

# Áp dụng thuật toán tiến hóa (EA) trong tự động bố trí cốt thép không va chạm

Nguyễn Anh Thư<sup>1,2\*</sup>, Trịnh Hoàng Anh<sup>1,2</sup>

<sup>1</sup> Bộ môn Thi Công và Quản lý xây dựng, Khoa Kỹ Thuật Xây Dựng, Trường Đại học Bách Khoa Tp.HCM

<sup>2</sup> Đại học Quốc gia Thành phố Hồ Chí Minh

## TỪ KHÓA

Tránh va chạm cốt thép  
Thiết kế tham số  
Tối ưu hóa đa mục tiêu  
Thuật toán tiến hóa  
NSGA\_II

## KEYWORDS

Clash-free rebars  
Parametric design  
Multi-objective optimization  
Evolutionary algorithm  
NSGA\_II

## TÓM TẮT

Việc bố trí cốt thép là một công việc tốn nhiều thời gian và công sức. Đặc biệt với những trường hợp cần phải điều chỉnh vị trí, hình dạng để đảm bảo các thanh thép không xung đột với nhau. Hiện nay, vấn đề vị trí của các thanh thép trong công tác bố trí mà không phát sinh va chạm đang bị bỏ qua từ khâu thiết kế, dẫn đến việc làm lại hoặc sửa chữa sai sót lặp lại nhiều lần. Nghiên cứu này trình bày một cách tiếp cận để tối ưu vị trí của thanh thép bằng cách áp dụng thuật toán tiến hóa trong thiết kế tham số. Trước tiên, khái niệm về sự va chạm của cốt thép được xác định, công thức toán học về sự va chạm được xây dựng để đo lường mức độ nghiêm trọng của va chạm. Một công thức để kiểm soát độ sai lệch của phương án bố trí mới so với phương án tham chiếu cũng được xây dựng. Bước thứ hai, một mô hình tối ưu hóa vị trí dựa trên thuật toán tiến hóa đa mục tiêu NSGA\_II được xây dựng, có sự kiểm soát bởi tiêu chuẩn quốc gia về thiết kế bê tông cốt thép. Ứng dụng quy trình trên vào một trường hợp điển hình giúp tiết kiệm hơn 85% thời gian trong việc tìm kiếm những giải pháp tối ưu so với phương pháp thủ công hiện tại.

## ABSTRACT

Rebar placement is a time-consuming and labor-intensive task. Especially in cases where the position and shape need to be adjusted to ensure the rebar does not conflict with each other. Currently, the issue of rebar positioning to avoid clashes is often overlooked during the design phase, leading to repeated rework or correction. This study presents an approach to optimize the position of rebar by applying an evolutionary algorithm in parametric design. First, the concept of rebar collision is defined, and a mathematical formula for collision is developed to measure the severity of collision. A formula to control the deviation of the new layout compared to the reference layout is also constructed. Second, an optimization model based on the NSGA\_II multi-objective evolutionary algorithm is built, controlled by national standards code for reinforced concrete design. Applying this process to a case study saves more than 85% of the time in finding optimal solutions compared to the current manual method.

## 1. Giới thiệu

Công trình xây dựng là sự kết hợp của rất nhiều thành phần khác nhau như kiến trúc, kết cấu, MEP, PCCC, hạ tầng...do đó, nhiệm vụ rất quan trọng của người thiết kế là phối hợp, điều phối về bố cục, kích thước của các thành phần trên để hạn chế tối đa xung đột giữa các bên. Xung đột được xác định và xử lý càng sớm thì chi phí phát sinh sẽ càng ít. Theo một thống kê, hơn 80 % những thất bại trong dự án xây dựng bắt nguồn từ lỗi thiết kế [2] và 46 % chi phí phát sinh xảy ra ở giai đoạn thiết kế [3]. Do đó việc phối hợp từ giai đoạn thiết kế và quản lý xung đột là rất quan trọng đối với thành công của một dự án.

Cốt thép là thành phần rất quan trọng trong kết cấu bê tông cốt thép (BTCT). Việc thiết kế cốt thép cần đảm bảo an toàn, đúng tiêu chuẩn, có thể thi công, tiết kiệm. Hiện nay, việc thiết kế cốt thép được

tính toán thủ công hoặc bán tự động bằng các phần mềm như CSI ETABS và Autodesk Robot Structural Analysis (RSA). Tính toán thủ công mất nhiều thời gian và đôi khi dẫn đến sai sót, thiết kế quá thừa hoặc thiết kế chưa đảm bảo [4]. Tính toán bằng phần mềm giúp quá trình thiết kế nhanh hơn nhưng vẫn cần nỗ lực của con người để bố trí cốt thép [4, 5]. Trong giai đoạn thiết kế chi tiết, bố trí cốt thép là công việc tốn nhiều thời gian và công sức. Kết quả của việc bố trí cốt thép có thể bị ảnh hưởng từ va chạm cốt thép (xung đột cứng) hoặc tắc nghẽn cốt thép (xung đột mềm) [6].

Va chạm hay tắc nghẽn cốt thép thường xảy ra tại các nút giao của các cấu kiện, nơi phải tuân thủ theo tiêu chuẩn thiết kế để đảm bảo an toàn. Sẽ tốn rất nhiều thời gian và công sức để bố trí cốt thép tránh tất cả các va chạm bằng phương pháp thử-sai thủ công hoặc ngay cả được sự trợ giúp của các phần mềm hiện có. Hiện nay mặc dù có một

\*Liên hệ tác giả: nathu@hcmut.edu.vn

Nhận ngày 15/05/2024, sửa xong ngày 03/06/2024, chấp nhận đăng ngày 10/06/2024

Link DOI: <https://doi.org/10.54772/jomc.03.2024.727>

số chương trình giúp phát hiện xung đột như Autodesk Navisworks Manage, Solibri Model Checker có thể phát hiện và trực quan hóa các thành phần xung đột [7] nhưng chúng chủ yếu tập trung vào việc phát hiện xung đột của các cấu kiện và MEP. Những phần mềm trên chưa thể tự động tránh các va chạm cốt thép hoặc đưa ra các giải pháp để bố trí lại cốt thép không có va chạm.

