# Constrained Optimization of Structures using Firefly Algorithm with Penalty Functions

**Article** · January 2015

CITATION
1

READS
1,319

**2 authors:**

Nhat-Duc Hoang
Duy Tan University
**150** PUBLICATIONS **1,979** CITATIONS
SEE PROFILE

Thang Duy Vu
Mien Tay Construction University, Vietnam
**16** PUBLICATIONS **64** CITATIONS
SEE PROFILE

TẠP CHÍ KHOA HỌC & CÔNG NGHỆ
Journal of Science and Technology 2(15) (2015) 75–84

TRƯỜNG ĐẠI HỌC DUY TÂN
DUYTAN UNIVERSITY

# Tối ưu hóa kết cấu có điều kiện ràng buộc sử dụng thuật toán bầy đom đóm và các hàm phạt

## Constrained Optimization of Structures using Firefly Algorithm with Penalty Functions

Hoàng Nhật Đức[a], Vũ Duy Thắng[b]

[a]*Viện Nghiên cứu và Phát triển Công nghệ cao & Khoa Xây dựng, Đại học Duy Tân, Việt Nam*
*Institute of Research and Development & Faculty of Civil Engineering , Duy Tân University, Vietnam*
[b]*Khoa Kiến Trúc, Đại học Duy Tân, Việt Nam*
*Faculty of Architecture, Duy Tân University, Vietnam*

**Tóm tắt**

Trong công nghiệp xây dựng, tối ưu hóa kết cấu là một nhiệm vụ quan trọng trong công tác thiết kế. Mục tiêu của công tác là lựa chọn ra các giải pháp kết cấu có chi phí tối ưu và đảm bảo các ràng buộc về điều kiện bền và chuyển vị. Thuật toán bầy đom đóm là một công cụ hiệu quả cho tối ưu hóa toàn cục. Vì vậy, nghiên cứu của chúng tôi đề xuất một mô hình sử dụng thuật toán này kết hợp với các hàm phạt trong việc giải các bài toán tối ưu hóa kết cấu xây dựng. Kết quả thực nghiệm cho thấy mô hình được đề xuất là một công cụ hữu hiệu để trợ giúp các kỹ sư trong việc thiết kế kết cấu.

*Từ khóa:* Tối ưu hóa kết cấu, thuật toán bầy đom đóm, xử lý ràng buộc, trí tuệ bầy đàn, hàm phạt.

**Abstract**

In construction industry, structure optimization is an important task in designing phase. The objective of this task is to determine a desirable structure with the optimal cost that ensure constraints of stress and deformation. The Firefly Algorithm (FA) is an effective tool for global optimization. Thus, our research proposed an optimization model based on the FA and penalty functions for solving structure optimization. Experimental results show that the proposed model is a useful method to assist engineers in structure design.

*Keywords:* Structure optimization, Firefly Algorithm, Constraint handling, Swarm Intelligence, Penalty function.



## 1. Mở đầu

Tối ưu hóa kết cấu là một nhiệm vụ quan trọng trong công tác thiết kế xây dựng. Vấn đề này xuất hiện khi trong nhiều phương án khả thi, người kỹ sư phải chọn lựa ra phương án kết cấu phải thỏa mãn các điều kiện như trọng lượng tối thiểu, đảm bảo độ cứng, có hiệu quả kinh tế, có tính thẩm mỹ [1, 2]. Căn cứ vào biến số thiết kế và hàm mục tiêu, tối ưu hóa được chia thành tối ưu tiết diện (liên tục và rời rạc) và tối ưu hình dáng [3]. Nghiên cứu của chúng tôi tập trung giải

*Email:* hoangnhatduc@duytan.edu.vn (Hoàng Nhật Đức)

quyết đến bài toán tối ưu tiết diện với hàm mục tiêu là cực tiểu hóa trọng lượng hay chi phí của kết cấu.

Trong bài toán thiết kế tối ưu tiết diện của kết cấu, nếu các biến số là liên tục thì vấn đề được mô phỏng như là một bài toán tối ưu hóa toàn cục. Đây là một bài toán phức tạp do không gian tìm kiếm lớn, sự phức tạp của hàm mục tiêu cũng như các ràng buộc, thách thức trong việc tìm ra các giải pháp vừa làm tối thiểu hóa hàm mục tiêu và vừa phải thỏa mãn nhiều ràng buộc [4–7]. Các phương pháp toán học truyền thống cho tối ưu hóa có ràng buộc có nhiều nhược điểm. Một số phương pháp (nhân tử Lagrange, Gradient, mặt phẳng cắt) chỉ khả dụng cho các hàm mục tiêu là các hàm lồi và chỉ giải được các vấn đề có số biến thiết kế nhỏ [1]. Các bài toán tối ưu kết cấu trong thực tế thường có hàm mục tiêu, các ràng buộc, và các biến thiết kế rất đa dạng (hàm lồi/lõm/gián đoạn, tuyến tính/phi tuyến, liên tục/rời rạc). Vì vậy, thực tế đòi hỏi các công cụ tốt hơn, tổng quát hơn, và mạnh hơn để giải quyết các bài toán tối ưu kết cấu.

Thuật toán bầy đom đóm, được đề xuất bởi Yang [8], là một thuật toán dựa trên trí tuệ bầy đàn. Thuật toán này lấy ý tưởng từ sự tương tác của đàn đom đóm trong tự nhiên thông qua sự phát sáng của chúng. Thuật toán đom đóm đã được minh chứng là rất hiệu quả cho các bài toán tối ưu hóa toàn cục [9]. Tuy vậy, các nghiên cứu về việc sử dụng thuật toán này cho tối ưu hóa kết cấu chịu điều kiện ràng buộc còn rất giới hạn.

