xác định như công thức (27)

$$\delta = \left( \frac{\Gamma(1 + \beta) * \sin\left(\frac{\pi * \beta}{2}\right)}{\Gamma\left(\frac{1 + \beta}{2}\right) * \beta * 2^{\frac{(\beta-1)}{2}}}\right)^{\frac{1}{\beta}} \quad (27)$$

Hàm số  $\Gamma$  được xác định như công thức (28) và (29)

- Đối với bất kỳ số nguyên dương nào ( $k$ ),

$$\Gamma(k) = (k - 1)! \quad (28)$$

- Đối với số phức

$$\Gamma(y) = \int_0^\infty w^{y-1} e^{-w} dw \quad (29)$$

**Bước 10:** Cập nhật vị trí của các quần thể tại bước lặp  $t+1$

$$\text{Nếu } X^{t+1} > x_{cận\ trên} \quad (30)$$

$$X^{t+1} = x_{cận\ dưới} + rand * x_{cận\ dưới} \quad (31)$$

$$\text{Nếu không } X^{t+1} < x_{cận\ dưới} \quad (32)$$

$$X^{t+1} = x_{cận\ dưới} + rand * x_{cận\ dưới} \quad (33)$$

$x_{min}$  và  $x_{max}$  là cận trên và cận dưới của không gian tìm kiếm.

**Bước 11:** Chọn giá trị tốt nhất cục bộ của từng quần thể và giá trị tốt nhất toàn cục cho lần lặp tiếp theo dựa trên hàm mục tiêu  $\phi(X)$

$$\text{Nếu } \phi(X^{t+1}) < \phi(X^t) \quad (34)$$

$$\phi(X^{t+1}) = \phi(X^{t+1}); X^{t+1} = X^{t+1} \quad (35)$$

Nếu không

$$\phi(X^{t+1}) = \phi(X^t); X^{t+1} = X^t \quad (36)$$

**Bước 12:** Các quần thể mới sẽ được xác định dựa trên sự lựa chọn từ random walks

$$K = rand(X^{t+1}) > P_a \quad (37)$$

Trong đó  $P_a$  là xác suất mà chim chủ có thể phát hiện trứng của chim cú cu

$$X^{t+2} = X^{t+1} + \alpha_n * K \quad (38)$$

$$\text{Nếu } X^{t+2} > x_{\text{cận trên}} \quad (39)$$

$$X^{t+2} = x_{\text{cận dưới}} + rand * x_{\text{cận dưới}} \quad (40)$$

$$\text{Nếu không } X^{t+1} < x_{\text{cận dưới}} \quad (41)$$

$$X^{t+2} = x_{\text{cận dưới}} + rand * x_{\text{cận dưới}} \quad (42)$$

**Bước 12:** Xác định giá trị tốt nhất cục bộ của từng quần thể và giá trị tốt nhất toàn cục

$$\text{Nếu } \phi(X^{t+2}) < \phi(X^{t+1}) \quad (43)$$

$$\phi(X^{t+2}) = \phi(X^{t+2}); X^{t+2} = X^{t+2} \quad (44)$$

Nếu không

$$\phi(X^{t+2}) = \phi(X^{t+1}); X^{t+2} = X^{t+1} \quad (45)$$

**Bước 13:** Lặp lại bước 9-12 cho đến khi hàm mục tiêu đạt được (sai số giữa kết quả tính toán và kết quả thực tế  $< 10^{-3}$  hoặc tổng số bước lặp kết thúc (100 bước).