Việc tiếp cận vấn đề tự động bố trí cốt thép không va chạm trong khung BTCT ngay từ giai đoạn thiết kế sẽ góp phần làm giảm vấn đề sửa chữa, làm lại tại công trường; dẫn đến việc giảm kiểm tra lại hay thậm chí là thiết kế lại - một trong những nguyên nhân lớn dẫn đến chậm trễ tiến độ và tăng chi phí xây dựng. Bài báo này thực hiện một nghiên cứu áp dụng thuật toán tiến hóa trong việc tự động tìm kiếm vị trí tối ưu để bố trí cốt thép không va chạm.

## 2. Tổng quan

### 2.1 Va chạm cốt thép

Theo Li và cộng sự [1] một cách tổng quát, va chạm của thép cột và dầm được phân thành 3 loại: (a) Va chạm giữa các thanh thép dầm theo các phương khác nhau, (b) Va chạm giữa các thanh thép dầm trên cùng một phương với nhau, (c) Va chạm giữa thép cột và dầm như trong Hình 1. Đối với trường hợp thứ nhất, cách hiệu quả và phổ biến được thực hiện tại công trường là di chuyển các thanh thép của một dầm lên hoặc xuống so với dầm còn lại. Khi di chuyển vị trí thanh thép cần đảm bảo về điều kiện bố trí thép thỏa tiêu chuẩn về khoảng cách, lớp bê tông bảo vệ và vai trò chính – phụ của dầm. Trong khi đối với trường hợp thứ ba việc di chuyển các thanh thép dầm hoặc cột sang phải hoặc trái để tránh va chạm có thể vi phạm khoảng cách tối thiểu trong bố trí thép, đôi khi không tồn tại phương án bố trí thép không va chạm nếu không thay đổi hình dạng thanh thép hoặc kích thước cấu kiện.

Các va chạm cốt thép hiện nay phát sinh do các bước thiết kế cốt thép chỉ mới dừng lại ở việc lựa chọn đường kính, số lượng thanh thép

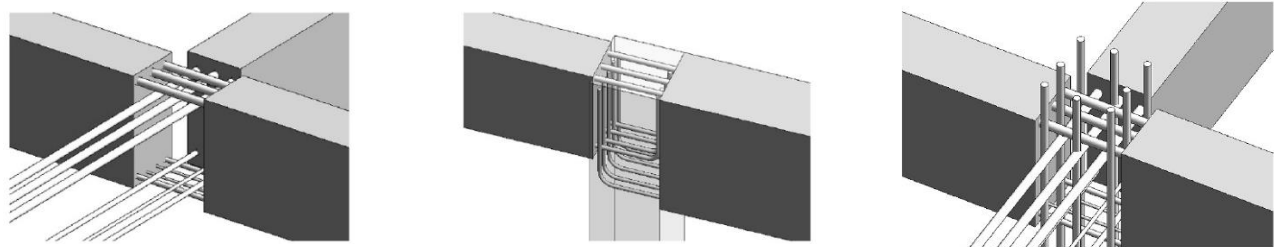
mà chưa chỉ ra được vị trí của mỗi thanh thép trong cấu kiện [1, 8]. Bên cạnh đó, do tính thuận tiện trong việc bố trí thép mà khoảng cách các thanh thép được xem như bố trí cách đều nhau, dẫn đến có sự xung đột về vị trí khi bố trí thép dầm và cột [1, 9].

Đối với cách bố trí cốt thép nhiều lớp, do lớp cốt thép bên trong sẽ luôn thẳng hàng với lớp ngoài cùng và đường kính cốt thép của lớp cốt thép bên trong cũng không vượt quá đường kính của lớp ngoài cùng, vì vậy, lớp ngoài cùng thường là phần tắc nghẽn nhất. Việc giải quyết các va chạm ở lớp ngoài cùng có thể đảm bảo không có va chạm trong toàn bộ cấu kiện.

### 2.2 Tối ưu hóa trong bố trí cốt thép

Tối ưu hóa trong cốt thép là một bài toán đa thức không xác định (Nondeterministic Polynomial NP-hard) bởi vì độ phức tạp về không gian và thời gian tính toán [10]. Các thuật toán tiến hóa (Evolutionary Algorithms, EA) đã được chứng minh là có hiệu quả để giải các vấn đề NP-hard như Thuật toán di truyền (GA) [11], Tối ưu hóa đàn kiến (ACO) [12], Tối ưu hóa đàn hạt (PSO) [13]. Với khả năng xác định các giải pháp tối ưu trong bài toán không gian lớn, đa mục tiêu, phức tạp, các thuật toán EA được xem như một công cụ hứa hẹn để giải quyết các bài toán về bố trí, quy hoạch trong các công trình xây dựng.

Tối ưu hóa di truyền là một kỹ thuật tối ưu hóa sử dụng Thuật toán di truyền (GA). GA thường hoạt động trên các giải pháp được mã hóa dưới dạng biểu diễn kiểu gen (gọi là nhiễm sắc thể) được biểu diễn từ các giá trị dữ liệu đầu vào ban đầu. GA bắt đầu bằng một tập hợp các giải pháp ban đầu (quần thể khởi tạo) được tạo ngẫu nhiên trong không gian tìm kiếm. Sau các hoạt động lựa chọn, trao đổi chéo, đột biến, một thế hệ mới được tạo ra. Quá trình tìm kiếm tiếp tục cho đến khi đáp ứng các tiêu chí dừng, ví dụ: khi đạt được số lượng thế hệ tối đa hoặc việc cải thiện các giải pháp tối ưu hóa theo thế hệ là không đáng kể [14].



