**BỘ CÔNG THƯƠNG**

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

**KHOA CÔNG NGHỆ THÔNG TIN**

-----o0o----



**TÊN ĐỀ TÀI:** **THUẬT GIẢI AT(ALGORITHM FOR TREE)**

|  |  |
| --- | --- |
| **Nhóm 4:**  **Mssv: 2001225895**  **Họ Và Tên: Nguyễn Tuấn Vũ** | **Giảng Viên Hướng Dẫn:**  **Trần Như Ý** |

**Thành phố Hồ Chí Minh, tháng 8 năm 2024**

Mục Lục

[**LỜI MỞ ĐẦU** 3](#_Toc176439155)

[**1:GIỚI THIỆU** 4](#_Toc176439156)

[**1.1 Lịch sử phát triển** 4](#_Toc176439157)

[**1.2 Ý tướng thuật toán** 4](#_Toc176439158)

[**2.THUẬT GIẢI AT** 5](#_Toc176439159)

[**2.1 Khái niệm về thuật giải** 5](#_Toc176439160)

[**2.2 Cách thức hoạt động** 5](#_Toc176439161)

[**2.3 Mô tả thuật toán** 6](#_Toc176439162)

[**2.4 Độ phức tạp** 6](#_Toc176439163)

[**3. PHÂN TÍCH THUẬT TOÁN** 7](#_Toc176439164)

[**3.1.Bài toán dùng thuật giải AT sơ đồ cây** 7](#_Toc176439165)

[**4.ƯU VÀ NHƯỢC NHƯỢT ĐIỂM CỦA THUẬT GIẢI** 9](#_Toc176439166)

[**4.1.Ưu điểm** 9](#_Toc176439167)

[**4.2.Nhược điểm** 9](#_Toc176439168)

[**5.SO SÁNH VƠI CÁC THUẬT GIẢI AKT** 10](#_Toc176439169)

[**5.1 Giống Nhau** 10](#_Toc176439170)

[**5.1 Khác Nhau** 11](#_Toc176439171)

[**6.ỨNG DỤNG** 12](#_Toc176439172)

[TÀI LIỆU THAM KHẢO 13](#_Toc176439173)

**Phân Công Công Việc**

|  |  |  |
| --- | --- | --- |
| **Họ Tên** | **Phân Công Công Việc** | **Hoàn Thành** |
| Nguyễn Tuấn Vũ | Word, code demo,ppt | 100% |

# **LỜI MỞ ĐẦU**

Trí Tuệ Nhân Tạo (AI) là một lĩnh vực khoa học và công nghệ đang ngày càng thu hút sự quan tâm mạnh mẽ từ cả giới chuyên môn lẫn công chúng. Từ những bộ phim khoa học viễn tưởng cho đến những ứng dụng thực tiễn trong cuộc sống hàng ngày, AI đang dần trở thành một phần không thể thiếu của xã hội hiện đại. Không chỉ đơn thuần là những cỗ máy có khả năng thực hiện các nhiệm vụ mà con người giao phó, AI còn có tiềm năng vượt xa hơn, tự học hỏi, thích nghi và thậm chí đưa ra các quyết định phức tạp dựa trên dữ liệu.

Trong bối cảnh cuộc cách mạng công nghiệp lần thứ tư, AI đóng vai trò là một trong những công nghệ chủ đạo, mang lại những đột phá đáng kể trong nhiều lĩnh vực như y tế, giáo dục, kinh doanh, và cả nghệ thuật. AI không chỉ giúp tối ưu hóa quy trình làm việc, nâng cao hiệu quả sản xuất mà còn mở ra những khả năng mới trong việc sáng tạo và khám phá những điều chưa từng được nghĩ đến.

Tuy nhiên, cùng với những cơ hội mà AI mang lại là những thách thức không nhỏ về đạo đức, bảo mật và tác động xã hội. Việc phát triển AI cũng đặt ra câu hỏi về tương lai của con người trong một thế giới ngày càng bị chi phối bởi máy móc và các thuật toán. Liệu AI sẽ trở thành một công cụ hỗ trợ đắc lực hay sẽ vượt qua con người về khả năng tư duy và sáng tạo?

Bài thuyết trình này sẽ giúp bạn khám phá sâu hơn về Trí Tuệ Nhân Tạo, từ lịch sử phát triển, các nguyên lý cơ bản đến những ứng dụng thực tiễn và tương lai của nó. Qua đó, chúng ta sẽ cùng suy ngẫm về vai trò của AI trong thế giới ngày mai và những cơ hội cũng như thách thức mà nó mang lại.

# **1:GIỚI THIỆU**

## **1.1 Lịch sử phát triển**

BFS khá đơn giản. Tuy vậy, trên thực tế, cũng như tìm kiếm chiều sâu và chiều rộng, hiếm khi ta dùng BFS một cách trực tiếp. Thông thường, người ta thường dùng các phiên bản của BFS là AT, AKT hoặc A\*

## **1.2 Ý tướng thuật toán**

Thuật giải "AT" là một phương pháp tìm kiếm sử dụng cách tiếp cận tương tự như thuật toán BFS, nhưng thay vì đơn thuần chỉ mở rộng các nút dựa trên mức độ sâu (độ sâu của nút trong cây tìm kiếm), nó đánh giá độ tốt của mỗi nút dựa trên giá trị hàm g(n). Đây là tổng chiều dài con đường đã đi từ trạng thái bắt đầu đến trạng thái hiện tại.

