**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG TP.HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**-----o0o-----**

****

**HỌC PHẦN:**

**TRÍ TUỆ NHÂN TẠO**

**THUẬT GIẢI GTS**

**(GREEDY-TRAVELING SALEMAN)**

**Thành phố Hồ Chí Minh, tháng 10 năm 2024**

**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG TP.HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**-----o0o-----**

****

**THUẬT GIẢI GTS**

**(GREEDY-TRAVELING SALEMAN)**

**Giảng viên hướng dẫn: ThS. Trần Như Ý**

Trưởng nhóm: Huỳnh Thanh Sơn\_2001224190

Thành viên:

1. Trần Thị Yến Nhi\_2001223311

**Thành phố Hồ Chí Minh, tháng 12 năm 2024**

# **MỤC LỤC**

[MỤC LỤC 3](#_Toc178632519)

[PHỤ LỤC BẢNG 4](#_Toc178632520)

[PHỤ LỤC HÌNH ẢNH 4](#_Toc178632521)

[PHÂN CÔNG CÔNG VIỆC 5](#_Toc178632522)

[MỞ ĐẦU 6](#_Toc178632523)

[NỘI DUNG 7](#_Toc178632524)

[**I. GIỚI THIỆU** 7](#_Toc178632525)

[**1. Thuật toán tham lam (Greedy Algorithm)** 7](#_Toc178632526)

[**2. Bài toán người bán hàng Traveling Salesman Problem (TSP)** 7](#_Toc178632527)

[**II. THUẬT TOÁN GREEDY TRAVELING SALESMAN (GTS)** 8](#_Toc178632528)

[**1. Ý tưởng** 8](#_Toc178632529)

[**2. Các bước thực hiện** 9](#_Toc178632530)

[**3. Ví dụ minh họa** 9](#_Toc178632531)

[**4. Ưu và nhược điểm** 13](#_Toc178632532)

[**5. Ứng dụng thực tế** 13](#_Toc178632533)

[**6. So sánh với một số thuật toán khác** 13](#_Toc178632534)

[KẾT LUẬN 14](#_Toc178632535)

[TÀI LIỆU THAM KHẢO 15](#_Toc178632536)

[PHỤ LỤC 16](#_Toc178632537)

# **PHỤ LỤC BẢNG**

[Bảng 1 10](#_Toc178633335)

[Bảng 2 11](#_Toc178633336)

[Bảng 3 14](#_Toc178633337)

# **PHỤ LỤC HÌNH ẢNH**

[Hình 1 11](#_Toc178632148)

# **PHÂN CÔNG CÔNG VIỆC**

|  |  |  |
| --- | --- | --- |
| **Họ và tên** | **Nội dung công việc** | **Mức độ hoàn thành** |
| Huỳnh Thanh Sơn | Code Demo, PowerPoint, Word, Thuyết trình |  |
| Trần Thị Yến Nhi | PowerPoint, Word, Thuyết trình |  |
| Mai Lê Bảo Khang | PowerPoint, Word, Thuyết trình |  |

# **MỞ ĐẦU**

Bài toán người bán hàng (Traveling Salesman Problem - TSP) là một trong những bài toán kinh điển trong lĩnh vực tối ưu hóa và lý thuyết đồ thị. Mục tiêu của bài toán này là tìm ra hành trình ngắn nhất để một người bán hàng ghé thăm một danh sách các thành phố và quay trở lại điểm xuất phát. Tuy nhiên, TSP thuộc bài toán NP-khó, nghĩa là rất khó để tìm được giải pháp tối ưu trong thời gian ngắn với số lượng thành phố lớn.

Để giải quyết bài toán TSP, các nhà khoa học đã phát triển nhiều thuật toán và phương phép tiếp cận khác nhau. Một trong những tiếp cận đơn giản là Thuật toán Greedy Traveling Salesman (GTS) , dựa trên phương pháp tiếp cận tham lam, là một giải pháp gần đúng để giải quyết bài toán người bán hàng. Mặc dù không đảm bảo tìm được lời giải tối ưu nhưng GTS có ưu điểm là đơn giản, dễ thực hiện và tính toán nhanh.