Thêm vào đó, để xử lý các ràng buộc khi sử dụng các thuật toán bầy đàn, các hàm phạt là một phương pháp phổ biến. Phương pháp này đơn giản, dễ sử dụng, và có thể áp dụng cho tất cả các loại ràng buộc (đẳng thức hay bất đẳng thức, tuyến tính hay phi tuyến, liên tục hay gián đoạn). Ý tưởng của phương pháp này là biến đổi dạng ban đầu của hàm mục tiêu bằng cách cộng thêm các giá trị nào đó (gọi là các giá trị phạt) vào hàm mục tiêu của các cá thể nếu như chúng không thỏa mãn điều kiện ràng buộc. Nếu các cá thể ở càng xa miền hợp lệ, các giá trị phạt càng tăng lên và ngược lại. Nếu các cá thể ở trong miền hợp lệ, thì các giá trị phạt bằng 0 [10].

Dẫu vậy, mặt khó khăn của phương pháp này là làm sao có thể xác định hệ số phạt một cách phù hợp [6]. Để giải quyết khó khăn này, bên cạnh phương pháp hàm phạt tĩnh [10], nhiều nghiên cứu đã đề xuất các phương pháp phức tạp để xác định hệ số phạt sử dụng trong các hàm phạt như phương pháp hàm phạt động [11], hàm phạt tự thích nghi [12]. Michalewicz [13] cho rằng nên sử dụng một hệ số phạt (hàm phạt tĩnh) vì các phương pháp hàm phạt động cho các bài toán khác nhau thường cho các kết quả khác nhau. Coello [5] so sánh các phương pháp sử dụng hàm phạt kết hợp với thuật toán tiến hóa và chỉ ra rằng xác định hệ số phạt cho các hàm phạt phải phụ thuộc vào từng bài toán cụ thể. Tuy vậy, rất ít nghiên cứu thực nghiệm đánh giá sự hiệu quả của từng loại hàm phạt cho bài toán tối ưu hóa kết cấu. Các hướng dẫn tổng quát cho việc hiệu chỉnh hệ số phạt khi giải các bài toán tối ưu kết cấu chịu các ràng buộc còn rất thiếu.

Vì vậy, để thúc đẩy sự sử dụng và nâng cao hiệu quả sử dụng của thuật toán bầy đàn và phương pháp xử lý ràng buộc bằng các hàm phạt trong tối ưu hóa kết cấu, nghiên cứu của chúng tôi có các mục đích sau:

- Thiết lập một mô hình tối ưu hóa kết cấu dựa trên thuật toán bầy đom đóm và phương thức xử lý ràng buộc bằng các hàm phạt.
- Khảo sát sự hiệu quả của các hàm phạt (hàm tĩnh, hàm động, hàm tự thích nghi) bằng các thí nghiệm tính toán.
- Đề xuất các hướng dẫn đơn giản dễ sử dụng để hiệu chỉnh hệ số phạt giúp cho các kỹ sư kết cấu trong việc sử dụng mô hình tối ưu hóa.

Phần còn lại của bài báo được trình bày như sau: Phần 2 của bài báo trình bày phương pháp nghiên cứu. Phần 3 đề xuất mô hình tối ưu hóa dựa trên dựa trên thuật toán bầy đom đóm và phương thức xử lý ràng buộc bằng các hàm phạt. Kết quả tính toán thực nghiệm được trình bày trong phần tiếp theo. Các kết luận của nghiên cứu được nêu trong phần cuối của bài báo.

## 2. Phương pháp nghiên cứu

### *2.1. Mô hình hóa bài toán tối ưu hóa có điều kiện ràng buộc*

Bài toán tối ưu hóa kết cấu tổng quát được mô hình hóa như sau [1]:

Cực tiểu hóa:

$$f(x_1\ x_2, \ldots, x_d),\ d = 1, 2, \ldots, D$$

Chịu các ràng buộc:

$g_q(x_1\ x_2, \ldots, x_d) \le 0$

$h_r(x_1\ x_2, \ldots, x_d) = 0$

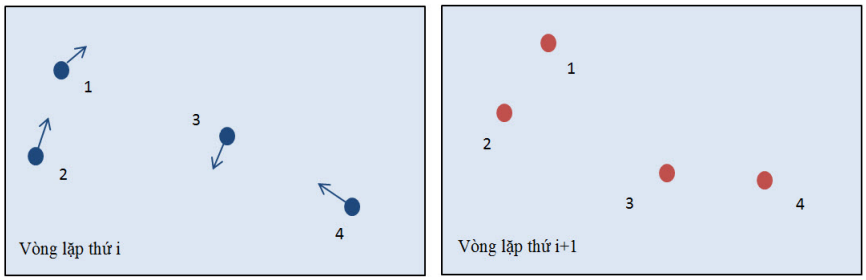
$x_d^L \le x_d \le x_d^U$

với $q = 1, 2, \ldots, M$; $r = 1, 2, \ldots, N$.

Trong đó, $f(x_1\ x_2, \ldots, x_d)$ là hàm mục tiêu, thường là khối lượng hay chi phí của kết cấu; $x_1\ x_2, \ldots, x_d$ là các biến số thiết kế; $g_q(x_1\ x_2, \ldots, x_d)$ và $h_r(x_1\ x_2, \ldots, x_d)$ là các ràng buộc (về mặt chịu lực, biến dạng); $x_d^L$ và $x_d^U$ là các cận trên và cận dưới của biến thiết kế $x_d$; $D$ là số lượng biến số thiết kế; $M$ và $N$ là số lượng ràng buộc bất đẳng thức và đẳng thức.

