# THUẬT TOÁN NHÁNH CẬN GIẢI MỘT SỐ BÀI TOÁN TỐI ƯU LIÊN QUAN ĐẾN CHU TRÌNH HAMILTON DỰA TRÊN BÀI TOÁN TSP

**Ngô Thúy Ngân**
*Trường Đại học Thủ đô Hà Nội*

***Tóm tắt:*** *Bài toán Người du lịch (TSP) là một bài toán tối ưu tổ hợp quan trọng, yêu cầu tìm chu trình ngắn nhất đi qua tất cả các đỉnh đúng một lần. Do thuộc nhóm NP-khó, TSP đòi hỏi các phương pháp giải mạnh để tìm lời giải tối ưu. Thuật toán nhánh cận (Branch and Bound – BnB) là một cách tiếp cận hiệu quả cho bài toán này. BnB chia nhỏ không gian tìm kiếm, tính cận dưới cho từng nhánh và loại bỏ các nhánh không khả thi, giúp giảm số phép thử so với quay lui truyền thống. Bài viết trình bày chi tiết cách áp dụng BnB vào TSP, bao gồm mô tả thuật toán, ví dụ minh họa và đánh giá thực nghiệm. Kết quả cho thấy BnB hoạt động tốt với bài toán có quy mô nhỏ và vừa, nhưng thời gian xử lý tăng nhanh khi số lượng đỉnh lớn. Tác giả đề xuất kết hợp BnB với các phương pháp heuristic và tối ưu lưu trữ để nâng cao hiệu quả khi giải các bài toán TSP lớn trong thực tế.*

***Từ khóa:*** *Bài toán người du lịch; bài toán tối ưu tổ hợp; chu trình Hamilton; đường Hamilton; NP - đầy đủ; NP - khó; thuật toán nhánh cận.*

Nhận bài ngày 10.04.2025; gửi phản biện, chỉnh sửa, duyệt đăng ngày 30.05.2025
Liên hệ tác giả: Ngô Thúy Ngân; email: ntngan@daihocthudo.edu.vn

## 1. ĐẶT VẤN ĐỀ

Trong lĩnh vực tối ưu tổ hợp, Chu trình Hamilton và bài toán Người du lịch (Traveling Salesman Problem - TSP) là hai bài toán quan trọng có nhiều ứng dụng thực tế trong khoa học máy tính, logistics, vận tải và lập lịch. Những bài toán này đòi hỏi việc tìm kiếm các hành trình tối ưu trong đồ thị có trọng số, đảm bảo các ràng buộc về lộ trình.

Chu trình Hamilton là một đường đi qua tất cả các đỉnh trong đồ thị đúng một lần và quay lại điểm xuất phát. Khi đồ thị có trọng số, bài toán tối ưu liên quan đến việc tìm chu trình Hamilton có tổng trọng số nhỏ nhất. Một trường hợp đặc biệt của bài toán này là bài toán Người du lịch, trong đó đồ thị được giả định là đầy đủ và trọng số thể hiện chi phí hoặc khoảng cách giữa các đỉnh.

Việc giải quyết bài toán TSP có độ phức tạp cao do số lượng hành trình tăng theo giai thừa của số đỉnh. Nhiều thuật toán đã được đề xuất để tìm lời giải tối ưu hoặc xấp xỉ, trong đó thuật toán nhánh cận (Branch and Bound - BnB) là một phương pháp mạnh mẽ giúp cắt giảm không gian tìm kiếm và cải thiện hiệu suất tính toán.

Bài báo này trình bày chi tiết phương pháp nhánh cận trong việc giải quyết bài toán TSP và ứng dụng vào việc tìm chu trình Hamilton tối ưu. Tác giả sẽ phân tích lý thuyết, mô tả thuật toán, minh họa bằng ví dụ cụ thể, và đánh giá hiệu quả của phương pháp này trong thực tế.

## 2. NỘI DUNG

### 2.1. Bài toán chu trình Halmiton

Chu trình Hamilton là một đường đi trong đồ thị vô hướng hoặc có hướng sao cho đi qua mỗi đỉnh đúng một lần và quay lại đỉnh xuất phát. Chu trình này là một tập hợp các cạnh tạo thành một chu trình đơn và bao gồm tất cả các đỉnh trong đồ thị.

Bài toán Chu trình Hamilton yêu cầu xác định xem trong một đồ thị có tồn tại chu trình Hamilton hay không và nếu có, tìm chu trình Hamilton có tổng trọng số nhỏ nhất [1].

Điều kiện tồn tại:

Một đồ thị có thể không có chu trình Hamilton. Một số tiêu chí quan trọng giúp xác định sự tồn tại của chu trình Hamilton bao gồm:

- Định lý Dirac: Nếu một đồ thị vô hướng G có n đỉnh ($n \geq 3$) và mỗi đỉnh có bậc ít nhất n/2 thì G có ít nhất một chu trình Hamilton.
- Định lý Ore: Nếu với mọi cặp đỉnh không kề nhau u, v trong đồ thị G, tổng bậc của hai đỉnh này ít nhất là n ($\deg(u) + \deg(v) \geq n$), thì đồ thị G có chu trình Hamilton.
- Đồ thị đầy đủ: Một đồ thị đầy đủ với n đỉnh luôn có ít nhất một chu trình Hamilton.