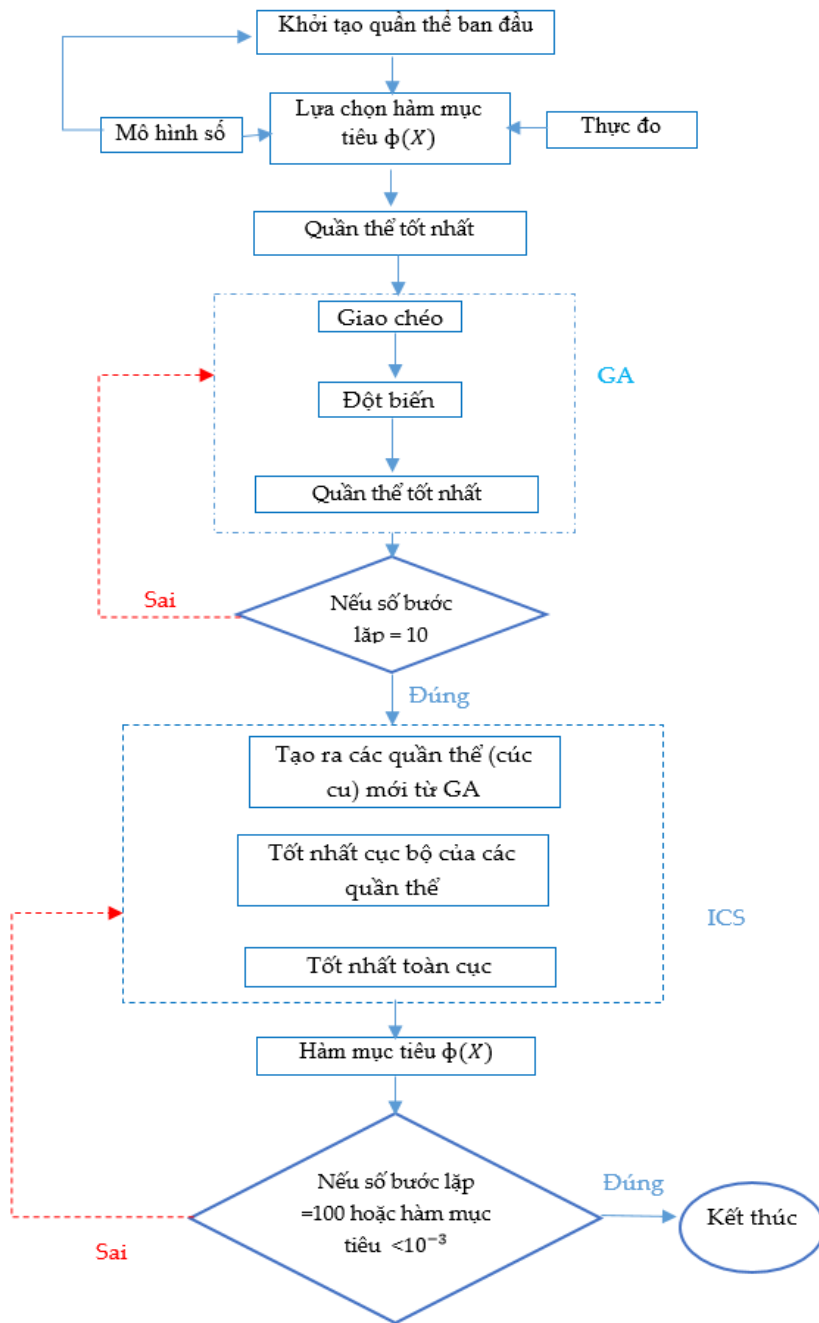
**Bước 14:** Giải pháp tốt nhất được xác định.

$$\phi(G_{best}) = \min(\phi(x)) \quad (46)$$

$$X = G_{best} \quad (47)$$

Sơ đồ khối quá trình kết hợp GA với ICS để cập nhật mô hình được thể hiện như Hình 1.





Hình 1. Sơ đồ khối sử dụng thuật toán HGAICS để cập nhật mô hình.

### 3. CẬP NHẬT MÔ HÌNH CHO CẦU GIÀN THÉP ĐÒ LÈN SỬ DỤNG HGAICS

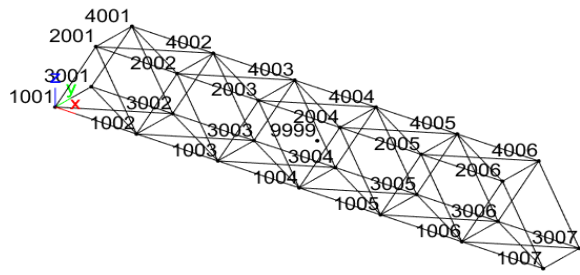
#### 3.1. Mô hình phần tử hữu hạn cầu giàn thép

Để đánh giá hiệu quả của thuật toán đề xuất, trong phần này cầu giàn thép đường sắt (cầu Đò Lèn) được lựa chọn để cập nhật mô hình. Cầu Đò Lèn (Hình 2) nằm tại Km778+155, tuyến đường sắt Hà Nội – TP. Hồ Chí Minh. Cầu gồm ba nhịp giàn thép bố trí theo sơ đồ

49,8m+54,6m+49,8m. Hai nhịp biên gồm 06 khoang dài 8,3m, nhịp giữa gồm 07 khoang dài 7,8m, khoảng cách giữa hai thanh giàn dưới là 4,8m, chiều cao giàn 8,0m.