### 2.2. *Thuật toán bầy đom đóm*

Đom đóm là một nhóm loài côn trùng cánh cứng nhỏ (trong tự nhiên có khoảng 2000 loài) có khả năng phát quang. Đom đóm là loài có tập tính hoạt động về ban đêm. Ánh sáng của đàn đom đóm là một cảnh tượng thường thấy trong bầu trời đêm vào mùa hè ở các vùng nhiệt đới. Sự phát sáng ở đom đóm là nhờ một loại phản ứng hóa học, được gọi là ánh sáng sinh học. Tiến trình này xảy ra trong cơ quan phát sáng chuyên biệt, thường nằm ở dưới bụng đom đóm. Đom đóm một quần thể dùng ánh sáng sinh học cho chọn lọc giới tính; vì vậy, ánh sáng được dùng cho việc lựa chọn bạn tình [14].




Hình 1. Sự di chuyển của bầy đom đóm

**Bắt đầu thuật toán**
Định nghĩa hàm mục tiêu $f(x)$, trong đó $x = (x_1,\ldots,x_d)$
Khởi tạo quần thể đom đóm
Tính toán cường độ sáng của cá thể $I$
Định nghĩa hệ số hấp thụ ánh sáng γ
**While** ($t$ < Số vòng lặp lớn nhất)
**For** $i$ = 1 to n (n = số lượng cá thể)
  **For** $j$=1 to n (n = số lượng cá thể)
    **If** ($I_j > I_i$), di chuyển cá thể $i$ đến gần cá thể $j$
    **End** if
    Đánh giá cá thể mới và cập nhật cường độ sáng;
  **End for** $j$
**End for** $i$
Xếp hạng các cá thể đom đóm và tìm ra cá thể tốt nhất
**End while**;
**Kết thúc thuật toán**

Hình 2. Thuật toán bầy đom đóm

Thuật toán bầy đom đóm được phát triển bởi Yang [8], lấy ý tưởng từ đặc tính tự nhiên của bầy đom đóm. Thuật toán này được thiết kế để giải các bài toán tối ưu hóa toàn cục trong đó mỗi cá thể đom đóm trong quần thể tương tác với nhau qua cường độ ánh sáng của chúng. Trong

quá trình tìm bạn giao phối, mỗi cá thể bị thu hút và di chuyển tới vị trí các cá thể khác phát sáng hơn nó. Bằng thực nghiệm, thuật toán bầy đom đóm đã được minh chứng là tốt hơn so với thuật toán bầy đàn.

Sự thu hút của một cá thể đom đóm tỷ lệ với cường độ sáng của nó. Dễ thấy, sự thu hút này càng giảm đối với một cá thể đom đóm khác ở càng xa vị trí của nó thì càng giảm. Nếu một cá thể không tìm được cá thể khác phát sáng hơn nó, cá thể đó sẽ di chuyển một cách ngẫu nhiên trong không gian tìm kiếm. Hình 1 mô tả sự di chuyển của các cá thể đom đóm trong một quần thể có 4 cá thể. Cường độ sáng của cá thể k được ký hiệu là $I(k)$. Giả sử $I(1) > I(2)$ và $I(3) > I(4)$. Các cá thể 2, 4 di chuyển đến gần các cá thể sáng hơn nó. Còn các cá thể 1 và 3 di chuyển một cách ngẫu nhiên.

Thuật toán quy định rằng sự phát sáng của một cá thể bị ảnh hưởng bởi vị trí của nó trong không gian tìm kiếm. Vì vậy, một cá thể ở vị trí có hàm mục tiêu càng nhỏ thì càng sáng và càng có khả năng thu hút các cá thể đom đóm khác đến với mình [15]. Thuật toán bầy đom đóm được miêu tả trong Hình 2.

Cường độ sáng $I$ được tính toán như sau:

$$I = I_S e^{-\gamma r^2} \quad (1)$$

Trong đó, $I_s$= cường độ sáng tại nguồn sáng; $\gamma$ = hệ số hấp thụ ánh sáng; $r$ = khoảng cách tới nguồn sáng.

Vì sự hấp dẫn của một cá thể đom đóm tỷ lệ thuận với ánh sáng phát từ nó, sự hấp dẫn của một cá thể (ký hiệu là $\beta$) được định nghĩa như sau:

$$\beta = \beta_0 e^{-\gamma r^2} \quad (2)$$

Khoảng cách giữa cá thể đom đóm $i$ tại vị trí $x_i$ và cá thể $j$ tại vị trí $x_j$ được tính toán như sau:

$$r_{ij} = \left\| x_i - x_j \right\| = \sqrt{\sum_{k=1}^{d} \left( x_{ik} - x_{jk} \right)^2} \quad (3)$$

Trong đó, $d$ = số chiều của không gian tìm kiếm; ||.|| là ký hiệu của khoảng cách Euclid.

Sự di chuyển của cá thể $i$ khi bị thu hút bởi một cá thể sáng hơn $j$ được định nghĩa như sau:

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2} \left( x_j - x_i \right) + \alpha \left( rand - 0.5 \right) \quad (4)$$

Trong đó, $x_i^{g+1}$ và $x_i^g$ là vị trí của cá thể $i$ tại vòng lặp thứ $g+1$ và $g$; $x_j^g$ là vị trí của cá thể $j$ tại vòng lặp $g+1$ và $g$; $\gamma$ = hệ số hấp thụ sáng (0.1 ~ 10); $\beta_0$ = sự hấp dẫn tại $r_{ij} = 0$; $\alpha$ = hệ số ảnh hưởng tới chuyển động ngẫu nhiên của đom đóm; $rand$ = 1 số ngẫu nhiên thuộc phân phối chuẩn.