Độ tốt của một nút n trong thuật giải AT được xác định bởi giá trị hàm g(n). Đây là tổng chi phí hoặc chiều dài của đường đi từ trạng thái bắt đầu đến trạng thái n. Trong thuật toán BFS tiêu chuẩn, chi phí này thường không được quan tâm, vì BFS chỉ tập trung vào việc tìm kiếm tất cả các nút ở một mức trước khi tiến sang mức tiếp theo.

Trong thuật giải AT, có thể xảy ra tình huống mà nút được mở rộng là nút có giá trị g(n) nhỏ nhất. Điều này tương tự như cách thức của một số thuật toán tìm kiếm chi phí thấp nhất (Uniform Cost Search).

So với BFS thông thường không quan tâm đến chi phí đường đi mà chỉ tập trung vào mức độ của các nút. AT, ngược lại, quan tâm đến tổng chi phí (g) để đánh giá nút nào nên được mở rộng.

# **2.THUẬT GIẢI AT**

## **2.1 Khái niệm về thuật giải**

Là một phương pháp tìm kiếm theo kiểu BFS với độ tốt của nút là giá trị hàm g tổng chiều dài con đường đã đi từ trạng thái bắt đầu đến trạng thái hiện tại.

Tìm kiếm thông qua tất cả các trạng thái: Thuật giải sẽ tiếp tục tìm kiếm cho đến khi tìm thấy đích hoặc không còn trạng thái nào để khám phá.

Phù hợp với các bài toán cần tìm đường đi tối ưu dựa trên chi phí thực tế đã đi: AT sẽ tìm ra đường đi có chi phí thấp nhất nếu có thể.

## **2.2 Cách thức hoạt động**

1. Đặt OPEN chứa trạng thái khởi đầu.

2. Cho đến khi tìm được trạng thái đích hoặc không còn nút nào trong OPEN, thực hiện :

2.a. Chọn trạng thái (Tmax) có giá trị g nhỏ nhất trong OPEN (và xóa Tmax khỏi OPEN)

2.b. Nếu Tmax là trạng thái kết thúc thì thoát.

2.c. Ngược lại, tạo ra các trạng thái kế tiếp Tk có thể có từ trạng thái Tmax. Đối với mỗi trạng thái kế tiếp Tk thực hiện :

g(Tk) = g(Tmax) + cost(Tmax, Tk);

Thêm Tk vào OPEN.

## **2.3 Mô tả thuật toán**

Mỗi đỉnh n tương ứng với một số g(n): giá thành  của đường đi từ đỉnh ban đầu đến đỉnh n.

Đỉnh:

+ Đỉnh đóng (Closed) : là những đỉnh đã được  xem xét.

+ Đỉnh ẩn (Hiden) : là những đỉnh mà tại  đó hàm g(n) chưa được xác định.

+ Đỉnh mở (Open) : là những đỉnh giả thiết  sẽ được xem xét ở bước sau.

**Bước 1**: + Mọi đỉnh n, mọi giá trị g(n) đều là ẩn. + Mở đỉnh đầu tiên và gọi đó là đỉnh S. Đặt g(S) = 0. **Bước 2** : Chọn đỉnh mở với giá thành g tương ứng là nhỏ nhất và gọi đó là đỉnh N.

+ Nếu N là mục tiêu: đường đi từ đỉnh ban đầu đến N là đường đi ngắn nhất và bằng  g(N). Dừng (Success). + Nếu không tồn tại một đỉnh mở nào nữa: cây biểu diễn vấn đề không có đường đi tới  mục tiêu. Dừng (Fail). + Nếu tồn tại nhiều hơn 1 đỉnh N (nghĩa là có 2 đỉnh N trở lên) mà có cùng giá thành  g(N) nhỏ nhất. Kiểm tra xem trong số đó có đỉnh nào là đích hay không. Nếu có: đường đi từ đỉnh ban đầu đến đỉnh N là ngắn nhất và bằng g(N), dừng  (Success). Nếu không có: Chọn ngẫu nhiên một trong các đỉnh đó và gọi là đỉnh N. **Bước 3**:Đóng đỉnh N và mở các đỉnh sau N (là những đỉnh có cung hướng từ N tới). Tại  mọi đỉnh S sau N tính : g(S) = g(N) + cost(N→S)

**Bước 4**: Quay lại bước 2

## **2.4 Độ phức tạp**

Độ phức tạp thời gian (Time Complexity): O(b^d):

Thuật giải AT hoạt động dựa trên tìm kiếm theo chiều rộng (BFS) nhưng với độ tốt của nút được đánh giá bởi giá trị hàm g(n). Do đó, độ phức tạp thời gian của thuật giải này thường tương tự với BFS nhưng có một số khác biệt:

* Trường hợp trung bình (Average-case): Trường hợp trung bình thường phụ thuộc vào cấu trúc của không gian trạng thái và cách mà chi phí g(n) phân bố. Nếu có nhiều nhánh không tối ưu, thời gian tìm kiếm có thể tăng lên đáng kể so với BFS.