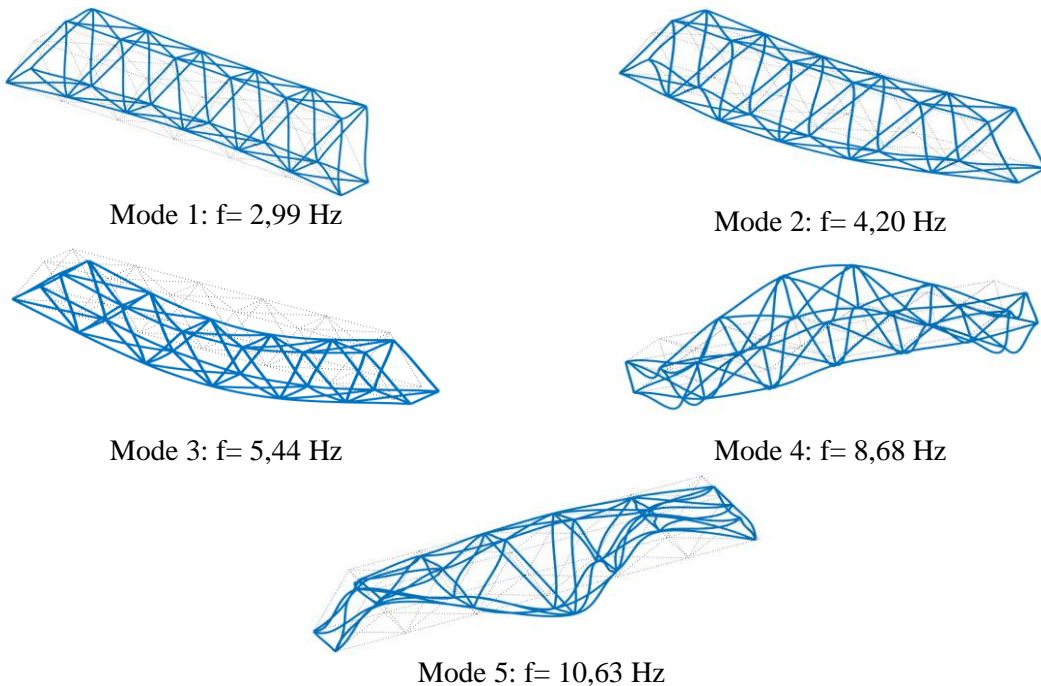
Hình 2. Cầu Đò Lèn Km778+155, tuyến đường sắt Hà Nội – TP. Hồ Chí Minh



Hình 3. Mô hình cầu Đò Lèn

Để phân tích ứng xử động của kết cấu, mô hình phần tử hữu hạn của cầu Đò Lèn (Hình 3) được xây dựng sử dụng chương trình Stabil được lập trình trên nền tảng MATLAB.

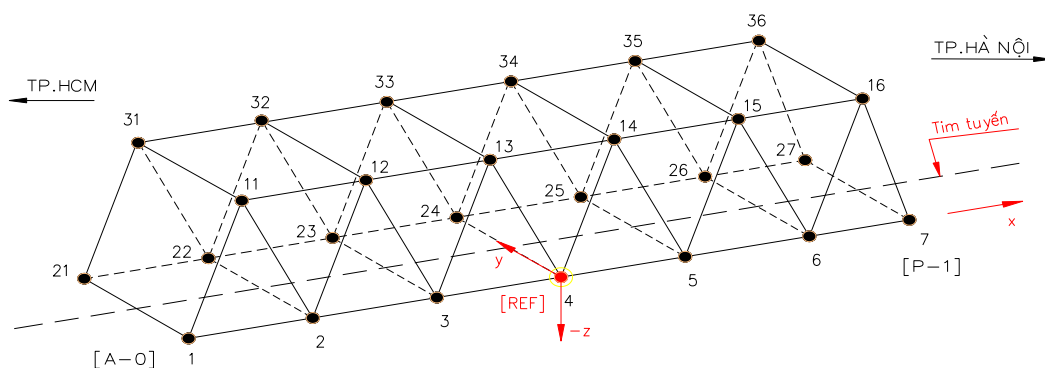
Mô hình phần tử hữu hạn của cầu được mô tả như sau: Cầu được mô hình hóa bởi 81 nút, 135 phần tử và 15 loại mặt cắt của các thanh giàn được sử dụng (các kích thước này được tham khảo từ các bản vẽ thiết kế cầu). Các thành phần kết cấu được mô hình hóa bằng cách sử dụng các phần tử dầm không gian. Phần tử này có sáu bậc tự do tại mỗi nút bao gồm các chuyển vị tịnh tiến theo các hướng  $x$ ,  $y$  và  $z$  và các chuyển vị xoay xung quanh các hướng  $x$ ,  $y$  và  $z$ . Hình dạng dao động của các mode của mô hình số được thể hiện như trong Hình 4.



Hình 4. Các mode từ mô hình số.