### *2.3. Xử lý ràng buộc với các hàm phạt*

Tối ưu hóa có ràng buộc là bài toán thường gặp trong tối ưu hóa kết cấu. Để xử lý các ràng buộc khi sử dụng thuật toán đàn đom đóm, nghiên cứu của chúng tôi sử dụng các hàm phạt. Lý do là vì kỹ thuật này khá đơn giản và có thể áp dụng cho tất cả các loại ràng buộc. Để hướng các cá thể đom đóm vào miền hợp lệ, ta biến đổi dạng ban đầu của hàm mục tiêu bằng cách cộng thêm các giá trị phạt nếu như cá thể đó không thỏa mãn điều kiện ràng buộc. Khi các cá thể ở trong miền hợp lệ, thì các giá trị phạt bằng 0. Đối với các cá thể ở càng xa miền hợp lệ, các giá trị phạt càng tăng lên.

Để sử dụng phương pháp này, chúng ta cần phải xác định giá trị của hàm phạt, thông qua hệ số phạt, một cách phù hợp. Việc xác định giá trị của hàm phạt là tùy thuộc vào từng vấn đề tối ưu hóa cụ thể [13]. Để giải quyết vấn đề này, các loại hàm phạt khác nhau đã đươc đề xuất: hàm phạt tĩnh [10], hàm phạt động [11], và hàm phạt tự thích nghi [12].

Giả sử như thuật toán bầy đom đom đang được sử dụng để giải quyết vấn đề tối ưu hóa sau:

Tìm cực tiểu của:

$$f(x) = (x_1 - 1)^2 + (x_2 + 1)^2 \quad (5)$$

Chịu ràng buộc:

$g_1 = 2 - x_1 - x_2 \le 0$ và $g_2 = x_1 - x_2 - 3 \le 0$

Gọi $f_o(x)$ là hàm mục tiêu ban đầu:

$$f_o(x) = (x_1 - 1)^2 + (x_2 + 1)^2 \quad (6)$$

Hàm mục tiêu mới $f_n(x)$ được định nghĩa như sau:

$$f_n(x) = f_o(x) + P_1(x) + P_2(x) \quad (7)$$

Trong đó, $P_1(x)$ và $P_2(x)$ là các hàm phạt tương ứng với ràng buộc 1 và 2.

Hàm mục tiêu mới $f_n(x)$ có thể được định nghĩa một cách tổng quát như sau:

$$f_n(x) = f_o(x) + \sum_{i=1}^{m} P_i(x) \quad (8)$$

Trong đó, $m$ = số lượng ràng buộc. $P_i(x)$ là hàm phạt tương ứng với ràng buộc thứ $i$.

Giả sử ràng buộc thứ $i$ có dạng: $g_i(x) \leq 0$, hàm phạt $P_i(x)$ được định nghĩa như sau:

$$P_i(x) = a_i.max(0, g_i(x)) \quad (9)$$

Trong đó, $a_i$ được gọi là hệ số phạt. Dễ thấy, khi $g_i(x) \leq 0$ thì $max(0, g_i(x)) = 0$ và $P_i(x) = 0$. Do vậy, $f_n(x) = f_o(x)$. Mặt khác, khi $g_i(x) > 0$ thì $max(0, g_i(x)) > 0$ và $P_i(x) > 0$. Do vậy, $f_n(x) > f_o(x)$, lượng khác biệt giữa hai đại lượng chính là giá trị của hàm phạt do cá thể đã vi phạm ràng buộc. Nói cách khác, cá thể đã bị phạt vì đi vào vùng không hợp lệ trong miền tìm kiếm. Chú ý rằng đối với các ràng buộc dạng bất đẳng thức, ta luôn có thể quy đổi về dạng $g_i(x) \leq 0$. Ví dụ, dạng gốc của ràng buộc là $g_i = 4–x_1 \geq 0$, ta có thể biến đổi một cách tương đương thành dạng: $g_i = x_1–4 \leq 0$. Đối với dạng ràng buộc là đẳng thức, ta có thể quy đổi thành một cặp ràng buộc bất đẳng thức. Nếu dạng gốc của ràng buộc là $g_i = 4–x_1 = 0$. Ta có thể quy đổi thành: $g_{i1} = 4–x_1 \leq 0$ và $g_{i2} = 4–x_1 \geq 0$. Hiển nhiên, nếu giá trị x thỏa mãn cả 2 ràng buộc $g_{i1}$ và $g_{i2}$, nó sẽ thỏa mãn ràng buộc $g_i$.

Các loại hàm phạt khác nhau chủ yếu chỉ khác nhau ở cách thức xác định hệ số phạt $a_i$. Phần tiếp theo của bài báo sẽ trình bày 3 dạng hàm phạt.

*2.3.1. Hàm phạt tĩnh*

Đối với hàm phạt tĩnh, hệ số phạt $a_i$ được xác định trước khi quá trình tối ưu hóa bắt đầu và giữ nguyên giá trị trong suốt quá trình tìm kiếm: $a_i = const$.

*2.3.2. Hàm phạt động*

Trong hàm phạt động, hệ số phạt ban đầu được xác định trước khi quá trình tối ưu hóa bắt đầu và giá trị của nó thay đổi theo thời gian tìm kiếm:

$$a_i = f(g) \quad (10)$$

Trong đó, $g$ = số vòng lặp tính đến thời điểm hiện tại. Dễ thấy hệ số phạt là hàm của số vòng lặp, hàm này có thể là tuyến tính hay phi tuyến. Một cách đơn giản, ta có thể sử dụng một hàm bậc nhất sau:

$$a_i = a_{si}.g \quad (11)$$

Trong đó, $a_{si}$ là bước tăng của hệ số phạt tương ứng với ràng buộc $i$. Ý nghĩa của hàm phạt này là: ở quãng thời gian ban đầu, hệ số phạt là nhỏ để cho phép các cá thể tự do khám phá không gian tìm kiếm; ở quãng thời gian sau, hệ số phạt nên tăng dần để hạn chế các cá thể rơi vào vùng không hợp lệ.