Độ phức tạp không gian (Space Complexity**)**: O(b^d)

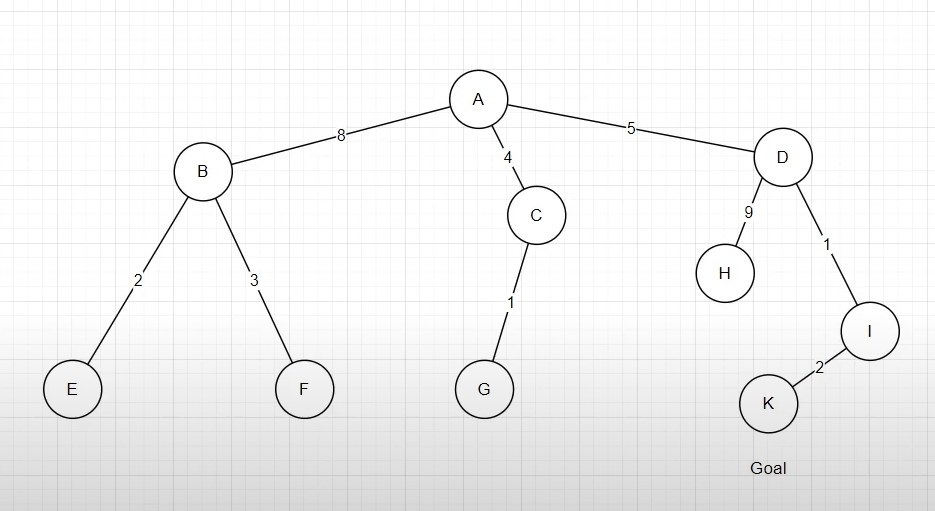
Độ phức tạp về không gian của thuật giải AT cũng tương tự như BFS, vì nó cần lưu trữ tất cả các nút đang chờ được mở rộng trong hàng đợi Open.

* Trường hợp xấu nhất (Worst-case): Hàng đợi Open có thể chứa số lượng nút bằng toàn bộ không gian trạng thái trong trường hợp xấu nhất, dẫn đến độ phức tạp về không gian là O(b^d).

# **3. PHÂN TÍCH THUẬT TOÁN**

## **3.1.Bài toán dùng thuật giải AT sơ đồ cây**

* Giải thuật AT tìm đường đi từ A -> K với chi phí thấp nhất



Phân Tích

|  |  |  |  |
| --- | --- | --- | --- |
| N | B(N) | MO | DONG |

Điểm Xuất Phát: A

Điểm Đích : K

N: là đinh đang xét

B(n): đỉnh kề với đỉnh đang xét

MO: chưa các đỉnh đang xét

DONG: đỉnh đã xét rồi

* Đầu tiên ta xét từ điểm xuất phát

|  |  |  |  |
| --- | --- | --- | --- |
| N | B(N) | MO | DONG |
|  |  | A(0) |  |
| A | B(8),C(4),D(5) | B(8),C(4),D(5) | A |

* Nếu đỉnh vừa xét không phải điểm đích ta tiếp tục lấy đỉnh trong tập MO có trong trọng số nhỏ nhất ra xét

|  |  |  |  |
| --- | --- | --- | --- |
| N | B(N) | MO | DONG |
|  |  | A(0) |  |
| A | B(8),C(4),D(5) | B(8),C(4),D(5) | A |
| C | G(1) | B(8),D(5),G(5) | A,C |

* Nếu đỉnh đang xét có đỉnh kề thì ta phải cộng các trọng số từ đỉnh đầu đến đỉnh kề tiếp theo

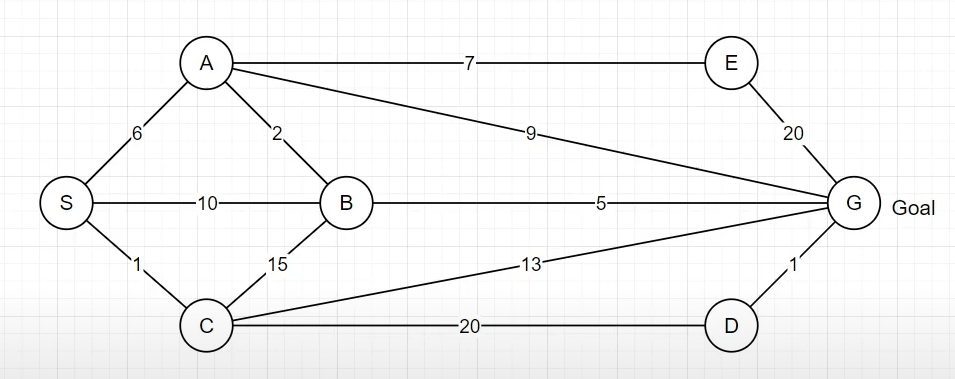
|  |  |  |  |
| --- | --- | --- | --- |
| N | B(N) | MO | DONG |
|  |  | A(0) |  |
| A | B(8),C(4),D(5) | B(8),C(4),D(5) | A |
| C | G(1) | B(8),D(5),G(5) | A,C |
| D | H(14), I(6) | B(8),G(5),H(14),I(6) | A,C,D |