(a) Va chạm giữa các thanh thép dầm theo các phương khác nhau

(b) Va chạm giữa các thanh thép dầm trên cùng một phương với nhau

(c) Va chạm giữa thép cột và dầm

Hình 1. Các loại va chạm điển hình [1].

### 2.3 Thiết kế tham số

Thiết kế tham số (Parametric design) là một phương pháp thiết kế sử dụng các thông số và quy tắc để định hình các yếu tố của một thiết kế. Các thông số này có thể là bất kỳ đặc điểm nào của thiết kế,

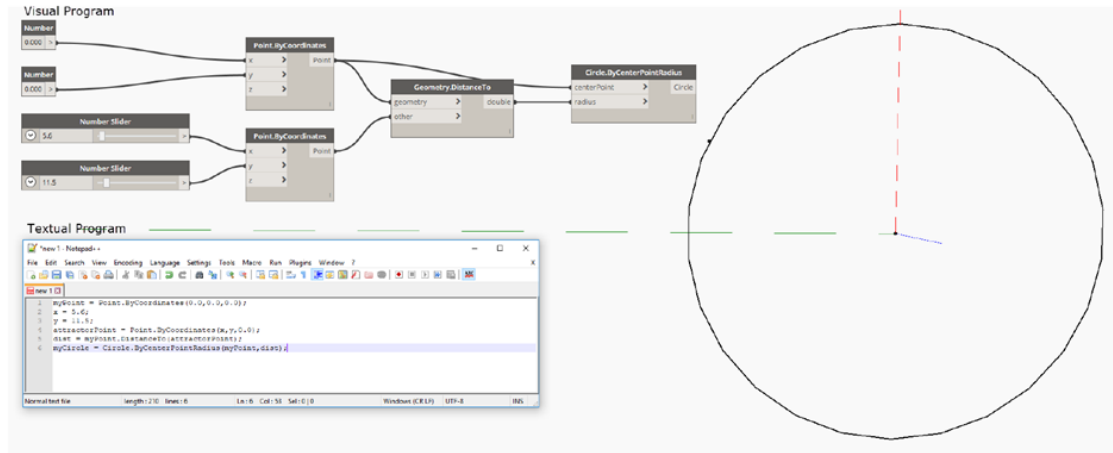
chẳng hạn như kích thước, hình dạng, hoặc các thuộc tính khác, và chúng có thể được thay đổi để tự động tạo ra các biến thể khác nhau của thiết kế [15]. Bằng cách thay đổi các tham số này thông qua các ứng dụng lập trình trực quan, người thiết kế có thể nhanh chóng xem xét và tối ưu hóa nhiều phương án thiết kế khác nhau. Hiện nay, có

hiều ứng dụng lập trình trực quan được sử dụng trong thiết kế tham số, như Dynamo, Grasshopper, Blender...

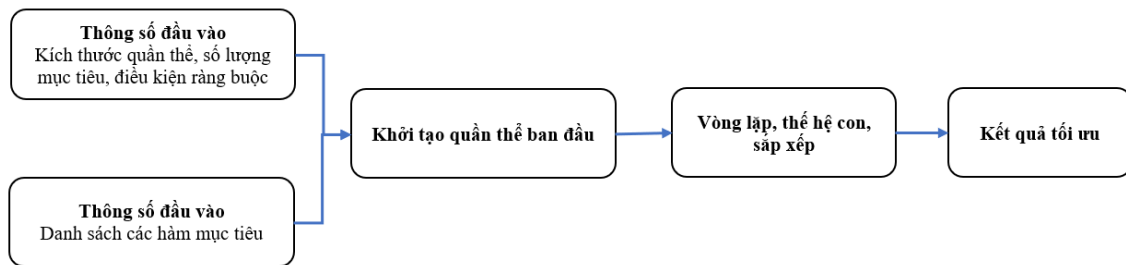
Dynamo là một môi trường lập trình trực quan cho phép người dùng tạo ra các quy trình tự động hóa và điều khiển các yếu tố thiết kế thông qua các tham số và logic hình học. Ứng dụng này có thể hoạt động độc lập hoặc kết hợp với các phần mềm khác như Autodesk Revit, Excel...Được ra mắt vào năm 2012 dưới dạng tiện ích bổ sung cho mô hình thông tin công trình - Building Information Modeling (BIM) dựa trên API Revit và .NET framework. Dynamo cho phép xử lý dữ liệu, hay câu lệnh bằng các thao tác đơn giản thông qua các nút, đại diện các

logic hình học được dùng để miêu tả vấn đề cần giải quyết. Sự khác biệt giữa lập trình trực quan và lập trình văn bản là sự kết nối của các nút được mã hóa so với việc viết các dòng mã.

Optimo được phát triển bởi Asl, Stoupine và các cộng sự vào năm 2015 [17], là một gói công cụ mã nguồn mở dành cho Dynamo, được sử dụng để thực hiện các tối ưu hóa thiết kế tham số. Nguyên lý hoạt động của Optimo dựa trên thuật toán tiến hóa di truyền sắp xếp không trội-II (NSGA-II) và kỹ thuật tối ưu hóa đa mục tiêu để tìm kiếm các giải pháp thiết kế tối ưu. Một cách tổng quát, hoạt động của Optimo có thể được mô tả qua năm bước như được minh họa trong Hình 3.



Hình 2. Ví dụ về lập trình trực quan và lập trình văn bản [16].