*2.3.3. Hàm phạt tự thích nghi*

Hàm phạt tự thích nghi cho phép hệ số phạt tăng hay giảm so với vòng lặp trước. Bean và Hadj-Alouane đề xuất quy luật sau:

- Nếu các cá thể tốt nhất trong $k$ thế hệ gần nhất đều vi phạm ràng buộc thì tăng hệ số phạt:

$$a_{i,g+1} = a_{i,g}.b_1 \quad (12)$$

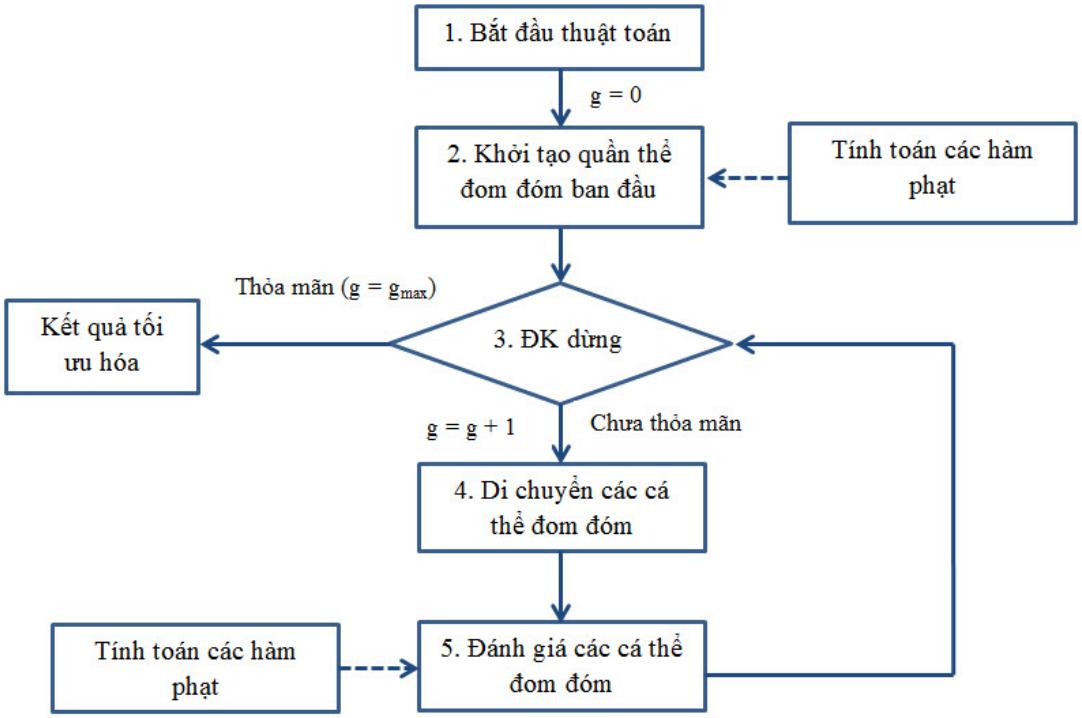
- Nếu các cá thể tốt nhất trong $k$ thế hệ gần nhất đều thỏa mãn ràng buộc thì giảm hệ số phạt:

$$a_{i,g+1} = a_{i,g}/b_2 \quad (13)$$

Trong đó, $a_{i,g}$ là hệ số phạt tại vòng lặp $g$ của ràng buộc thứ $i$; $b_1 > 1$ và $b_2 > 1$ là các hệ số tăng và hệ số giảm.

## 3. Mô hình tối ưu hóa dựa trên thuật toán bầy đom đóm và các hàm phạt

Mô hình tối ưu hóa dựa trên thuật toán đom đóm và các hàm phạt, (ĐĐHF) được cho trong Hình 3. Mô hình này sử dụng thuật toán đom đóm để tìm kiếm lựa chọn tốt nhất của các biến thiết kế trong không gian tìm kiếm. Thêm vào đó, các hàm phạt được xử dụng để xử lý các ràng buộc của bài toán tối ưu hóa kết cấu.




Hình 3. Tối ưu hóa dựa trên thuật toán đom đóm và các hàm phạt (ĐĐHF)

**(1) Bắt đầu thuật toán**: Các thông số của thuật toán bao gồm số vòng lặp tối đa ($g_{max}$), số cá thể đom đóm trong quần thể ($N$), hệ số hấp thụ ánh sáng ($\gamma$), loại và thông số của hàm phạt được quy định tại bước này. Thông thường, các thông số này có thể được đặt như sau: $N = 5d$, trong đó $d$ là số biến thiết kế. $\gamma = 1.g_{max}$ cần đủ lớn để cho quá trình tối ưu hóa hội tụ.

**(2) Khởi tạo quần thể đom đóm ban đầu**: Các cá thể đom đóm trong vòng lặp đầu tiên được khởi tạo một cách ngẫu nhiên trong khoảng cho phép của từng biến thiết kế. Cũng tại bước này, quần thể đom đóm được đánh giá thông qua hàm mục tiêu cộng với giá trị của các hàm phạt.

**(3) Kiểm tra điều kiện dừng**: Nếu vòng lặp hiện tại $g < g_{max}$ thì quá trình tối ưu hóa tiếp tục diễn ra. Khi điều kiện dừng được thỏa mãn, biến số thiết kế tốt nhất đã được tìm ra bởi thuật toán.

**(4) Di chuyển các cá thể đom đóm**: Các cá thể di chuyển đến cá thể đom đóm khác sáng hơn. Nếu không tìm được cá thể nào sáng hơn, cá thể sẽ di chuyển ngẫu nhiên trong không gian tìm kiếm.

**(5) Đánh giá các cá thể đom đóm**: Các cá thể sẽ được đánh giá và xếp loại để tìm ra các cá thể tốt hơn. Thông tin của cá thể tốt nhất sẽ được lưu lại.