* Nếu đỉnh đang xét không có điểm kề thì ta tiếp tục xét các đỉnh còn lại trong tập “MO”
* Cứ như vậy cho đến khi tìm được điểm đích

|  |  |  |  |
| --- | --- | --- | --- |
| N | B(N) | MO | DONG |
|  |  | A(0) |  |
| A | B(8),C(4),D(5) | B(8),C(4),D(5) | A |
| C | G(1) | B(8),D(5),G(5) | A,C |
| D | H(14), I(6) | B(8),G(5),H(14),I(6) | A,C,D |
| G |  | B(8),H(14),I(6) | A,C,D,G |
| I | K(8) | B(6),H(14),K(8) | A,C,D,G,I |
| B | E(10),F(11) | H(14),K(8),E(10),F(11) | A,C,D,G,I,B |
| K | GOAL |  |  |

* Đường đi từ A-> K với tổng chi phí thấp nhát là: 8
* A -> D -> I -> K

# **3.2 Dạng đồ thị**



* Giải thuật AT tìm đường đi từ S -> G với chi phí thấp nhất

Phân Tích

|  |  |  |  |
| --- | --- | --- | --- |
| N | B(N) | MO | DONG |

Điểm Xuất Phát: A

Điểm Đích : K

N: là đinh đang xét

B(n): đỉnh kề với đỉnh đang xét

MO: chưa các đỉnh đang xét

DONG: đỉnh đã xét rồi

* Đầu tiên ta xét từ điểm xuất phát

|  |  |  |  |
| --- | --- | --- | --- |
| N | B(N) | MO | DONG |
|  |  | S(0) |  |
| S | A(6),B(10),C(1) | A(6),B(10),C(1) | S |

* Nếu đỉnh vừa xét không phải điểm đích ta tiếp tục lấy đỉnh trong tập MO có trong trọng số nhỏ nhất ra xét

|  |  |  |  |
| --- | --- | --- | --- |
| N | B(N) | MO | DONG |
|  |  | S(0) |  |
| S | A(6),B(10),C(1) | A(6),B(10),C(1) | S |
| C | B(16), G(14),D(21) | A(6),B(10),B(16),G(14),D21 | C |

* Nếu đỉnh đang xét có đỉnh kề thì ta phải cộng các trọng số từ đỉnh đầu đến đỉnh kề tiếp theo

|  |  |  |  |
| --- | --- | --- | --- |
| N | B(N) | MO | DONG |
|  |  | S(0) |  |
| S | A(6),B(10),C(1) | A(6),B(10),C(1) | S |
| C | B(16), G(14),D(21) | A(6),B(10),B(16),G(14),D21 | C |
| A | E(13),G(15),B(8) | B(10),B(16),G(14),D21,E(13),G(15),B(8) |  |

* Nếu đỉnh đang xét không có điểm kề thì ta tiếp tục xét các đỉnh còn lại trong tập “MO”
* Cứ như vậy cho đến khi tìm được điểm đích

|  |  |  |  |
| --- | --- | --- | --- |
| N | B(N) | MO | DONG |
|  |  | S(0) |  |
| S | A(6),B(10),C(1) | A(6),B(10),C(1) | S |
| C | B(16), G(14),D(21) | A(6),B(10),B(16),G(14),D21 | S,C |
| A | E(13),G(15),B(8) | B(10),B(16),G(14),D21,E(13),G(15),B(8) | S,C,A |
| B(8) | A(10),G(13),C(23) | B(10),B(16),G(14),D21,E(13),G(15),G(13),C(23) | S,C,A,B |
| B(10) | A(12),G(15),C(25) | B(10),B(16),G(14),D21,E(13),G(15),G(13),C(23), | S,C,A,B |
| E | G(33) | B(10),B(16),G(14),D21,G(15),G(13),C(23) | S,C,A,BE |
| G | GOAL |  |  |

* Đường đi từ S-> G với tổng chi phí thấp nhất là: 13

# **4.ƯU VÀ NHƯỢC NHƯỢT ĐIỂM CỦA THUẬT GIẢI**

## **4.1.Ưu điểm**

**Tìm kiếm theo chi phí thực tế:** AT ưu tiên các nút dựa trên chi phí thực tế đã đi, tức là giá trị g(n). Điều này đảm bảo rằng thuật giải có thể tìm thấy đường đi có chi phí thấp nhất từ trạng thái bắt đầu đến một trạng thái nào đó, mà không cần ước lượng chi phí còn lại như trong thuật toán A\*.

**Đơn giản và dễ triển khai:** Thuật giải AT dựa trên tìm kiếm theo chiều rộng (BFS) với một chút biến đổi để xem xét chi phí, do đó nó tương đối đơn giản và dễ triển khai.

**Tìm giải pháp tối ưu trong không gian trạng thái có chi phí đồng nhất:** Trong các bài toán mà chi phí chuyển đổi giữa các trạng thái là đồng nhất hoặc gần như đồng nhất, thuật giải AT có thể tìm ra giải pháp tối ưu một cách hiệu quả.

**Khả năng thích ứng với các bài toán cụ thể:** AT có thể được tùy chỉnh hoặc kết hợp với các phương pháp khác để phù hợp với các bài toán cụ thể, chẳng hạn như tìm kiếm trong không gian trạng thái mà chi phí thực tế là yếu tố quan trọng nhất.

## **4.2.Nhược điểm**

**Độ phức tạp về thời gian và không gian cao:** Giống như BFS, AT có thể phải mở rộng tất cả các trạng thái trong không gian tìm kiếm trước khi tìm được giải pháp tối ưu, dẫn đến độ phức tạp về thời gian và không gian là O(b^d) trong trường hợp xấu nhất. Điều này khiến thuật giải AT không hiệu quả trong các không gian trạng thái lớn.