### 3.2. Thí nghiệm hiện trường để xác định các đặc trưng động học cầu Đò Lèn

Sau khi quá trình đánh giá và xác định các dạng dao động từ mô hình số, cơ bản có thể xây dựng được một mạng lưới điểm đo dao động, dựa vào đó có thể xác định dạng mode và đặc điểm dao động để nghiên cứu. Bố trí các điểm đo tối ưu, cũng như việc lắp đặt được đầu đo lên điểm đo cần được xét đến. Về cơ bản nên chọn các điểm đo dễ dàng lắp đặt, nhưng vẫn phải cung cấp đủ tham số để phân tích, đánh giá một cách chính xác nhất sự làm việc của kết cấu. Sơ đồ bố trí điểm đo được thể hiện như hình 5.



Hình 5. Sơ đồ không gian bố trí các điểm đo dao động tại các nút giàn.

Sau khi đã lắp đặt toàn bộ các đầu đo cần thiết, liên kết các đầu đo với máy tính để lưu trữ số liệu (Hình số 6a), tiến hành đo dao động của KCN cầu dưới tác dụng của các đoàn tàu chạy qua (Hình số 6b).



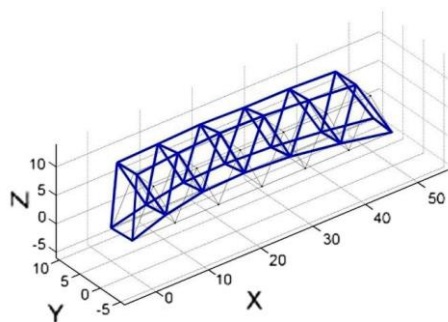
(a)



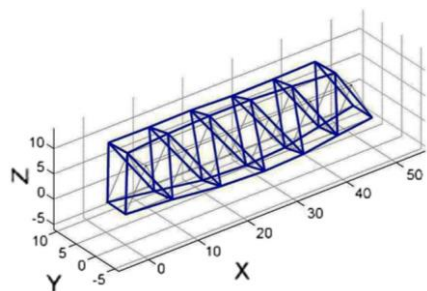
(b)

Hình 6. Công tác thí nghiệm hiện trường: (a) Lắp đặt các đầu đo dao động trên các nút giàn thép; (b) Liên kết hệ thống thu dữ liệu với máy tính và đầu đo.

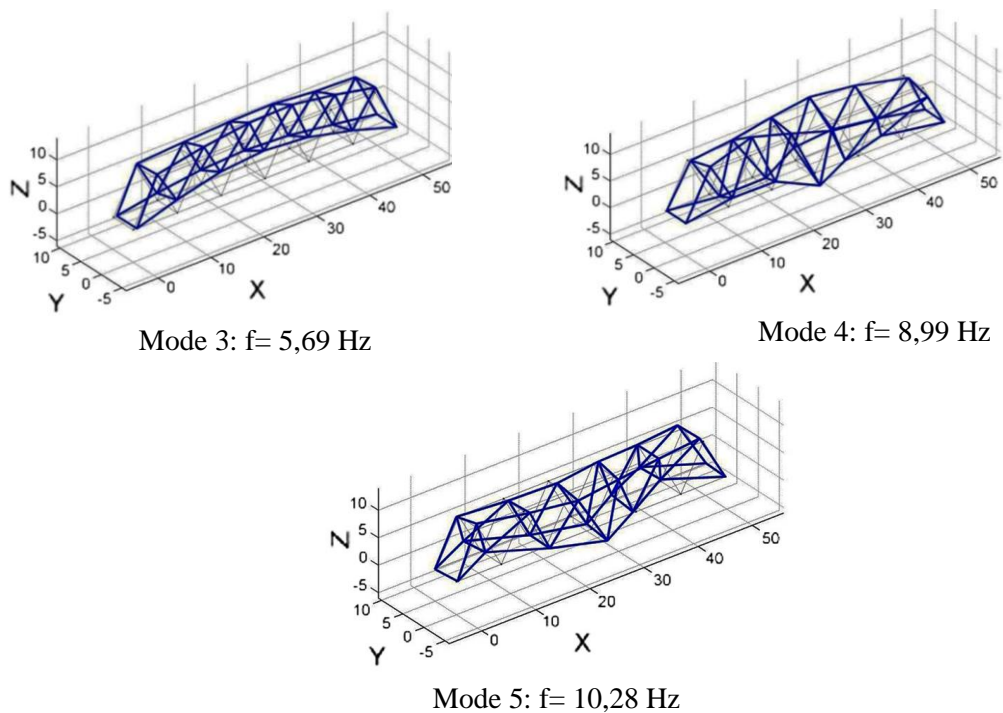
Kết quả dao động của 5 mode đầu tiên được thể hiện như trong hình số 7 và Bảng số 1.