### *3.1. Tìm cực trị của hàm đơn giản*

Tìm cực tiểu của:

$$f(x) = (x_1-1)^2 + (x_2-2)^2 + 10 \qquad (14)$$

Chịu ràng buộc:

$g_1(x) = 4-x_1-x_2 \le 0;$

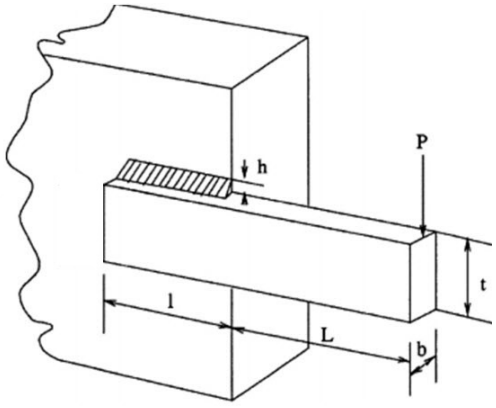
$g_2(x) = x_1-x_2-1 \le 0;$

$-100 \le x_1, x_2 \le 100.$

Dễ thấy cực trị của bản thân hàm $f(x)$ trên $R$ là $f_{min} = 10$ và $x = [1, 2]$. Còn khi xét đến các ràng buộc thì $f_{min} = 10.5$ và $x = [1.5, 2.5]$.

### *3.2. Thiết kế dầm thép hàn*

Bài toán thiết kế tối ưu hóa chi phí xây dựng của một dầm thép hàn [1] (xem Hình 4) có 4 biến thiết kế ($\mathbf{h}, \mathbf{l}, \mathbf{b}, \mathbf{t}$). Mục tiêu của bài toán là tối thiểu hóa chi phí của dầm thép chịu các ràng buộc về ứng suất cắt ($T$), ứng suất uốn ($S$), tải trọng tập trung ($P$), độ võng ($D$), và các ràng buộc khác về kích thước kết cấu. Vấn đề tối ưu hóa này được mô hình hóa như sau:




Hình 4. Bài toán thiết kế dầm thép hàn

Tìm cực tiểu của:

$$f(x) = 1.10471\mathbf{h}^2.\mathbf{l} + 0.04811\mathbf{t}.\mathbf{b}.(14.0 + \mathbf{l}) \quad (15)$$

Chịu các ràng buộc sau:

$g_1(x) = T(x)–13600 \le 0;$

$g_2(x) = S(x)–30000 \le 0;$

$g_3(x) = \mathbf{h}–\mathbf{b} \le 0;$

$g_4(x) = 6000–P_c(x) \le 0;$

$g_5(x) = D(x)–0.25 \le 0;$

$0.125 \le \mathbf{h} \le 10;\ 0.1 \le \mathbf{l}, \mathbf{b}, \mathbf{t} \le 10.$

Deb [16] sử dụng thuật toán di truyền tìm được kết quả sau:

• Các biến thiết kế: $\mathbf{h} = 0.2489$, $\mathbf{l} = 6.1730$, $\mathbf{t} = 8.1789$, và $\mathbf{b} = 0.2533$

• Giá trị của hàm mục tiêu: $f = 2.43$

### *3.3. Thiết kế lò xo chịu nén/kéo*

Bài toán thiết kế lò xo chịu nén/kéo [17] (xem hình 5) có 3 biến thiết kế là đường kính lò xo ($D$), đường kính dây ($d$), và số cuộn trong lò xo ($P$). Đặt $x_1 = D$, $x_2 = d$, và $x_3 = P$, bài toán được cho như sau:


Hình 5. Bài toán thiết kế lò xo

Tìm cực tiểu của:

$$f(x) = (x_3 + 2)\, x_2 x_1^2 \quad (16)$$

Chịu các ràng buộc sau:

$$g_1(x) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \le 0;$$

$$g_2(x) = \frac{4x_2^2 - x_1 x_2}{12566\left(x_2 x_1^3 - x_1^4\right)} + \frac{1}{5108 x_1^2} \le 0;$$

$$g_3(x) = 1 - \frac{140.45 x_1}{x_2^2 x_3} - 1 \le 0;$$

$$g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \le 0;$$

$$0.05 \le x_1 \le 0.25;$$

$$0.25 \le x_2 \le 1.3;$$

$$2 \le x_3 \le 15;$$

### *3.4. Kết quả tính toán*

Kết quả tính toán của mô hình ĐĐHF được cho trong các Bảng 1 và Bảng 2. Đối với các ví dụ tính toán hàm đơn giản và dầm thép, số vòng lặp tối đa của thuật toán đom đóm lần lượt là 1000 và 5000. Tương ứng với mỗi loại hàm phạt và các thông số hàm phạt, kết quả tối ưu hóa, được thể hiện thông qua biến thiết kế tìm được, giá trị của hàm mục tiêu, và tình trạng của các ràng buộc (TM = thỏa mãn/ KTM = không thỏa mãn). Đối với vấn đề tối ưu hàm đơn giản, kết quả cho thấy mô hình ĐĐHF đã tìm ra kết quả tối ưu. Đối với vấn đề tối ưu dầm thép, kết quả cho thấy mô hình ĐĐHF ($f = 1.91$) tương ứng với hàm phạt tĩnh đã tìm ra kết quả tốt hơn kết quả được báo cáo trong [16] ($f = 2.43$). Trong đó, hàm phạt tĩnh cho kết quả tốt nhất. Khi sử dụng ĐĐHF để tối ưu lò xo, cả 3 phương pháp xử lý ràng buộc đều cho kết quả $f = 0.01283$ [18], $f = 0.01273$ [17] và tương đương với kết quả được tìm ra trong [19].