**Không sử dụng hàm heuristic:** AT chỉ dựa trên chi phí đã đi (g(n)) mà không sử dụng hàm heuristic (h(n)) để ước lượng chi phí còn lại đến đích. Điều này có thể dẫn đến việc thuật giải mở rộng nhiều nút hơn so với cần thiết, đặc biệt trong các bài toán mà việc ước lượng chi phí còn lại là có giá trị.

**Không phù hợp với không gian trạng thái phức tạp:** Trong các không gian trạng thái lớn và phức tạp, nơi có nhiều nhánh không tối ưu hoặc không gian trạng thái không đồng nhất về chi phí, AT có thể gặp khó khăn và không hiệu quả.

**Tốc độ tìm kiếm có thể chậm:** Do không có khả năng dự đoán chi phí còn lại, AT có thể mất nhiều thời gian hơn để tìm ra giải pháp, đặc biệt trong các bài toán mà khoảng cách đến đích rất quan trọng.

# **5.SO SÁNH VƠI CÁC THUẬT GIẢI AKT**

## **5.1 Giống Nhau**

OPEN: Danh sách các trạng thái sẽ được xem xét hoặc mở rộng trong tương lai.

CLOSED: Danh sách các trạng thái đã được xem xét hoặc mở rộng.

Cả AT và AKT đều sử dụng OPEN để quản lý các trạng thái cần xử lý tiếp theo và có thể sử dụng CLOSED để tránh việc xử lý lại các trạng thái đã xem xét.

Hàm g(n): Đại diện cho chi phí thực tế từ trạng thái ban đầu đến trạng thái hiện tại n. cả hai thuật toán đều tính toán và cập nhật g(n) trong quá trình tìm kiếm để đánh giá chi phí đã đi qua.

Cả AT và AKT đều tiến hành mở rộng các trạng thái hiện tại bằng cách sinh ra các trạng thái kế tiếp dựa trên các hành động có thể thực hiện từ trạng thái hiện tại.

## **5.1 Khác Nhau**

|  |  |  |
| --- | --- | --- |
| Đặc điểm | Thuật giải AT | Thuật giải AKT |
| Hàm đánh giá | Chỉ sử dụng hàm g(n): giá trị đường đi thực tế từ trạng thái khởi đầu đến trạng thái n. | Sử dụng cả hàm g(n) và hàm ước lượng h'(n): f(n) = g(n) + h'(n). |
| Hàm f(n) | Không tồn tại, chỉ sử dụng g(n) để chọn trạng thái tốt nhất. | Sử dụng hàm f(n) là tổng của g(n) và ước lượng chi phí từ trạng thái hiện tại đến trạng thái đích h'(n). |
| Cách chọn trạng thái tốt nhất | Chọn trạng thái có g(n) nhỏ nhất trong tập OPEN. | Chọn trạng thái có f(n) nhỏ nhất trong tập OPEN (f(n) = g(n) + h'(n)). |
| Ước lượng | Không sử dụng hàm ước lượng, chỉ dựa vào chi phí thực tế. | Sử dụng hàm ước lượng h'(n) để giúp thuật toán tìm đường đi nhanh hơn, có thể dẫn đến việc mở ít trạng thái hơn. |
| Tính chất tìm kiếm | Tìm kiếm theo chi phí thực tế (tương tự Dijkstra). | Kết hợp tìm kiếm chi phí thực tế và ước lượng (tương tự A\*). |
| |  | | --- | | Hiệu quả |  |  | | --- | |  | | Thường mở rộng nhiều trạng thái hơn, vì không có dự đoán nào về trạng thái đích. | Hiệu quả hơn nếu hàm ước lượng h'(n) tốt, giúp tập trung tìm kiếm vào các trạng thái có khả năng dẫn đến đích. |

# **6.ỨNG DỤNG**

Bài toán di chuyển trong mạng lưới giao thông: Trong trường hợp tất cả các đoạn đường có chi phí di chuyển như nhau (ví dụ: cùng một khoảng cách, thời gian, hoặc nhiên liệu tiêu thụ), thuật giải AT có thể được sử dụng để tìm đường đi ngắn nhất từ điểm này đến điểm khác mà không cần ước lượng chi phí còn lại.

Tìm kiếm trong các mê cung đơn giản: Nếu tất cả các bước đi trong mê cung có chi phí giống nhau, AT có thể đảm bảo tìm ra đường đi ngắn nhất từ điểm xuất phát đến đích.