Hình 3. Sơ đồ thuật toán Optimo trong Dynamo.

### 2.4 Các nghiên cứu trước đây

Trong nhiều năm qua, đã có những nỗ lực để nghiên cứu về tối ưu hóa thiết kế cốt thép cho các cấu kiện BTCT. Một số nghiên cứu đã áp dụng Mạng thần kinh nhân tạo (ANN) để tối ưu hóa cốt thép trong dầm BTCT bằng phương pháp gia cố đơn và kép để chi phí xây dựng là tối thiểu [18, 19]. Nghiên cứu đưa ra giải pháp tối ưu nhưng các giải pháp chỉ cung cấp tỷ lệ cốt thép cần thiết trên kích thước cấu kiện mà không cung cấp thêm thông tin chi tiết. Ahmadkhanlou và Adeli [20] đã phát triển mô hình động học thần kinh để giảm chi phí xây dựng cho tấm sàn một chiều. Mô hình đã cung cấp một thiết kế cốt thép ít chi phí hơn so với các tấm BTCT trước đây.

GA được phát triển vào khoảng năm 1960 – 1970 bởi Holland và đồng nghiệp [21]. Kể từ đó, các biến thể của GA được phát triển và

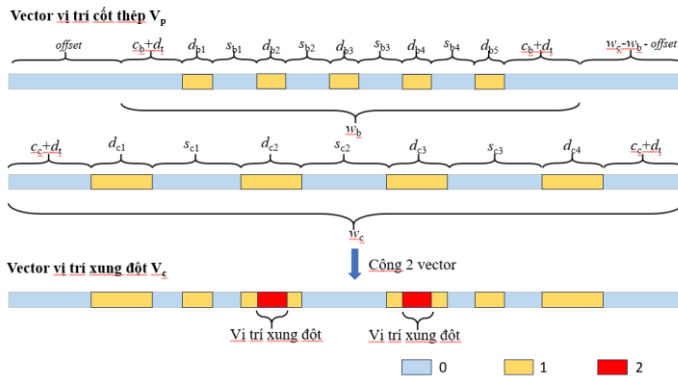
nghiên cứu cho các vấn đề về tối ưu [22]. Điểm nổi bật của GA là hiệu quả cao trong việc tìm kiếm các giải pháp gần tối ưu. Bằng các hàm phạt, GA có thể được sử dụng để giải quyết các ràng buộc vấn đề tối ưu hóa, khi mà sử dụng thuật toán đàn kiến (ACO) trong những vấn đề này thì thường gặp phải các sai lệch (bias) [23]. Hơn nữa ở giai đoạn đầu thuật toán ACO có tốc độ hội tụ tương đối chậm do đặc tính thiếu pheromone giai đoạn đầu. Nếu so sánh GA với thuật toán đàn hạt (PSO) thì GA có khả năng tìm kiếm giải pháp mới trong toàn bộ miền giải pháp trong khi PSO phụ thuộc vào các giải pháp trước đó, mang tính định hướng hơn [24]. Nhìn chung, PSO thường được sử dụng để giải các vấn đề liên tục. Do tính ưu việt của GA so với các thuật toán tiến hóa nên nó thường được dùng để giải quyết các vấn đề về tối ưu thiết kế thép [4].

Leps và Sejnoha [11] đã sử dụng thuật toán di truyền (GA) để tối ưu hóa cốt thép trong dầm nhằm giảm chi phí xây dựng, nhưng nghiên cứu vẫn luôn chia dầm BTCT thành ba phần để tính toán, bất kể điều kiện về tải trọng, dẫn đến kết quả bị hạn chế trong việc đạt tối ưu tổng thể. Sahab và cộng sự (2005) [25] cũng sử dụng GA để tối ưu hóa cốt thép và giảm thiểu chi phí cho sàn phẳng. Mangal (2018) [4] đã áp dụng thuật toán lai GA ba giai đoạn (hGA) để tự động tối ưu hóa cốt thép chịu nén, chịu kéo và chịu cắt. Tuy nhiên nghiên cứu chỉ mới áp dụng cho dầm đơn. Mangal (2021) đã cải tiến GA hai giai đoạn để nghiên cứu về thiết kế cốt thép không va chạm với chi phí vật liệu tối thiểu. Kết quả nghiên cứu cho ra tỷ lệ cốt thép tối ưu được sử dụng mà không đề cập đến vị trí các thanh thép.

Ở một hướng tiếp cận khác, các nghiên cứu của Liu và cộng sự [5, 6, 26] xem xét vấn đề bố trí cốt thép không va chạm như là một bài toán quy hoạch đường đi. Các nghiên cứu xem xét các thanh cốt thép như là những nhân tố thông minh, các thuật toán được áp dụng sẽ điều hướng di chuyển của các nhân tố đi đến đích và tránh các xung đột trên đường đi. Ưu điểm của việc tiếp cận theo phương pháp này là thời gian ra kết quả nhanh; áp dụng được nhiều thuật toán để cải thiện hiệu suất. Tuy nhiên, các nghiên cứu trên chỉ áp dụng cho cốt thép dầm và đường đi của các thanh cốt thép làm giảm chiều cao làm việc của cấu kiện.

### 3. Phương pháp nghiên cứu

#### 3.1 Xây dựng thuật toán tối ưu bố trí cốt thép không va chạm



Hình 4. Ví dụ về phương pháp xác định va chạm.