# **NỘI DUNG**

## **I. GIỚI THIỆU**

### **1. Thuật toán tham lam (Greedy Algorithm)**

#### **1.1. Khái niệm**

Thuật toán tham lam (Greedy Algorithm) là một phương pháp tối ưu hóa, trong đó tại mỗi bước, thuật toán chọn giải pháp tốt nhất tại thời điểm đó với hy vọng đạt được giải pháp tối ưu toàn cục. Thuật toán sẽ tiếp tục giải các bài toán con dựa trên lựa chọn này. Tuy nhiên, do chỉ tập trung vào giải pháp tốt ở hiện tại mà không xét lại các lựa chọn trước đó, giải thuật tham lam thường không đảm bảo tìm được giải pháp tối ưu toàn cục.

Thuật toán tuân theo các nguyên tắc sau:

+ Chọn bước đi tốt nhất: Ở mỗi bước, thuật toán sẽ chọn giải pháp tối ưu nhất tại thời điểm hiện tại.

+ Không quay lại: Sau khi đã chọn một bước đi thì thuật toán sẽ không xét lại bước đi đó nữa.

#### **1.2. Ưu điểm và nhược điểm của thuật toán**

**\* Ưu điểm:**

+ Thời gian thực thi nhanh

+ Dễ cài đặt

**\* Nhược điểm:**

+ Không đảm bảo sẽ tìm được giải pháp tốt nhất cho các bài toán phức tạp, thuật toán sẽ cho ra một giải pháp không tối ưu.

### **2. Bài toán người bán hàng Traveling Salesman Problem (TSP)**

#### **2.1. Định nghĩa**

Bài toán người bán hàng (Traveling Salesman Problem – TSP) là một bài toán tối ưu hóa cổ điển trong toán học và khoa học máy tính. Bài toán yêu cầu tìm lộ trình đi qua N thành phố, mỗi thành phố chỉ được ghé thăm một lần và quay trở về điểm xuất phát sao cho tổng chiều dài quãng đường đi là ngắn nhất.

*Giả thuyết*: Cho tập các thành phố và khoảng cách giữa các thành phố

*Kết quả*: Một chu trình tối ưu có tổng chiều dài nhỏ nhất, đi qua các thành phố một lần và trở về thành phố ban đầu.

#### **2.2. Phân loại**

Có hai dạng chính:

+ TSP đối xứng: Khoảng cách từ thành phố A đến thành phố B bằng khoảng cách từ thành phố B đến thành phố A.

+ TSP không đối xứng: Khoảng cách giữa hai thành phố có thể khác nhau tùy vào chiều di chuyển.

#### **2.3. Tính phức tạp của bài toán**

TSP thuộc bài toán NP-hard. Nghĩa là khi số lượng thành phố tăng lên, độ phức tạp của việc tìm lời giải tối ưu tăng lên theo hàm mũ. Đối với TSP đối xứng, với N thành phố thì số lượng lộ trình có thể được tính theo công thức:

(n - 1)!

**II. THUẬT TOÁN GREEDY TRAVELING SALESMAN (GTS)**

### **1. Ý tưởng**

Thuật toán là một phương pháp giải quyết bài toán người bán hàng (Traveling Salesman Problem – TSP) bằng cách sử dụng cách tiếp cận tham lam (Greedy). Bài toán TSP yêu cầu tìm đường đi ngắn nhất từ từng thành phố để người bán hàng có thể đi qua tất cả các thành phố một lần và trở về thành phố xuất phát ban đầu.