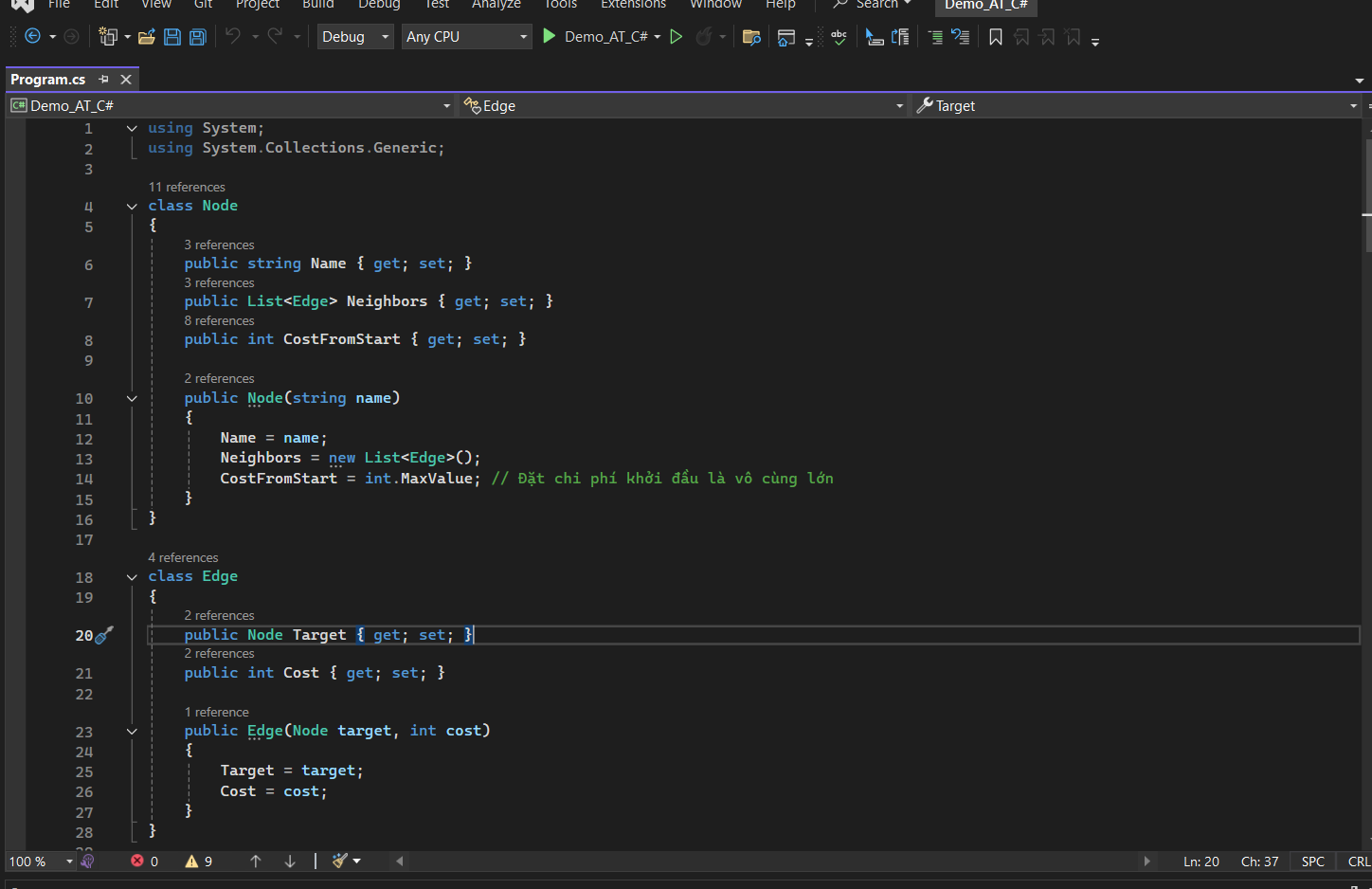
Lập lịch công việc với chi phí chuyển đổi cố định: Trong một số bài toán lập lịch, nếu chi phí chuyển đổi giữa các nhiệm vụ hoặc máy móc là đồng nhất, AT có thể tìm ra lịch trình tối ưu với chi phí thấp nhất bằng cách ưu tiên các nhiệm vụ có tổng chi phí thực tế thấp nhất