Li và cộng sự [1] đã miêu tả một phương pháp số để nhận diện va chạm giữa các thanh thép từ dầm và cột như được mô tả trong Hình 4. Theo đó, một cách tổng quát, với  $n$  thanh thép được bố trí trong dầm hoặc cột, các vector khoảng cách  $V_{sb} = [s_{b1}, s_{b2}, \dots, s_{bn-1}]^T$  (1),  $V_{sc} = [s_{c1}, s_{c2}, \dots, s_{cn-1}]^T$  (2) dùng để xác định khoảng cách của từng thanh thép. Trong đó,  $s_i$  là khoảng cách thứ  $i^{th}$  của thanh thép trong dầm hoặc cột. Một vector dùng để xác định cách bố trí thép trong dầm hoặc cột có dạng:

$$V_{pb} = [offset, c_b, d_t, D_{b1}, s_{b1}, D_{b2}, \dots, s_{bn-1}, D_{bn}, d_t, c_b, |w_c - w_b| - offset] \quad (3)$$

$$V_{pc} = [offset, c_c, d_t, D_{c1}, s_{c1}, D_{c2}, \dots, s_{cn-1}, D_{cn}, d_t, c_c, |w_c - w_b| - offset] \quad (4)$$

trong đó: (đơn vị mm)

$c_b, c_c$ : lần lượt là chiều dày lớp bê tông bảo vệ của dầm và cột

$d_t$ : đường kính cốt đai (giả thiết là như nhau tại dầm và cột)

$D_{bi}, D_{ci}$ : lần lượt là đường kính thanh thép thứ  $i$  của dầm và cột

$s_{bi}, s_{ci}$ : lần lượt là khoảng cách thứ  $i$  của thanh thép trong dầm và cột

$w_b, w_c$ : lần lượt là chiều rộng của dầm và cột có phương tương ứng đang xét

$offset$ : độ chênh lệch giữa mặt ngoài của dầm và cột.

#### 3.1.1 Xác định chỉ số va chạm $I_{clash}$

Trong nghiên cứu này, khái niệm chỉ số va chạm  $I_{clash}$  được dùng để đánh giá mức độ va chạm giữa các phần tử là các thanh thép trong cách bố trí được kiểm tra. Chỉ số này giúp đánh giá mức độ nghiêm trọng của va chạm, từ đó hướng tới việc tìm ra các giải pháp tối ưu để giảm thiểu hoặc loại bỏ va chạm. Chỉ số va chạm  $I_{clash}$  được mô tả theo công thức sau:

$$I_{clash} = \frac{n_{clash}}{w_c} \quad (5)$$

trong đó:

$n_{clash}$ : số phần tử va chạm tại vị trí đang xét

$w_c$ : chiều rộng của cột có phương tương ứng đang xét.

Để xác định số phần tử va chạm  $n_{clash}$ , tuần tự thực hiện các bước sau:

**Bước 1:** Từ (3) và (4) lần lượt tạo các vector  $V_{pb}, V_{pc}$  có kích thước  $(w_c * 1)$  với các giá trị "0" và "1". Trong đó, "1" đại diện cho vị trí bị chiếm với các thanh thép.

**Bước 2:** Tạo một vector va chạm  $V_{clashes}$  bằng cách thực hiện phép cộng giữa hai vector  $V_{pb}, V_{pc}$ . Các vị trí có giá trị lớn hơn hoặc bằng "2" trong  $V_{clashes}$  đại diện cho các vị trí va chạm.

**Bước 3:** Từ vector  $V_{clashes}$ , tính số lượng các phần tử lớn hơn hoặc bằng 2, đó là số phần tử va chạm  $n_{clash}$  cần tìm. Sau đó, tính toán chỉ số va chạm  $I_{clash}$  theo công thức (5).

#### 3.1.2 Xác định độ lệch va chạm $\delta_{clash}$

Từ thực tế thi công lắp đặt cốt thép ở công trường, người công nhân khi gặp phải va chạm trong lúc bố trí thép, xu hướng tự nhiên là di dời các thanh thép một khoảng nhỏ đủ để giải quyết vấn đề va chạm. Xuất phát từ thực tiễn, khái niệm về độ lệch va chạm  $\delta_{clash}$  được xây dựng, dùng để đo độ lệch giữa vị trí các thanh thép mới so với vị trí tham chiếu ban đầu. Độ lệch va chạm  $\delta_{clash}$  được tính toán theo công thức sau:

$$\delta_{clash}(x) = \sqrt{\sum_{i=1}^n (x_i - x_{0i})^2} \quad (6)$$

trong đó:

$x = [V_{sb}, V_{sc}]$ : là vector khoảng cách của các thanh thép

$x_i$ : là phần tử thứ  $i$  của vector  $x$  đang xét

$x_{0i}$ : là phần tử thứ  $i$  của vector tham chiếu

$n$ : số phần tử của vector  $x$ .

### 3.2 Hàm mục tiêu

Với chỉ số va chạm  $I_{clash}$  được trình bày ở phần trên, lúc này mục tiêu của việc xác định và tránh va chạm cốt thép được chuyển đổi thành việc xác định và giảm chỉ số va chạm  $I_{clash}$  với các biến  $x = [V_{sb}, V_{sc}]$ . Các giá trị tham chiếu đầu vào của  $x$  được xác định dựa trên sự thuận tiện trong công tác bố trí thép, các thanh thép được xem như bố trí cách đều nhau  $x_0$ .

Bên cạnh đó, việc bố trí cốt thép không va chạm có thể có nhiều phương án. Nói cách khác, tồn tại nhiều giá trị  $x = [V_{sb}, V_{sc}]$  mà tại đó chỉ số va chạm  $I_{clash} = 0$ . Do đó, giá trị độ lệch va chạm  $\delta_{clash}$  được thêm vào để đánh giá ưu thế của giá trị  $x_k$  so với các giá trị  $x$  còn lại. Với các giá trị  $\delta_{clash}$  càng nhỏ cho thấy việc di chuyển hay điều chỉnh các thanh thép trong bố cục đang xét ít tốn công sức hơn. Từ đó đạt được hiệu quả tối ưu trong việc lựa chọn bố cục không va chạm một cách hợp lý nhất.