### **2. Các bước thực hiện**

**Bước 1:** Khởi tạo

+ Chọn một thành phố làm điểm xuất phát (ngẫu nhiên)

+ Đánh dấu thành phố này đã thăm

**Bước 2:** Tìm thành phố gần nhất chưa thăm

+ Xác định tất cả các thành phố chưa thăm qua

+ Tính khoảng cách từ thành phố hiện tại đến tất cả các thành phố chưa thăm

+ Chọn thành phố có khoảng cách ngắn nhất từ thành phố hiện tại đến nó

**Bước 3:** Di chuyển đến thành phố gần nhất

+ Di chuyển đến thành phố gần nhất vừa chọn ở Bước 2

+ Đánh dấu thành phố đó là đã thăm

**Bước 4:** Lặp lại

+ Lặp lại các bước trên đến khi các thành phố đều đã được thăm

**Bước 5:** Kết thúc

+ Quay lại thành phố bắt đầu và kết thúc.

### **3. Ví dụ minh họa**

**Ví dụ 1:** Giả sử bạn là một người bán hàng cần ghé thăm 5 thành phố A, B, C, D, E. Nhiệm vụ là đi qua tất cả các thành phố một lần và quay lại thành phố xuất phát sao cho quãng đường di chuyển ngắn nhất có thể.

Khoảng cách giữa các thành phố như sau:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| A | 0 | 10 | 15 | 20 | 25 |
| B | 10 | 0 | 35 | 25 | 30 |
| C | 15 | 35 | 0 | 30 | 20 |
| D | 20 | 25 | 30 | 0 | 15 |
| E | 25 | 30 | 20 | 15 | 0 |

Bảng 1

**Giải**

***Bước 1:*** Khởi tạo. Chọn thành phố A làm điểm xuất phát

***Bước 2:*** Tìm thành phố gần nhất từ A

A -> B: 10 (Chọn thành phố B vì gần nhất)

A -> C: 15

A -> D: 20

A -> E: 25

***Bước 3:*** Cập nhật trạng thái: B đã được thăm (Bạn đang đứng ở thành phố B)

***Bước 4:*** Tìm thành phố gần nhất từ B

B -> C: 35

B -> D: 25 (Chọn thành phố D vì gần nhất)

B -> E: 30

***Bước 5:*** Cập nhật trạng thái: D đã được thăm (Bạn đang đứng ở thành phố D)

***Bước 6:*** Tìm thành phố gần nhất từ D

D -> C: 30

D -> E: 15 (Chọn thành phố E vì gần nhất)

***Bước 7:*** Cập nhật trạng thái: E đã được thăm (Bạn đang đứng ở E)

***Bước 8:*** Tìm thành phố gần nhất từ E

E -> C: 20 (Chọn thành phố C vì chỉ còn C là thành phố chưa được thăm)

***Bước 9:*** Cập nhật trạng thái: C đã được thăm (Bạn đang đứng ở C)

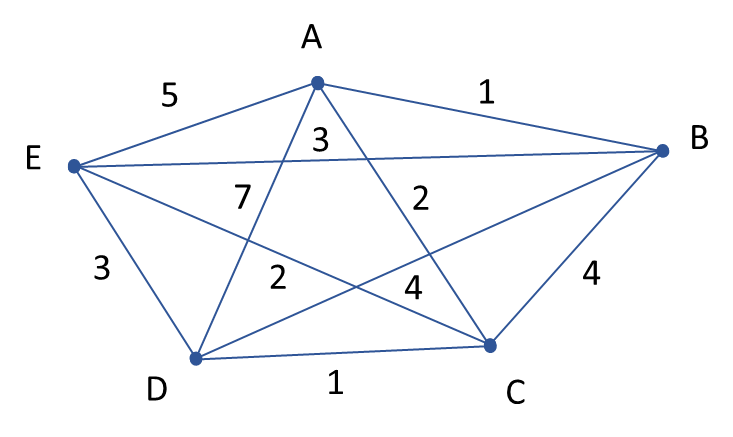
***Bước 10:*** Quay về thành phố xuất phát ban đầu

C -> A: 15

***Kết thúc:***

- Lộ trình: A -> B -> D -> E -> C -> A

- Chi phí di chuyển: 10 + 25 + 15 + 20 + 15 = 85

**Ví dụ 2:** Tìm chu trình qua 5 thành phố có chi phí tối thiểu. Nhiệm vụ là đi qua tất cả các thành phố một lần và quay lại thành phố xuất phát sao cho quãng đường di chuyển ngắn nhất có thể.