Bảng 1. Kết quả tính toán cho tối ưu hàm đơn giản ($g_{max}$ = 1000).

| Loại hàm phạt | Thông số hàm phạt | $x_1$ | $x_2$ | $f$ | RB1 | RB2 |
|---|---|---|---|---|---|---|
| Tĩnh | a = 0.1 | 1.05 | 2.05 | 10.10 | KTM | TM |
| | a = 1 | 1.50 | 2.50 | 10.50 | TM | TM |
| | a = 2 | 1.50 | 2.50 | 10.50 | TM | TM |
| Động | $a_s$ = 0.1 | 1.50 | 2.50 | 10.50 | TM | TM |
| Tự thích nghi | $a_o$ = 0.1; $b_1$ = 1.11; $b_2$ = 1.12 | 1.39 | 2.38 | 10.47 | KTM | TM |
| | $a_o$ = 1; $b_1$ = 1.11; $b_2$ = 1.12 | 1.50 | 2.50 | 10.50 | TM | TM |

Ký hiệu: KMT = không thỏa mãn. TM = thỏa mãn. RB = ràng buộc.

Bảng 2. Kết quả tính toán cho tối ưu dầm thép hàn ($g_{max}$ = 5000)

| Loại hàm phạt | Thông số hàm phạt | $x_1$ ($h$) | $x_2$ ($l$) | $x_3$ ($t$) | $x_4$ ($b$) | $f$ | RB1 | RB2 | RB3 | RB4 | RB5 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Tĩnh | a = 0.1 | 0.18 | 1.34 | 7.99 | 0.10 | 0.96 | KTM | KTM | KTM | KTM | TM |
| | a = 1 | 0.25 | 1.90 | 8.29 | 0.24 | 1.82 | KTM | TM | KTM | TM | TM |
| | a = 10 | 0.24 | 2.65 | 9.07 | 0.24 | 1.91 | TM | TM | TM | TM | TM |
| Động | $a_s$ = 0.1 | 0.23 | 3.02 | 9.06 | 0.24 | 1.95 | TM | TM | TM | TM | TM |
| | $a_s$ = 1 | 0.23 | 3.23 | 9.01 | 0.24 | 1.97 | TM | TM | TM | TM | TM |
| | $a_s$ = 10 | 0.22 | 3.33 | 9.16 | 0.24 | 1.99 | TM | TM | TM | TM | TM |
| Tự thích nghi | $a_o$ = 0.1; $b_1$ = 1.11; $b_2$ = 1.12 | 0.23 | 2.80 | 9.08 | 0.24 | 1.93 | KTM | TM | TM | TM | TM |
| | $a_o$ = 1; $b_1$ = 1.11; $b_2$ = 1.12 | 0.22 | 3.11 | 9.16 | 0.24 | 1.97 | TM | TM | TM | TM | TM |
| | $a_o$ = 5; $b_1$ = 1.11; $b_2$ = 1.12 | 0.23 | 2.65 | 9.28 | 0.24 | 1.93 | TM | TM | TM | TM | TM |
| | $a_o$ = 10; $b_1$ = 1.11; $b_2$ = 1.12 | 0.23 | 3.07 | 8.93 | 0.24 | 1.95 | TM | TM | TM | TM | TM |

Ký hiệu: KMT = không thỏa mãn. TM = thỏa mãn. RB = ràng buộc.

Bảng 3. Kết quả tính toán cho tối ưu lò xo ($g_{max}$ = 1000)

| Loại hàm phạt | Thông số hàm phạt | $x_1$ ($D$) | $x_2$ ($d$) | $x_3$ ($P$) | $f$ | RB1 | RB2 | RB3 | RB4 |
|---|---|---|---|---|---|---|---|---|---|
| Tĩnh | a = 0.1 | 0.0507 | 0.3332 | 12.8140 | 0.0127 | TM | TM | TM | TM |
| | a = 1 | 0.0512 | 0.3458 | 11.9641 | 0.0127 | TM | TM | TM | TM |
| | a = 10 | 0.0557 | 0.4601 | 7.0791 | 0.0129 | TM | TM | TM | TM |
| Động | $a_s$ = 0.1 | 0.0523 | 0.3714 | 10.482 | 0.0127 | TM | TM | TM | TM |
| | $a_s$ = 1 | 0.0562 | 0.4755 | 6.6695 | 0.0130 | TM | TM | TM | TM |
| Tự thích nghi | $a_o$ = 0.1; $b_1$ = 1.11; $b_2$ = 1.12 | 0.052 | 0.3638 | 10.8872 | 0.0127 | TM | TM | TM | TM |
| | $a_o$ = 1; $b_1$ = 1.11; $b_2$ = 1.12 | 0.0537 | 0.4081 | 8.8107 | 0.0127 | TM | TM | TM | TM |

Ký hiệu: KMT = không thỏa mãn. TM = thỏa mãn. RB = ràng buộc.

Đối với hàm phạt tĩnh, nếu giá trị của hệ số phạt quá nhỏ, kết quả tính toán cho thấy các cá thể đom đóm chưa quan tâm đến việc tìm kiếm trong vùng hợp lệ. Khi các giá trị phạt tăng đủ lớn, các kết quả tìm được đều thỏa mãn các ràng buộc. Đối với hàm phạt động, giá trị của bước tăng của hệ số phạt ($a_s$) có ảnh hưởng quyết định đến quá trình tìm kiếm. Giá trị ban đầu bằng 0 và sẽ tăng dần theo số vòng lặp. Theo kết quả tính toán, giá trị của $a_s$ khá nhỏ (0.1) cũng đủ để hướng các cá thể vào việc tìm kiếm giải pháp tối ưu trong vòng hợp lệ. Đối với hàm phạt tự thích

nghi, để đơn giản hóa, nhóm tác giả cố định các hệ số tăng ($b_1 = 1.11$) và hệ số giảm ($b_2 = 1.11$). Có thể thấy trong Bảng 1, khi hệ số phạt ban đầu của hàm phạt tự thích nghi đặt bằng 0.1, thuật toán đã không thể tìm ra giải pháp thỏa mãn ràng buộc thứ nhất. Như vậy, giá trị ban đầu của hàm phạt này cũng cần phải đủ lớn để quá trình tối ưu hóa diễn ra thành công.