Do đó, hàm mục tiêu của bài toán được trình bày dưới dạng công thức sau:

$$y = \min F(x) = \min \begin{cases} I_{clash}(x) \\ \delta_{clash}(x) \end{cases} \quad (7)$$

trong đó  $x = [V_{sb}, V_{sc}]$

Trong hai mục tiêu cần tối ưu là  $I_{clash}$  và  $\delta_{clash}$  thì giá trị  $I_{clash}$  có mức độ ưu tiên cao hơn và được sử dụng để đánh giá cá thể có độ thích nghi tốt hơn trong quần thể được lựa chọn. Giá trị  $I_{clash}$  càng nhỏ cho thấy mức độ va chạm giữa các thanh cốt thép càng thấp, và giá trị tối ưu của  $I_{clash}$  có thể đạt đến "0" có nghĩa là cách bố trí cốt thép thỏa mãn mục tiêu không va chạm. Bên cạnh đó, giá trị  $\delta_{clash}$  càng nhỏ cho thấy cách bố trí càng ít thay đổi so với cách bố trí đều. Do đó, với nhiều cá thể đạt được giá trị  $I_{clash}$  giống nhau, thì cá thể nào có giá trị  $\delta_{clash}$  nhỏ hơn sẽ ít tốn chi phí hơn và là cá thể tối ưu hơn.

### 3.3 Ràng buộc của bài toán

Hàm mục tiêu (7) được tính toán dựa trên các ràng buộc sau:

$$30 \leq s_{bi} \leq 400 \text{ Theo TCVN 5574: 2018} \quad (8)$$

$$50 \leq s_{ci} \leq 400 \text{ Theo TCVN 5574: 2018} \quad (9)$$

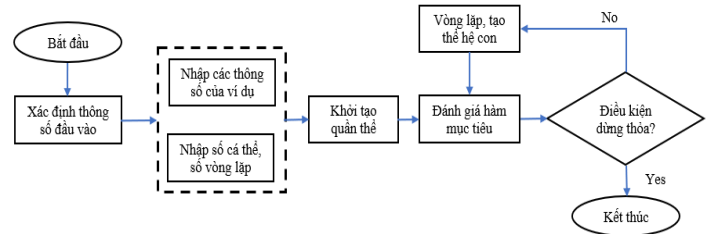
$$\sum_{i=1}^{n-1} s_{bi} \leq w_b - (2 * (c_b + d_t) + \sum D_b) \quad (10)$$

$$\sum_{i=1}^{n-1} s_{ci} \leq w_c - (2 * (c_c + d_t) + \sum D_c) \quad (11)$$

Công thức (8), (9) được tham khảo từ TCVN 5574:2018 về Thiết kế kết cấu bê tông và bê tông cốt thép, ràng buộc khoảng cách tối thiểu và tối đa giữa các thanh cốt thép chịu lực trong cùng một lớp thép. Công thức (10) và (11) lần lượt là tổng khoảng cách giữa các thanh cốt thép đối với dầm và cột. Vì khởi tạo quần thể và tối ưu hóa là những quá trình ngẫu nhiên, do đó các cá thể được tạo ra phải nằm trong vùng tìm kiếm, đồng thời tổng kích thước của các phần tử trong cá thể cũng không được lớn hơn chiều rộng của dầm và cột. Khoảng cách được hiểu là khoảng cách thông thủy giữa hai thanh cốt thép liền nhau.

## 4. Áp dụng quy trình vào trường hợp điển hình

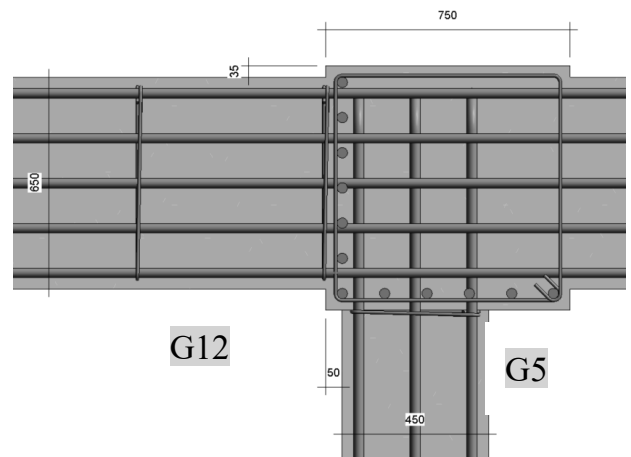
Mục tiêu của việc áp dụng quy trình vào trường hợp điển hình nhằm đánh giá khả năng tìm kiếm giải pháp của quy trình đề xuất, đồng thời xác định thời gian tìm kiếm giải pháp tối ưu. Sơ đồ mô phỏng việc áp dụng quy trình vào trường hợp điển hình được trình bày trong Hình 5.



Hình 5. Sơ đồ mô phỏng quy trình đề xuất.

### 4.1 Xác định các thông số đầu vào

Dầm G5 có chiều rộng 450 mm với 3 thanh cốt thép đường kính 32 mm ở lớp trên. Dầm G12 có chiều rộng bằng nhau 650 mm với 5 thanh thép đường kính 28 mm. Cột C9 có chiều rộng 750 mm với 6 thanh cốt thép theo phương X và 7 thanh cốt thép theo phương Y, đường kính 32 mm. Độ lệch giữa mặt ngoài của dầm và cột (offset) lần lượt là 50mm và 35 mm như Error! Reference source not found.. Để đơn giản trong tính toán, lớp bê tông bảo vệ được lấy bằng 25 mm và đường kính thép đai của cột là 10mm. Các giá trị điều kiện biên dưới và điều kiện biên trên của các biến lần lượt là 50 mm và 150 mm. Tổng số biến cần tối ưu là 7 biến theo phương X và 10 biến theo phương Y. Quần thể ban đầu được khởi tạo với 200 cá thể và số vòng lặp cho dầm G5 là 200 và dầm G12 là 50.