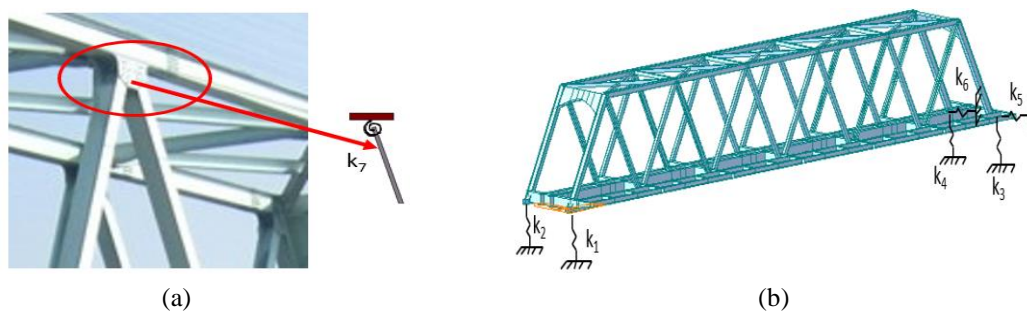
Mode 1:  $f = 3,06 \text{ Hz}$



Mode 2:  $f = 4,56 \text{ Hz}$



Hình 7. Tần số dao động tự nhiên và hình dạng các mode thu được từ thực nghiệm.



Hình 8. Các thông số kết cấu bất định được chọn để cập nhật trong mô hình: (a) Lò xo tại nút giàn; (b) Lò xo ở gối cầu.

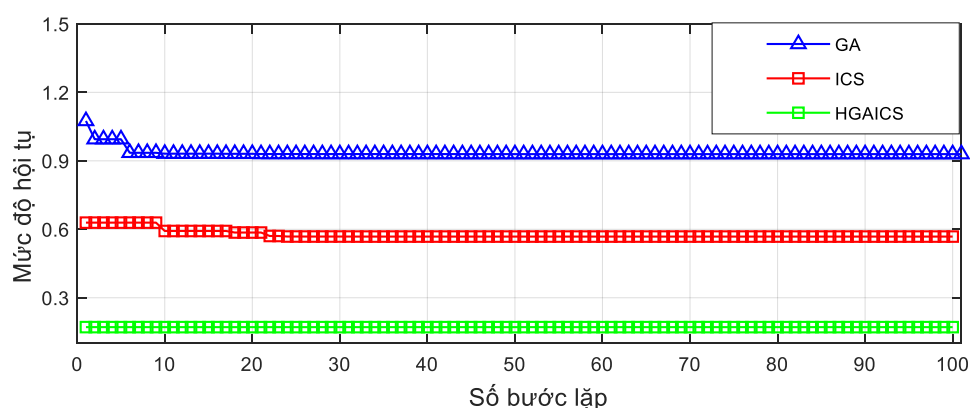
Bảng 1. Tổng hợp tần số dao động tự nhiên cầu Đò Lèn theo kết quả tính và thực nghiệm.

Thứ tự	Tần số (Hz)		Chênh lệch (%)	Dạng dao động
	Từ kết quả mô hình số	Từ kết quả thực nghiệm		
1	2,99	3,06	2,15	Dao động uốn thẳng đứng
2	4,20	4,56	7,82	Dao động lắc ngang
3	5,44	5,69	4,31	Dao động uốn thẳng đứng
4	8,68	8,99	3,41	Dao động uốn ngang
5	10,63	10,28	3,45	Dao động xoắn

### 3.3. Cập nhật mô hình cho cần giàn thép Đò Lèn sử dụng HGAICS

Tám thông số bất định ảnh hưởng lớn nhất đến đặc trưng động học của kết cấu, bao gồm mô đun đàn hồi của các thanh giàn ( $E$ ), độ cứng của 6 lò xo dưới gối cầu ( $k_1, k_2, k_3, k_4, k_5, k_6$ ) và độ cứng của lò xo tại các khớp của giàn ( $k_7$ ) (thể hiện như trong Hình 8) được chọn để cập nhật mô hình. Độ cứng ban đầu của lò xo được ước tính dựa trên các công thức tham khảo [13], trong khi độ cứng của lò xo tại gối cầu được tính toán dựa theo loại gối cầu.