## 4. Kết luận

Tối ưu hóa kết cấu chịu nhiều điều kiện ràng buộc là một nhiệm vụ quan trọng và phức tạp trong công tác thiết kế xây dựng. Chúng tôi đề xuất một mô hình tối ưu hóa ĐĐHF dựa trên thuật toán bầy đom đóm và các hàm phạt (tĩnh, động, và tự thích nghi). Nghiên cứu này đánh giá sự hiệu quả của từng loại hàm phạt cho bài toán tối ưu hóa. Thông qua kết quả tính toán một hàm đơn giản và dầm thép, nghiên cứu đưa ra một số kết luận như sau:

• Hàm phạt tĩnh đơn giản, dễ sử dụng, chỉ yêu cầu một tham số. Kết quả của nó phụ thuộc nhiều vào việc xác định hệ số phạt. Tuy nhiên, khi hệ số phạt được xác định phù hợp, hàm phạt này có thể giúp thuật toán đom đóm tìm ra giải pháp có giá trị tốt.

• Hàm phạt động và tự thích nghi có hệ số phạt linh động thay đổi theo thời gian tối ưu hóa và theo tình trạng của quá trình tối ưu. Tuy vậy, hàm tự thích nghi yêu cầu nhiều tham số hơn các hàm phạt tĩnh và hàm phạt động.

• Giá trị của hệ số phạt nên được xác định tùy theo từng vấn đề cụ thể; nhưng một cách tổng quát, giá trị của phạt không quá nhỏ để kéo các cá thể vào vùng hợp lệ; nhưng cũng không quá lớn vì các cá thể sẽ không khám phá vùng bất hợp lệ. Thông tin ở vùng bất hợp lệ có thể rất quan trọng vì điểm cực trị có thể nằm sát đường ranh giới.

• Thuật toán bầy đom đóm là một thuật toán hiệu quả cho việc giúp các kỹ sư xây dựng trong công tác thiết kế kết cấu.

Hướng nghiên cứu tiếp theo của bài báo bao gồm việc cải tiến khả năng tối ưu hóa của thuật toán bầy đom đóm, kết hợp thuật toán này với các kỹ thuật xử lý ràng buộc phức tạp, và sử dụng mô hình để giải quyết các bài toán tối ưu hóa kết cấu khác.

## Tài liệu tham khảo


[1] G. V. Reklaitis, A. Ravindran, and K. M. Ragsdell. *Engineering Optimization Methods and Applications*. Wiley, New York, 1983.

[2] H. Rahami, A. Kaveh, and Y. Gholipour. Sizing, geometry and topology optimization of trusses via force method and genetic algorithm. *Engineering Structures*, 30:2360–2369, 2008.

[3] A. T. Vũ and Q. C. Nguyen. Thiết kế tối ưu kết cấu thép bằng thuật toán tiến hóa. *Tạp trí Khoa hoc và Công nghệ*, 45(4):111–118, 2007.

[4] A. P. Engelbrecht. *Computational Intelligence-An Introduction*. John Wiley & Sons Ltd, 2 edition, 2007.

[5] C. A. Coello. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering*, 191:1245–1287, 1/4 2002.

[6] K. Deb. An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186:311–338, 2000.

[7] N.-D. Hoang. Nide: A novel improved differential evolution for construction project crashing optimization. *Journal of Construction Engineering*, 2014:7, 2014.

[8] X.-S. Yang. *Firefly algorithm*. Luniver Press, Bristol, UK, 2008.

[9] I. Fister, I. Fister Jr, X.-S. Yang, and J. Brest. A comprehensive review of firefly algorithms. *Swarm Evol. Comput.*, 13:34–46, 12 2013.

[10] A. Homaifar, S. H.-V. Lai, and X. Qi. Constrained optimization via genetic algorithms. *Simulation*, 62:242–254, 1994.

[11] J. A. Joines and C. R. Houck. On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with gas. In *Proceedings of the International Conference on Evolutionary Computation*, pages 579–584. IEEE Press, Piscataway, 1994.

[12] J. C. Bean and A. B. Hadj-Alouane. A dual genetic algorithm for bounded integer programs. Technical report, Technical Report TR 92-53, Dept. of Industrial and Operiations Engineering, the University of Michigan, 1992.

[13] Z. Michalewicz. Genetic algorithms, numerical optimization, and constraints. In *Proceedings of the Sixth International Conference on Genetic Algorithms, Morgan Kauman, San Mateo*, pages 151–158, 1995.

[14] K. F. Stanger-Hall, J. E. Lloyd, and D. M. Hillis. Phylogeny of north american fireflies (coleoptera: Lampyridae): Implications for the evolution of light signals. *Molecular Phylogenetics and Evolution*, 45:33–49, 10 2007.

[15] N.-D. Hoang, A.-D. Pham, and M.-T. Cao. A novel time series prediction approach based on a hybridiza-

tion of least squares support vector regression and swarm intelligence. *Applied Computational Intelligence and Soft Computing*, 2014:8, 2014.

[16] K. Deb. Optimal design of a welded beam structure via genetic algorithms. *AIAA Journal*, 29:2013–2015, 1991.

[17] J. S. Arora. *Introduction to Optimum Design*. McGraw-Hill, New York, 1989.

[18] A. D. Belegundu. *A study of mathematical programming methods for structural optimization*. Department of Civil and Environmental Engineering, University of Iowa, Iowa City, Iowa, 1982.

[19] C. A. C. Coello. Use of a self-adaptive penalty approach for engineering optimization problems. *Computers in Industry*, 41:113–127, 2000.