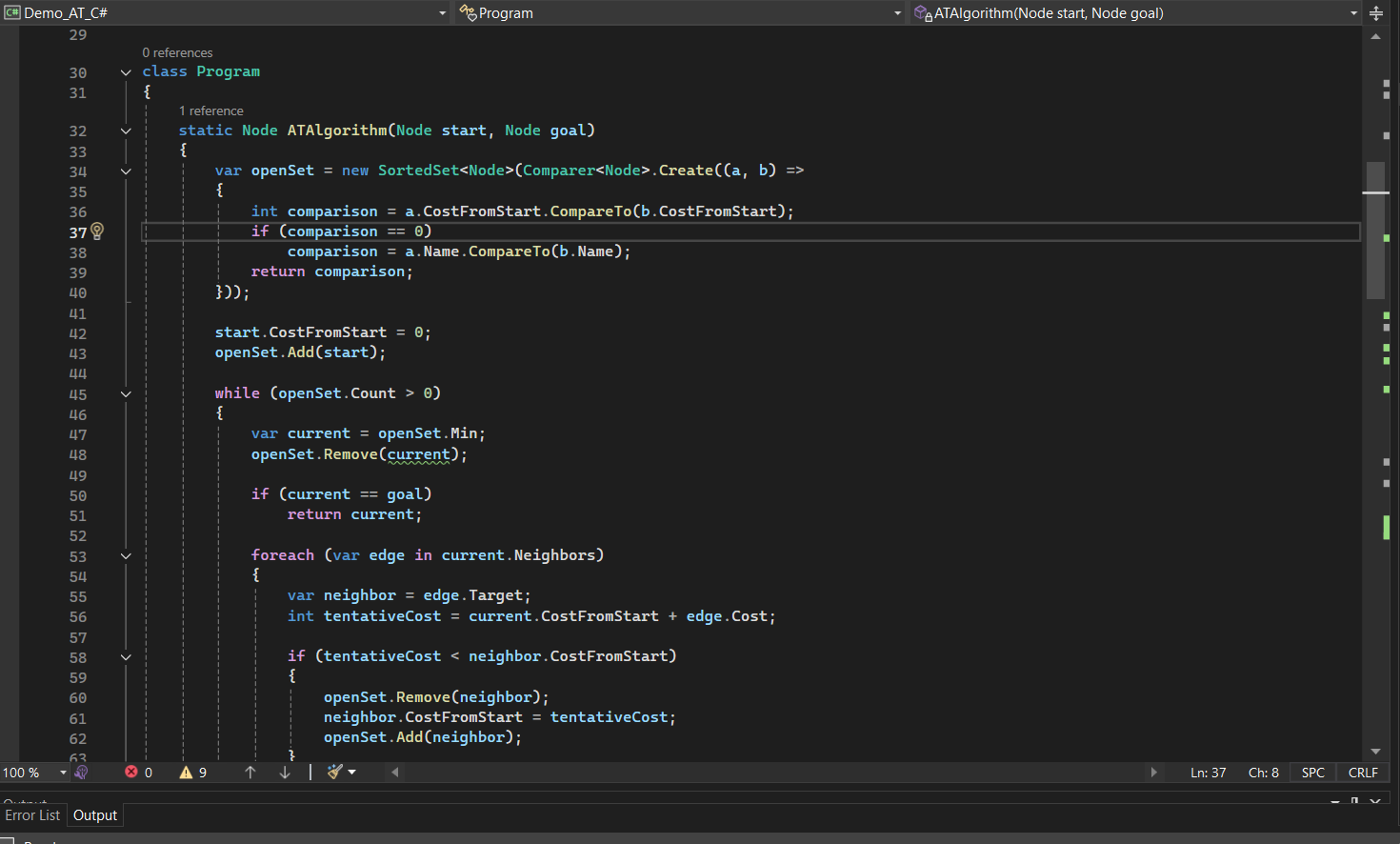
Tối ưu hóa các bài toán quản lý tài nguyên: Trong các hệ thống mà tài nguyên có chi phí sử dụng khác nhau (nhưng không biết trước chi phí tương lai), AT có thể giúp tìm ra các chiến lược phân bổ tài nguyên tối ưu dựa trên chi phí đã sử dụng cho đến thời điểm hiện tại.

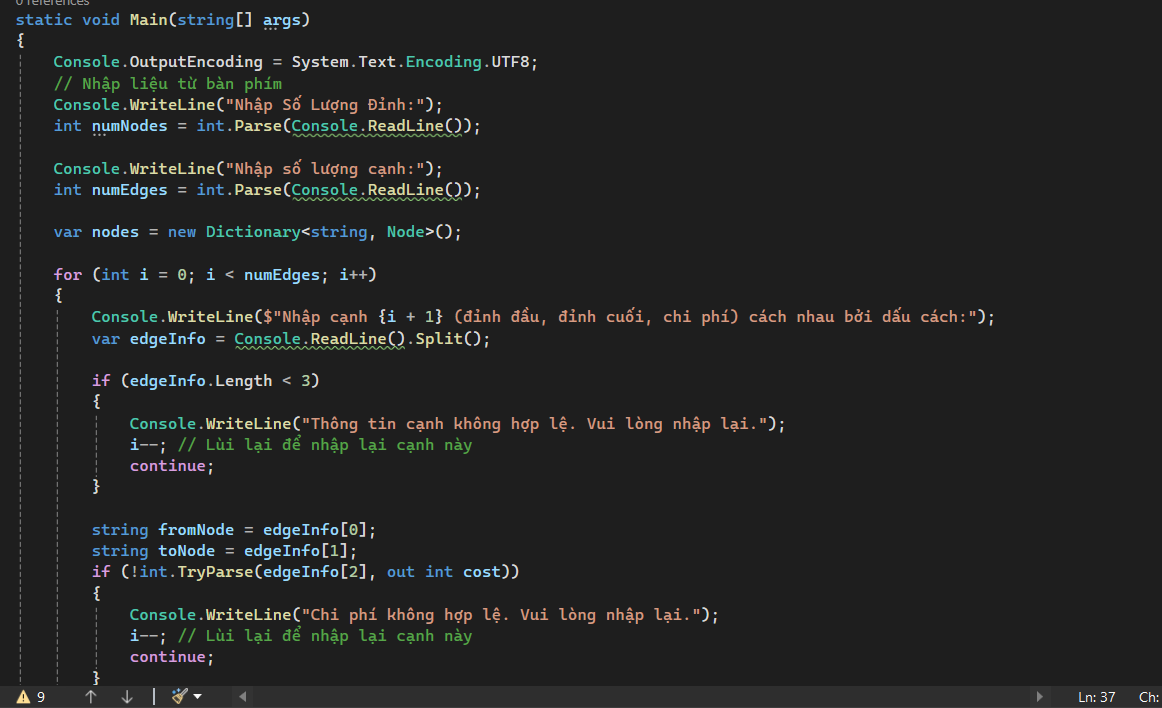
Quản lý di chuyển robot trong không gian có các rào cản đồng nhất: Khi di chuyển một robot trong một không gian mà các bước di chuyển có chi phí đồng nhất (ví dụ: các ô trên bàn cờ), AT có thể tìm ra lộ trình hiệu quả nhất từ điểm bắt đầu đến đích.