Thuật toán đề xuất HGAICS, được sử dụng để tìm kiếm giải pháp tối ưu toàn cục. Để so sánh với HGAICS, thuật toán GA và ICS riêng rẽ cũng được sử dụng. Đối với HGAICS, 100 quần thể được sử dụng. Đối với ICS, xác suất để chim chủ phát hiện ra trứng chim cúc cu là  $P_a$  là 0,25; các hệ số này được chọn dựa theo các nghiên cứu tương tự [12]. Để so sánh với HGAICS, GA và ICS cũng sử dụng 100 quần thể, hệ số lai chéo và đột biến lần lượt là 0,8 và 0,1. Tiêu chí dừng của các vòng lặp trong cả ba thuật toán được thiết lập như sau: độ lệch của giá trị hàm mục tiêu giữa hai lần lặp liên tiếp thấp hơn  $10^{-3}$  hoặc số lần lặp tối đa là 100.



Hình 9. Sai số giữa mô hình số và kết quả thực đo (a) GA; (b) ICS ; (c) HGAICS

Bảng 2. Các chỉ số đánh giá hiệu quả của các thuật toán.

Thuật toán	Sai số của hàm mục tiêu	Thời gian (giây)
GA	0,90	68212
ICS	0,58	12023
HGAICS	0,17	14314

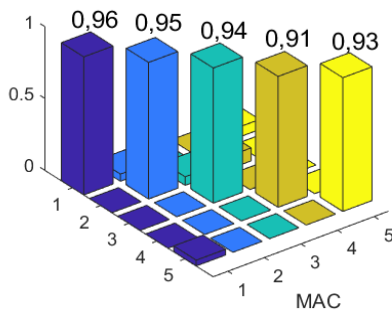
Bảng 2 và Hình 9 cho thấy, HGAICS có độ chính xác vượt trội hơn hẳn GA và ICS. Cụ thể, độ lệch giữa mô hình số và thực đo sau khi cập nhật mô hình sử dụng HGAICS là 0.17, trong khi đó sai số khi sử dụng ICS và GA lần lượt là 0,58 và 0,90. Ngoài ra ICS và HGAICS cũng giảm thời gian tính toán so với GA, cụ thể ICS và HGAICS lần lượt mất 12023s và 14314s để chạy 100 bước lặp, trong khi đó GA phải dùng tới 68212s. Kết quả này có thể được giải thích dựa trên cách tiếp cận để tìm ra giải pháp tốt nhất trong cả 3 thuật toán. GA làm việc chủ yếu dựa trên tối ưu cục bộ, không gian tìm kiếm ngẫu nhiên giới hạn. Hơn nữa qua mỗi bước, bởi vì có quá nhiều tham số phải điều chỉnh, quá trình tính toán thường rất dài. Với ICS, việc áp dụng linh hoạt độ dài bước  $\alpha$  đã giúp lựa chọn được vận tốc phù hợp cho mỗi giai đoạn, từ đó nâng cao độ chính xác và giảm thời gian tính toán. Tuy nhiên bởi vì ICS phụ thuộc vào chất lượng của quần thể ban đầu và không có các khả năng như giao chéo hay đột biến của GA thể cải thiện chất lượng của các quần thể thế hệ mới nên giảm độ chính xác của



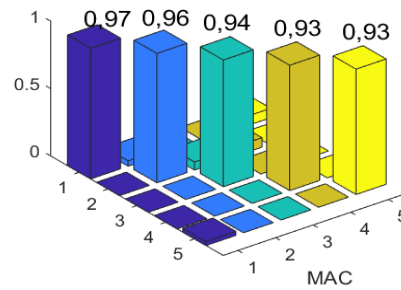
kết quả thu được. Trong khi đó thuật toán đề xuất HGAICS có thể kết hợp ưu điểm của 2 phương pháp GA (có khả năng lai chéo và đột biến để nâng cao chất lượng của các quần thể) và phương pháp ICS (khả năng tìm kiếm toàn cục ngẫu nhiên), vì vậy có thể khắc phục được nhược điểm của thuật toán GA và CS riêng rẽ, cải thiện được độ chính xác của kết quả thu được.

Bảng 3. Các tần số phương thức từ mô hình FEM sau khi cập nhật mô hình so với phép đo

Mốt	Trước khi cập nhật mô hình (Hz)	Cập nhật mô hình-GA (Hz)	Cập nhật mô hình- ICS (Hz)	Cập nhật mô hình- HGAICS (Hz)	Thực nghiệm (Hz)
1	2,99 (2,15%)	3,00 (1,90%)	3,03 (0,98%)	3,07 (0,33%)	3,06
2	4,20 (7,82%)	4,21 (7,59%)	4,25 (6,79%)	4,32 (5,27%)	4,56
3	5,44 (4,31%)	5,46 (4,07%)	5,51 (3,16%)	5,60 (1,66%)	5,69
4	8,68 (3,41%)	8,71 (3,16%)	8,79 (2,22%)	8,92 (0,74%)	8,99
5	10,63 (3,29%)	10,78 (4,63%)	10,87 (5,42%)	10,93 (5,94%)	10,28