Ví dụ:

Giả sử có một đồ thị với 5 đỉnh và các cạnh như sau:

V = {A, B, C, D, E}

E = {(A, B), (A, C), (A, D), (B, C), (B, D), (B, E), (C, D), (C, E), (D, E)}

Một chu trình Hamilton trong đồ thị này có thể là: A → B → E → C → D → A.

Nếu đồ thị là có trọng số, bài toán sẽ tìm chu trình Hamilton có tổng trọng số nhỏ nhất.

### 2.2. Bài toán người du lịch và thuật toán nhánh - cận

Bài toán Người du lịch là một trường hợp cụ thể của bài toán Chu trình Hamilton trong đó đồ thị G là đồ thị đầy đủ (mọi cặp đỉnh đều có cạnh nối), với mục tiêu tìm một chu trình Hamilton có tổng trọng số nhỏ nhất. Cụ thể:

- Xác định một hoán vị của n đỉnh ($p_1, p_2, ..., p_n$) sao cho tổng trọng số là nhỏ nhất.
- Ràng buộc: mỗi đỉnh chỉ được thăm đúng một lần.

Ví dụ:

Giả sử một nhân viên bán hàng cần đi qua 4 thành phố {A, B, C, D} và quay lại thành phố xuất phát. Khoảng cách giữa các thành phố được cho bởi ma trận trọng số:

| | **A** | **B** | **C** | **D** |
|---|---|---|---|---|
| A | ∞ | 10 | 15 | 20 |
| B | 10 | ∞ | 35 | 25 |
| C | 15 | 35 | ∞ | 30 |
| D | 20 | 25 | 30 | ∞ |

Mục tiêu là tìm chu trình tối ưu có tổng trọng số nhỏ nhất.

Giải quyết bài toán bằng thuật toán nhánh cận:

1. Bước khởi tạo: Xác định cận dưới ban đầu bằng cách lấy tổng trọng số hai cạnh nhỏ nhất từ mỗi thành phố, sau đó chia đôi.

2. Phân nhánh: Bắt đầu từ thành phố A, thử tất cả các khả năng đi đến các thành phố còn lại.
3. Tính toán cận dưới: Sau mỗi lần chọn một thành phố, cập nhật cận dưới để đánh giá xem có nên tiếp tục với nhánh đó hay không.
4. Tối ưu hóa: Nếu tìm được chu trình có tổng trọng số nhỏ hơn giá trị tối ưu hiện tại, cập nhật nghiệm tốt nhất.
5. Tiếp tục lặp cho đến khi không còn nhánh nào cần xét.

Ví dụ, nếu chúng ta chọn hành trình: A → B → D → C → A thì tổng trọng số là:

- A → B: 10
- B → D: 25
- D → C: 30
- C → A: 15
- Tổng: 10 + 25 + 30 + 15 = 80

So sánh với các hành trình khác để tìm hành trình có trọng số nhỏ nhất [2].

### 2.3. Thuật toán nhánh cận

Thuật toán nhánh cận hoạt động dựa trên việc cắt giảm không gian tìm kiếm bằng cách ước lượng cận dưới và phân nhánh theo các lựa chọn tốt nhất. Các bước chính của thuật toán như sau:

1. **Khởi tạo cận dưới**: Xác định cận dưới ban đầu bằng cách tính tổng trọng số hai cạnh nhỏ nhất từ mỗi đỉnh, sau đó chia đôi để có giá trị ước lượng.
2. **Phân nhánh**: Bắt đầu từ một thành phố (ví dụ A), tạo ra các nhánh bằng cách chọn thành phố tiếp theo có trọng số nhỏ nhất.
3. **Tính toán cận dưới cho mỗi nhánh**:
   - Nếu cận dưới của một nhánh lớn hơn nghiệm tối ưu tạm thời, loại bỏ nhánh đó để giảm không gian tìm kiếm.
   - Nếu nhánh đạt đến một nghiệm đầy đủ (đã đi qua tất cả các thành phố và quay lại điểm xuất phát), kiểm tra xem tổng trọng số có nhỏ hơn nghiệm tốt nhất hiện tại không.
4. **Cập nhật nghiệm tốt nhất**: Nếu tìm thấy một chu trình có tổng trọng số nhỏ hơn nghiệm tối ưu hiện tại, cập nhật giá trị tối ưu.
5. **Tiếp tục mở rộng nhánh hợp lệ** cho đến khi không còn nhánh nào cần xét [3].

## 3. ĐÁNH GIÁ

Thuật toán nhánh cận là một phương pháp hiệu quả để giải bài toán Chu trình Hamilton và bài toán Người du lịch, đặc biệt khi số lượng đỉnh không quá lớn. Dưới đây là một số đánh giá về hiệu suất và ứng dụng của thuật toán:

### 3.1. Ưu điểm

- **Tìm lời giải tối ưu**: Không giống như các phương pháp xấp xỉ như thuật toán tham lam hay heuristic, thuật toán nhánh cận đảm bảo tìm được lời giải tối ưu nếu có đủ tài nguyên tính toán.