Hình 6. Mô phỏng trường hợp điển hình.

### 4.2 Nhập các thông số đầu vào

**Bảng 1.** Các thông số đầu vào.

Thông số đầu vào		G5	G12	Thông số đầu vào		G5	G12
1	Bề rộng cột wc:	750	750	8	Số thanh thép dầm:	3	5
2	Bề rộng dầm wb:	450	650	9	Offset:	50	35
3	Lớp bê tông bảo vệ:	25	25	10	Ràng buộc biên dưới:	50	50
4	Đường kính thép đai:	10	10	11	Ràng buộc biên trên:	150	150
5	Đường kính thép cột:	32	32	12	Kích thước quần thể:	200	200
6	Đường kính thép dầm:	32	28	13	Số hàm mục tiêu:	2	2
7	Số thanh thép cột:	6	7	14	Số vòng lặp:	50	50

**4.3 Kết quả quy trình tối ưu vị trí**

Sau khi kết thúc quy trình tối ưu, dầm G5 tại vòng lặp 50 trả về 17 cá thể có giá trị  $I_{clash}$  đạt cực tiểu bằng 0. Trong đó cá thể có độ lệch  $\delta_{clash}$  thấp nhất là 49.13247. Dầm G12 trả về 5 cá thể có giá trị  $I_{clash}$  đạt cực tiểu bằng 0, cá thể có độ lệch  $\delta_{clash}$  thấp nhất là 46.23851.

Nghiên cứu cũng tiến hành đối chứng giữa quy trình tối ưu hóa bằng thuật toán được đề xuất với phương pháp bố trí thép thủ công đang được áp dụng. Nghiên cứu mời được 5 cá nhân có trên 5 năm kinh nghiệm trong lĩnh vực tư vấn thiết kế, thi công và quản lý thi công thuộc các công ty xây dựng tại thành phố Hồ Chí Minh. Các cá nhân được mời tham gia khảo sát đã mô phỏng lại trường hợp điển hình bằng cách vẽ, bố trí vị trí các thanh thép thỏa mãn yêu cầu không va chạm cũng như đảm bảo các ràng buộc (8-11), công cụ sử dụng là AutoCad, Autodesk Revit và thời gian thao tác mỗi trường hợp được ghi nhận lại. Bảng 2 so sánh thời gian xác định vị trí thanh thép không va chạm giữa quy trình đề xuất với phương pháp sắp xếp thủ công. So với trung bình thời gian sắp xếp thủ công, thời gian tính toán của quy trình đề xuất có thể giúp tiết kiệm được trên 85 % thời gian tìm kiếm phương án tối ưu. Bên cạnh đó, trong quá trình thực hiện các vòng lặp tính toán, quy trình đề xuất còn cung cấp nhiều hơn một phương án để có thể lựa chọn.

**Bảng 2.** So sánh quy trình đề xuất và phương pháp thủ công.

THĐH	Kích thước cột x dầm	Tổng số thanh thép	Thời gian tính toán (giây)	Thời gian thủ công trung bình (giây)	Thời gian tiết kiệm
Dầm G5	750x450	9	17	168	89,8%
Dầm G12	750x650	12	45	360	87,5%

**5. Kết luận**

Nghiên cứu đề xuất phương pháp giải quyết bài toán va chạm trong bố trí cốt thép bằng thuật toán tiến hóa. Việc ứng dụng quy trình đề xuất trong nghiên cứu vào thực tế giúp giảm thiểu thời gian giải

quyết xung đột tại hiện trường, góp phần làm giảm thiểu lãng phí tài nguyên của xã hội, giảm thiểu việc trì hoãn và công sức lao động do các sửa đổi tại chỗ. Trường hợp điển hình được sử dụng trong nghiên cứu, cũng chứng minh được quy trình giúp tiết kiệm hơn 85 % thời gian trong việc tìm kiếm những giải pháp tối ưu so với phương pháp thủ công hiện tại. Bên cạnh đó, ứng dụng quy trình cũng giúp tìm kiếm và đề xuất nhiều giải pháp bố trí tốt ngay từ giai đoạn thiết kế.

Mặc dù bước đầu nghiên cứu cũng đã cố gắng cung cấp một số kết quả đáng khích lệ, tuy nhiên, quy trình nghiên cứu vẫn tồn tại ba hạn chế lớn. Thứ nhất, phạm vi nghiên cứu chỉ mới tập trung ở các cấu kiện dầm, cột, vách nơi có hàm lượng và kích thước thép lớn. Đối với các cấu kiện sàn, móng, tường, dầm liên tục sẽ được nghiên cứu trong tương lai. Thứ hai, nghiên cứu chưa liên kết mô hình tối ưu với các phần mềm tính toán để tự động hóa việc mô phỏng phương án bố trí thép tối ưu. Thứ ba, thuật toán NSGA\_II sử dụng trong mô hình mặc dù là một thuật toán mạnh mẽ và phổ biến trong tối ưu hóa đa mục tiêu, tuy nhiên vẫn có những hạn chế trong tính toán khi kích thước quần thể quá lớn hoặc số lượng mục tiêu cần xem xét lớn. Do đó, cần nghiên cứu kết hợp NSGA\_II với các thuật toán khác để có thể lưu trữ những giải pháp tối ưu đã có, mục đích làm giảm thời gian và chi phí trong vấn đề tối ưu hóa.