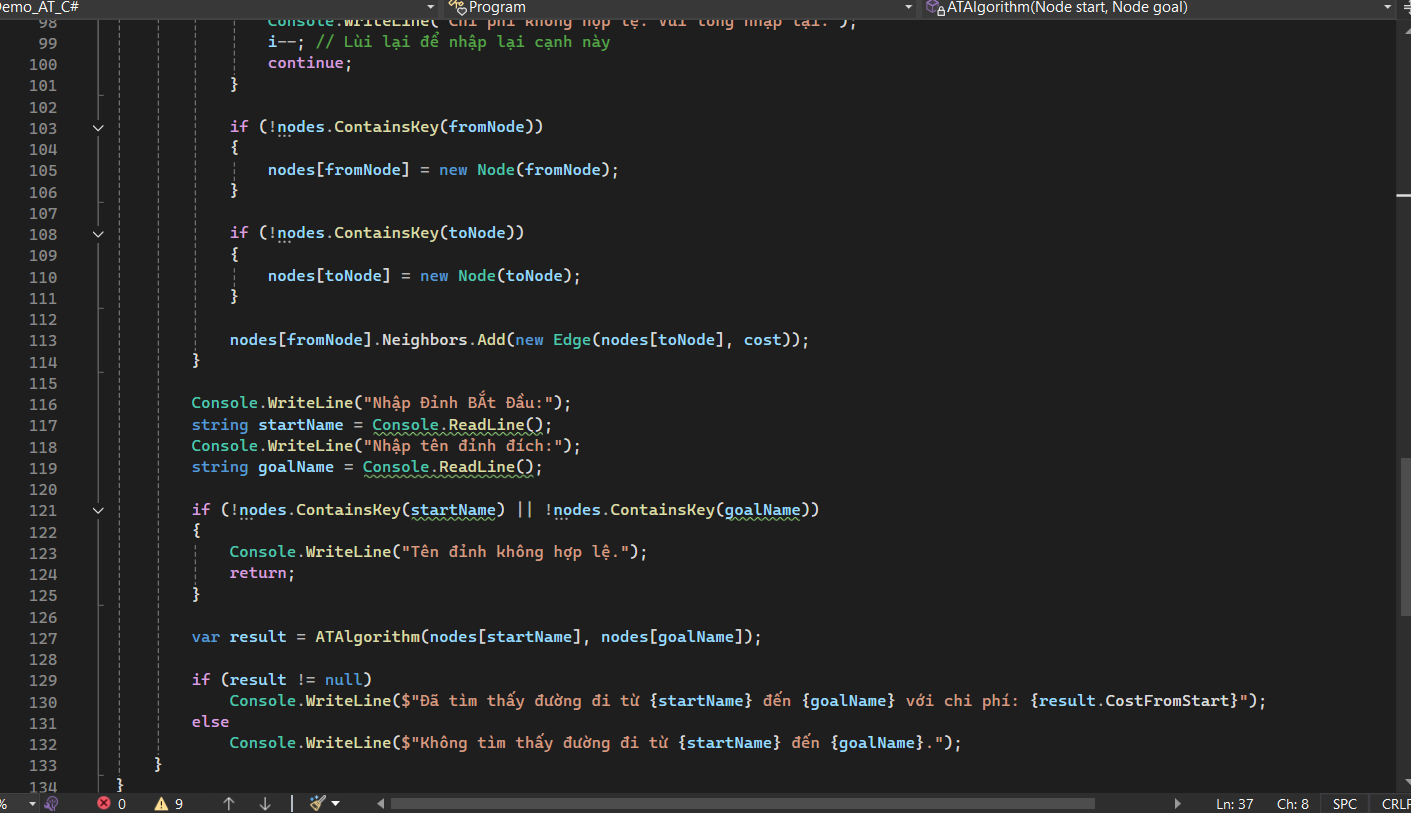
Tối ưu hóa đường đi của xe giao hàng: Trong một chuỗi cung ứng mà chi phí di chuyển giữa các điểm giao hàng là đồng nhất hoặc không thể dự đoán được chi phí tương lai, AT có thể giúp tìm ra lộ trình giao hàng với chi phí tối thiểu.

**7. Demo ví dụ**









# TÀI LIỆU THAM KHẢO

Tri Tue Nhan Tao: <https://www.researchgate.net/profile/Nguyen-Dinh-Cuong-2/publication/331310387_Artificial_Intelligence_TRI_TUE_NHAN_TAO_Bai_giang_Nha_trang-2012_NGUYEN_DINH_CUONG_Bo_mon_Cong_Nghe_Phan_Mem_Khoa_Cong_Nghe_Thong_Tin/links/5c71f2a7299bf1268d1fdb86/Artificial-Intelligence-TRI-TUE-NHAN-TAO-Bai-giang-Nha-trang-2012-NGUYEN-DINH-CUONG-Bo-mon-Cong-Nghe-Phan-Mem-Khoa-Cong-Nghe-Thong-Tin.pdf>

Thuat Giai AT, AKT: <https://doan.edu.vn/do-an/bai-giang-tri-tue-nhan-tao-bai-2-thuat-giai-at-akt-van-the-thanh-45833/>