(a)



(b)

Hình 10. Giá trị MAC của hình thái dao động sau khi cập nhật mô hình: (a) ICS; (b) HGAICS

Bảng 3 và Hình 10 tổng hợp kết quả tần số dao động tự nhiên và giá trị MAC của dao động từ mô hình số và thực nghiệm. Qua kết quả có một số nhận xét:

- Sau khi cập nhật mô hình, kết quả thu được giữa mô hình số và thực nghiệm đạt được sự tương đồng cao.
- Các giá trị MAC từ 0,91 đến 0,97 (Hình 10) cho thấy sự tương ứng chặt chẽ giữa các hình thái dao động của mô hình số và thực nghiệm sau khi cập nhật mô hình. Giá trị MAC thể hiện quan hệ hình dạng dao động của mô hình số và mô hình thực nghiệm, giá trị MAC nằm trong khoảng (0,1); giá trị tiến gần 1 thể hiện mô hình số và mô hình thực nghiệm có sự tương đồng cao; ngược lại giá trị MAC tiến về 0 thể hiện rằng mô hình số và mô hình thực nghiệm không có sự tương đồng.
- Kết quả của tần số dao động tự nhiên và giá trị MAC của mô hình số xác định bởi HGAICS gần với kết quả thực nghiệm hơn so với GA và ICS.

Quá trình cập nhật mô hình cũng điều chỉnh các thông số bất định của cầu (Bảng 5). Những thay đổi trong bảng 5 cho thấy các thông số trước và sau khi cập nhật không chênh

lệch quá nhiều. Độ cứng của gối cầu và lò xo  $k_1, k_2, k_3, k_4, k_5, k_6$  và  $k_7$  có xu hướng tăng dần. Điều này chứng tỏ rằng độ cứng của gối cầu và lò xo đã được đánh giá bé hơn giá trị thực tế.

Bảng 4. Giá trị của các tham số bất định trước và sau khi cập nhật mô hình

	$k_1$ ( $10^{10}$ N. m/rad)	$k_2$ ( $10^{10}$ N. m/rad)	$k_3$ ( $10^{10}$ N. m/rad)	$k_4$ ( $10^{10}$ N. m/rad)	$k_5$ ( $10^7$ N/ m)	$k_6$ ( $10^7$ N/ m)	$k_7$ ( $10^5$ N. m/rad)	$E$ ( $10^5$ MPa)
Biên trên	1,0	1,0	1,0	1,0	1,0	1,0	7,0	1,9
Trước cập nhật	1,1	1,1	1,1	1,1	1,4	1,4	8,0	2,00
Sau cập nhật (GA)	1,17	1,18	1,16	1,17	1,47	1,46	7,6	2,02
Sau cập nhật (ICS)	1,22	1,23	1,20	1,22	1,52	1,50	7,3	2,05
Sau cập nhật (HGAICS)	1,23	1,26	1,26	1,30	1,54	1,52	7,1	2,07
Biên dưới	2,0	2,0	2,0	2,0	2,0	2,0	9,0	2,2

#### 4. KẾT LUẬN

Trong nghiên cứu này, mô hình cầu Đò Lèn đã được cập nhật và xác định các tham số bất định bao gồm mô đun đàn hồi của các thanh giàn, độ cứng của gối cầu, độ cứng của các liên kết tại các nút giàn. Thuật toán lai HGAICS cũng được đề xuất để cập nhật mô hình. Thuật toán đề xuất có được ưu điểm của thuật toán GA (tạo các quần thể ban đầu chất lượng tốt) và khả năng tìm kiếm toàn cục của ICS, vì vậy sai số giữa kết quả tính toán và thực đo sử dụng thuật toán HGAICS nhỏ hơn sử dụng thuật toán GA và ICS. Sau khi cập nhật mô hình, kết quả thu được giữa mô hình số và thực nghiệm đạt được sự tương đồng cao (độ lệch lớn nhất giữa tính toán và thực nghiệm bé hơn 10%), đặc biệt với HGAICS bé hơn 6,3%. Kết quả của tần số dao động tự nhiên của mô hình số xác định bởi HGAICS gần với kết quả thực nghiệm hơn so với GA và ICS. Ngoài ra HGAICS cũng giảm thời gian tính toán hơn rất nhiều so với GA.

Mặc dù những kết quả thu được từ nghiên cứu này sử dụng thuật toán HGAICS có độ chính xác cao, tuy nhiên phải thừa nhận rằng, việc áp dụng các thuật toán tối ưu dựa trên nguyên tắc tìm kiếm ngẫu nhiên thường tốn nhiều thời gian để xác định được giải pháp tối ưu hơn các phương pháp khác hiện đại mới như học máy.