**Lời cảm ơn**

Chúng tôi xin cảm ơn Trường Đại học Bách Khoa, ĐHQG-HCM đã hỗ trợ cho nghiên cứu này.

**Tài liệu tham khảo**

[1]. M. Li, B. C. L. Wong, Y. Liu, C. M. Chan, V. J. L. Gan, and J. C. P. Cheng, "DfMA-oriented design optimization for steel reinforcement using BIM and hybrid metaheuristic algorithms," *Journal of Building Engineering*, vol. 44, 2021, doi: 10.1016/j.job.2021.103310.

[2]. R. Lopez, P. E. Love, D. J. Edwards, and P. R. Davis, "Design error classification, causation, and prevention in construction engineering," *Journal of performance of constructed facilities*, vol. 24, no. 4, pp. 399-408, 2010.

[3]. P. E. Love, D. J. Edwards, J. Smith, and D. H. Walker, "Divergence or congruence? A path model of rework for building and civil engineering projects," *Journal of performance of constructed facilities*, vol. 23, no. 6, pp. 480-488, 2009.

- [4]. M. Mangal and J. C. Cheng, "Automated optimization of steel reinforcement in RC building frames using building information modeling and hybrid genetic algorithm," *Automation in Construction*, vol. 90, pp. 39-57, 2018.
- [5]. J. Liu, P. Liu, L. Feng, W. Wu, D. Li, and Y. F. Chen, "Automated clash resolution for reinforcement steel design in concrete frames via Q-learning and Building Information Modeling," *Automation in Construction*, vol. 112, 2020, doi: 10.1016/j.autcon.2019.103062.
- [6]. J. Liu, S. Li, C. Xu, Z. Wu, N. Ao, and Y. F. Chen, "Automatic and optimal rebar layout in reinforced concrete structure by decomposed optimization algorithms," *Automation in construction*, vol. 126, p. 103655, 2021.
- [7]. L. Wang and F. Leite, "Formalized knowledge representation for spatial conflict coordination of mechanical, electrical and plumbing (MEP) systems in new building projects," *Automation in construction*, vol. 64, pp. 20-26, 2016.
- [8]. M. Mangal, Q. Wang, and J. Cheng, "Automated clash resolution of steel rebar in RC beam-column joints using BIM and GA," in *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, 2017, vol. 34: IAARC Publications.
- [9]. A. M. Jarkas, "The influence of buildability factors on rebar fixing labour productivity of beams," *Construction management and economics*, vol. 28, no. 5, pp. 527-543, 2010.
- [10]. M. Mangal, M. Li, V. J. L. Gan, and J. C. P. Cheng, "Automated clash-free optimization of steel reinforcement in RC frame structures using building information modeling and two-stage genetic algorithm," *Automation in Construction*, vol. 126, 2021, doi: 10.1016/j.autcon.2021.103676.
- [11]. M. Lepš and M. Šejnoha, "New approach to optimization of reinforced concrete beams," *Computers & structures*, vol. 81, no. 18-19, pp. 1957-1966, 2003.
- [12]. M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE transactions on systems, man, and cybernetics, part b (cybernetics)*, vol. 26, no. 1, pp. 29-41, 1996.
- [13]. R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *MHS'95. Proceedings of the sixth international symposium on micro machine and human science*, 1995: Ieee, pp. 39-43.
- [14]. A. E. Eiben, P.-E. Raue, and Z. Ruttkey, "Genetic algorithms with multi-parent recombination," in *International conference on parallel problem solving from nature*, 1994: Springer, pp. 78-87.
- [15]. C. M. Eastman, *BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors*. John Wiley & Sons, 2011.
- [16]. J. R. Olesen, "Intelligent optimization of steel structures in the early design phase through Visual Programming," Master, Technical University of Denmark, 2018.
- [17]. M. R. Asl, A. Stoupine, S. Zarrinmehr, and W. Yan, "Optimo: A BIM-based multi-objective optimization tool utilizing visual programming for high performance building design," *Proceedings of eCAADe 2015*, pp. 673-682, 2015.
- [18]. M. N. Hadi, "Neural networks applications in concrete structures," *Computers & structures*, vol. 81, no. 6, pp. 373-381, 2003.
- [19]. B. Saini, V. Sehgal, and M. Gambhir, "Genetically optimized artificial neural network based optimum design of singly and doubly reinforced concrete beams," 2006.
- [20]. F. Ahmadkhanlou and H. Adeli, "Optimum cost design of reinforced concrete slabs using neural dynamics model," *Engineering Applications of Artificial Intelligence*, vol. 18, no. 1, pp. 65-72, 2005.
- [21]. L. B. Booker, D. E. Goldberg, and J. H. Holland, "Classifier systems and genetic algorithms," *Artificial intelligence*, vol. 40, no. 1-3, pp. 235-282, 1989.
- [22]. A. Konak, D. W. Coit, and A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial," *Reliability engineering & system safety*, vol. 91, no. 9, pp. 992-1007, 2006.
- [23]. M. Dorigo and C. Blum, "Ant colony optimization theory: A survey," *Theoretical computer science*, vol. 344, no. 2-3, pp. 243-278, 2005.
- [24]. R. C. Eberhart and Y. Shi, "Comparison between genetic algorithms and particle swarm optimization," in *International conference on evolutionary programming*, 1998: Springer, pp. 611-616.
- [25]. M. Sahab, A. Ashour, and V. Toropov, "A hybrid genetic algorithm for reinforced concrete flat slab buildings," *Computers & structures*, vol. 83, no. 8-9, pp. 551-559, 2005.
- [26]. J. Liu, C. Xu, Z. Wu, and Y. F. Chen, "Intelligent rebar layout in RC building frames using artificial potential field," *Automation in Construction*, vol. 114, 2020, doi: 10.1016/j.autcon.2020.103172.