Hình 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| A | 0 | 1 | 2 | 7 | 5 |
| B | 1 | 0 | 4 | 4 | 3 |
| C | 2 | 4 | 0 | 1 | 2 |
| D | 7 | 4 | 1 | 0 | 3 |
| E | 5 | 3 | 2 | 3 | 0 |

Bảng 2

Tour = {}; Cost = 0; V = A

W {B, C, D, E}

→ W = B (có giá thành bé nhất)

Tour = {(A, B)}; Cost = 1; V = B

W {C, D, E}

→ W = E (có giá thành bé nhất)

Tour = {(A, B), (B, E)}; Cost = 1 + 3 = 4

V = E

W {C, D}

→ W = C (có giá thành bé nhất)

Tour = {(A, B), (B, E), (E, C)}; Cost = 4 + 2 = 6

V = C

W {D}

→ W = D (có giá thành bé nhất)

Tour = {(A, B), (B, E), (E, C), (C, D)}; Cost = 6 + 1 = 7

V = D

Tour = {(A, B), (B, E), (E, C), (C, D), (D, A)}; Cost = 7 + 7 = 14

***Kết quả:***

Tour A → B → E → C → D → A

Chi phí: Cost = 14

### **4. Ưu và nhược điểm**

#### **4.1. Ưu điểm**

- Đơn giản và dễ thực hiện: vì nó chỉ dựa trên một quy tắc là tại mỗi bước chỉ cần chọn thành phố gần nhất để đi tiếp.

- Tốc độ nhanh

- Thực tế và hữu ích trong tình huống khẩn cấp

#### **4.2. Nhược điểm**

- Không tối ưu: thuật toán chỉ đưa ra giải pháp tốt nhất ở thời điểm hiện tại, không xem xét các bước tiếp theo nên có thể bỏ qua các lộ trình tốt hơn.

- Thiếu linh hoạt: vì khi đã chọn một thành phố rồi thì không thể quay lại sửa đổi, dẫn đến kết quả có thể không tối ưu.

- Không phù hợp cho các bài toán lớn: vì thuật toán không xem xét toàn bộ lộ trình, nên khi ta tăng số lượng thành phố lên thì việc chọn lộ trình chỉ dựa trên khoảng cách gần nhất không còn hiệu quả nữa, dẫn đến là lộ trình tìm được không tối ưu.

### **5. Ứng dụng thực tế**

Hãy tưởng tượng một công ty vận chuyển hàng hóa cần giao hàng đến nhiều địa điểm khác nhau. Công ty này muốn tối ưu hóa tuyến đường để tiết kiệm thời gian và nhiên liệu. Tuy nhiên, số lượng địa điểm rất lớn, và việc tính toán tuyến đường tốt nhất qua tất cả các địa điểm có thể rất phức tạp.

**GTS (Greedy Traveling Salesman)** giúp giải quyết vấn đề này bằng cách **tìm đường đi ngắn nhất từng bước một**. Ví dụ: khi đến một địa điểm, xe giao hàng sẽ chọn địa điểm **gần nhất** để đi tiếp, và cứ thế tiếp tục cho đến khi giao xong. Nhờ phương pháp này, thời gian xử lý sẽ nhanh hơn, vì họ không phải tính toán mọi khả năng có thể cho toàn bộ tuyến đường.

Hay trong các thành phố lớn, việc quản lý giao thông là một thách thức. Khi cần tối ưu hóa lộ trình của xe buýt, xe tải hay các phương tiện vận tải công cộng, việc tính toán tất cả các tuyến đường tốt nhất có thể mất rất nhiều thời gian.