#### LỜI CẢM ƠN

Nghiên cứu này được tài trợ bởi Bộ Giáo dục và Đào tạo, trong đề tài mã số B2021-GHA-04.

#### TÀI LIỆU THAM KHẢO

- [1]. N.H. Tran, T.T. Bui, Damage detection in a steel beam structure using particle swarm optimization and experimentally measured results, Science Journal of Transportation, 9 (2019) 3-9.
- [2]. K.C. Xuan, B. Le Xuan, A.N. Truong, H.T. Quang, T.N. Trung, An intelligence-based optimization of the internal burnishing operation for surface roughness and vicker hardness, Tạp chí Khoa học Giao thông vận tải, 72 (2021) 395-410.

- [3]. L. Nguyen-Ngoc et al., Damage detection in structures using Particle Swarm Optimization combined with Artificial Neural Network, *Smart Structures and Systems*, 28 (2021) <http://dx.doi.org/10.12989/sss.2021.28.1.001>
- [4]. H. Tran-Ngoc et al., Efficient Artificial neural networks based on a hybrid metaheuristic optimization algorithm for damage detection in laminated composite structures, *Composite Structures*, 262 (2021) 113339. <https://doi.org/10.1016/j.compstruct.2020.113339>
- [5]. D. H. Nguyen et al., Damage detection in truss bridges using transmissibility and machine learning algorithm: Application to Nam O bridge. *Smart Structures and Systems*, 26 (2020) 35-47. <https://doi.org/10.12989/sss.2020.26.1.035>
- [6]. H. V. Dang et al., Data-driven structural health monitoring using feature fusion and hybrid deep learning, *IEEE Transactions on Automation Science and Engineering*, 18 (2020) 2087-2103. <https://doi.org/10.1109/TASE.2020.3034401>
- [7]. H. Tran et al., An efficient approach for model updating of a large-scale cable-stayed bridge using ambient vibration measurements combined with a hybrid metaheuristic search algorithm, *Smart Structures and Systems*, 25 (2020) 487-499. <http://dx.doi.org/10.12989/sss.2020.25.4.487>
- [8]. V. L. Ho et al., System identification based on vibration testing of a steel I-beam, In *International Conference on Numerical Modelling in Engineering*, Springer, Singapore. (2018) 254-268. [https://doi.org/10.1007/978-981-13-2405-5\\_21](https://doi.org/10.1007/978-981-13-2405-5_21)
- [9]. H. Tran-Ngoc et al., A novel machine-learning based on the global search techniques using vectorized data for damage detection in structures, *International Journal of Engineering Science*, 157 (2020) 103376. <https://doi.org/10.1016/j.ijengsci.2020.103376>
- [10]. C. Na, S.P. Kim, H.G. Kwak, Structural damage evaluation using genetic algorithm, *Journal of Sound and Vibration*, 330 (2011) 2772-2783. <https://doi.org/10.1016/j.jsv.2011.01.007>
- [11]. D.S. Jung, C.Y. Kim, Finite element model updating on small-scale bridge model using the hybrid genetic algorithm, *Structure and Infrastructure engineering*, 9 (2013) 481-495. <https://doi.org/10.1080/15732479.2011.564635>
- [12]. H. Tran-Ngoc et al., An efficient artificial neural network for damage detection in bridges and beam-like structures by improving training parameters using cuckoo search algorithm, *Engineering Structures*, 199 (2019) 109637. <https://doi.org/10.1016/j.engstruct.2019.109637>
- [13]. H. Tran-Ngoc et al., Model updating for Nam O bridge using particle swarm optimization algorithm and genetic algorithm, *Sensors*, 18 (2018) 4131. <https://doi.org/10.3390/s18124131>
- [14]. H. Xu, J. K. Liu, Z.R. Lu, Structural damage identification based on cuckoo search algorithm, *Advances in Structural Engineering*, 19 (2016) 849-859. <https://doi.org/10.1177/1369433216630128>
- [15]. H. Tran-Ngoc et al., Finite element model updating of a multispan bridge with a hybrid metaheuristic search algorithm using experimental data from wireless triaxial sensors, *Engineering with Computers*, (2021) 1-19. <https://doi.org/10.1007/s00366-021-01307-9>
- [16]. X. S. Yang, S. Deb, Engineering optimisation by cuckoo search, *International Journal of Mathematical Modelling and Numerical Optimisation*, 1 (2020) 330-343. <https://doi.org/10.1504/IJMMNO.2010.035430>