Sử dụng GTS, các phương tiện có thể chọn tuyến đường tối ưu theo kiểu **từng đoạn một**. Điều này không hoàn hảo như phương pháp tối ưu hóa toàn cục, nhưng nó đủ tốt để đưa ra giải pháp nhanh chóng và hiệu quả trong thực tế, giúp giảm bớt kẹt xe và tiết kiệm thời gian cho các phương tiện.

### **6. So sánh với một số thuật toán khác**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Thuật toán** | **Ưu điểm** | **Nhược điểm** | **Độ phức tạp** | **Chất lượng giải pháp** |
| Brute Force | Đảm bảo tối ưu | Độ phức tạp rất cao | O(n!) | Tốt nhất |
| GTS | Đơn giản, nhanh | Không đảm bảo tối ưu | O(n2) | Trung bình |
| Thuật toán di truyền | Linh hoạt, tìm nhiều giải pháp | Không đảm bảo tối ưu, cần nhiều lần lặp | Phụ thuộc vào cài đặt | Tốt |

Bảng 3

### **7. Triển khai ứng dụng**

***Tổng quan chức năng trang web***

Trang web của nhóm là một trang web đơn giản, tập trung vào việc khám phá và trình bày các di tích lịch sử tại Thành phố Hồ Chí Minh. Với giao diện thân thiện và tích hợp công nghệ bản đồ, trang web của nhóm sẽ mang lại trải nghiệm tương tác với bản đồ và những tính năng hữu ích cho người dùng. Các tính năng chính bao gồm:

* Một trang chủ
* Một trang thể hiện danh sách các di tích và mô tả cho các di tích
* Một trang dùng thuật toán để xác định nên đường đi giữa các di tích lịch sử

***Trang Chủ - trang giao diện chính***

Trang chủ của hệ thống cung cấp cái nhìn tổng quan về mục đích và các chức năng của trang web. Giao diện được thiết kế trực quan với thanh điều hướng cho phép người dùng dễ dàng chuyển đổi giữa các trang con như "Trang Chủ", "Di Tích", "Hành Trình", và "GET Data From API".

***Trang Di Tích – trang danh sách và mô tả các địa điểm lịch sử***

Trang này trình bày chi tiết danh sách các di tích lịch sử nổi bật tại TP.HCM, đi kèm thông tin mô tả ngắn gọn. Đây là nơi người dùng có thể tìm hiểu thêm về các địa điểm di sản văn hóa với hình ảnh minh họa và nội dung hấp dẫn.

***Trang Hành Trình – trang lập kế hoạch di chuyển***

Trang của nhóm cung cấp cho người dùng chức năng lập kế hoạch hành trình khám phá các di tích lịch sử ở TP.HCM. Các tính năng và các đặc điểm như sau:

* **Tích hợp bản đồ tương tác:** Người dùng có thể quan sát bản đồ trực tiếp, phóng to, thu nhỏ, kéo để di chuyển và chọn điểm bắt đầu hành trình.
* **Thuật toán TSP Greedy:** Trang sử dụng thuật toán "Greedy Travelling Salesman Problem" để tính toán hành trình tối ưu nhất giữa các di tích lịch sử, giúp cho người dùng thuận tiện cho việc lên lịch trình cho chuyến đi qua các di tích một cách ngắn nhất trên mỗi di tích.
* **Hiển thị hành trình:** Các điểm đến được đánh dấu trên bản đồ, kết hợp thêm tên di tích, cùng với đường đi giữa các di tích được vẽ bằng màu xanh sẽ giúp người dùng dễ theo dõi. Hành trình tối ưu xác định được cũng được liệt kê dưới dạng danh sách ở bên trái màn hình của trang web.
* **Thông tin chi tiết:** Người dùng có thể nhấp vào các điểm trên bản đồ để xem thêm thông tin chi tiết như tên và địa chỉ.

# **KẾT LUẬN**

Greedy Traveling Salesman(GTS) là một thuật toán gần đúng, hiệu quả với các bài toán nhỏ. Nó có ưu thế về mặt thời gian thực thi và dễ triển khai. Nhưng bên cạnh đó thì chất lượng lời giải có thể không tối ưu trong các bài toán lớn hoặc các bài toán phức tạp. Chính vì vậy mà để cải thiện hiệu quả, GTS có thể kết hợp với các phương pháp khác hoặc sử dụng như một bước khởi tạo trong các thuật toán mạnh hơn.

# **TÀI LIỆU THAM KHẢO**

1. Geeksforgeeks. (2023, 08 19). *Traveling Salesman Problem using Dynamic Programming*, https://www.geeksforgeeks.org/traveling-salesman-problem-tsp-implementation/
2. Trường Đại học Công thương TPHCM (2024). *Slide bài giảng Trí tuệ nhân tạo.*
3. TRACKOBIT. (2023, 9 14). *What is a Traveling Salesman Problem (TSP)?*, https://trackobit.com/blog/what-is-a-traveling-salesman-problem-explained
4. Tutorialspoint. (n.d.). *Traveling Salesman Problem (Greedy Approach)*, https://www.tutorialspoint.com/data\_structures\_algorithms/Traveling\_salesman\_problem.htm

# **PHỤ LỤC**

**Phụ lục 1. Biên bản họp nhóm**

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**Độc lập – Tự do – Hạnh phúc**

**BIÊN BẢN HỌP NHÓM**

**1. Thời gian, địa điểm, thành phần tham dự**

**1.1. Thời gian:**

**1.2. Địa điểm:**

**1.3. Thành phần tham dự:**

+ Chủ trì: Huỳnh Thanh Sơn (Trưởng nhóm)

+ Tham dự: 3

+ Vắng: 0

**2. Nội dung cuộc họp**

**2.1. Nhóm trưởng phân công công việc cho các thành viên như sau:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Họ tên** | **MSSV** | **Nhiệm vụ** | **Ghi chú** |
| Huỳnh Thanh Sơn | 2001224190 | Code Demo, PowerPoint, Word, Thuyết trình |  |
| Trần Thị Yến Nhi | 2001223311 | PowerPoint, Word, Thuyết trình |  |
| Mai Lê Bảo Khang | 2001221995 | PowerPoint, Word, Thuyết trình |  |

**2.2. Ý kiến của các thành viên:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Họ tên** | **MSSV** | **Đồng ý** | **Không đồng ý** | **Khác** |
| Huỳnh Thanh Sơn | 2001224190 | 🗹 |  |  |
| Trần Thị Yến Nhi | 2001223311 | 🗹 |  |  |
| Mai Lê Bảo Khang | 2001221995 | 🗹 |  |  |

**2.3. Kết luận cuộc họp**

Thống nhất lại nội dung cuộc họp sau khi có ý kiến của từ thành viên

*(Đây là căn cứ đánh giá mức độ hoàn thành công việc của từng thành viên)*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Họ tên** | **MSSV** | **Nhiệm vụ** | **Đánh giá mức độ hoàn thành** | **Ghi chú** |
| Huỳnh Thanh Sơn | 2001224190 | Code Demo, PowerPoint, Word, Thuyết trình |  |  |
| Trần Thị Yến Nhi | 2001223311 | PowerPoint, Word, Thuyết trình |  |  |
| Mai Lê Bảo Khang | 2001221995 | PowerPoint, Word, Thuyết trình |  |  |

Cuộc họp đi đến thống nhất và kết thúc lúc giờ phút cùng ngày.

**Phụ lục 2. Tiêu chí đánh giá**

|  |  |  |
| --- | --- | --- |
| **Tiêu chí đánh giá** | | **Thang điểm** |
| Cấu trúc | |  |
| Nội dung | Các nội dung thành phần |  |
| Lập luận |  |
| Kết luận/kết quả |  |
| Hình thức trình bày | |  |
| **Tổng** | |  |