TRƯỜNG ĐẠI HỌC HỌC VĂN LANG

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN MÔN HỌC**

**TRÍ TUỆ NHÂN TẠO**

NGÀNH: CÔNG NGHỆ THÔNG TIN

***Tên Đồ án:***

**ÁP DỤNG THUẬT TOÁN GIẢI THUẬT TÌM KIẾM**

**-BREADTH FIRST SEARCH**

**SVTH: ĐINH XUÂN TRỌNG**

**MSSV: 2274802010933**

**GVHD: TRẦN NGỌC VIỆT**

Tp. Hồ Chí Minh – năm 2024

LỜI CẢM ƠN-------------

Viết một báo cáo đồ án môn học là một trong những việc khó nhất mà chúng em phải  
hoàn thành trong quá trình học một môn học. Trong quá trình thực hiện đề tài chúng  
em đã gặp rất nhiều khó khăn và bỡ ngỡ. Nếu không có những sự giúp đỡ và lời động  
viên chân thành của nhiều người có lẽ chúng em khó có thể hoàn thành tốt tiểu luận  
này. Đầu tiên chúng em xin gửi lời biết ơn chân thành đến thầy Trần Ngọc Việt,  
người trực tiếp hướng dẫn chúng em hoàn thành tiểu luận này.  
Những ý kiến đóng góp của thầy là vô cùng hữu ích, nó giúp chúng em nhận ra các  
khuyết điểm của đồ án. Cảm ơn thầy và các bạn trường Đại học Văn Lang là những  
người đã cùng nhóm em sát cánh và trải nghiệm để hoàn thành đồ án môn học.  
Trong quá trình thực hiện đồ án này, tôi xin bày tỏ lòng biết ơn sâu sắc đến các thầy cô đã hướng dẫn và truyền đạt kiến thức, đặc biệt là giảng viên phụ trách đã nhiệt tình hướng dẫn và hỗ trợ tôi trong quá trình thực hiện đề tài. Bên cạnh đó, tôi cũng cảm ơn gia đình và bạn bè đã động viên, hỗ trợ tinh thần để tôi hoàn thành đồ án này một cách tốt nhất.

**Mục Lục**

[**1.1 Lý do chọn đề tài** 4](#_Toc182263447)

[**1.1.1 Giới thiệu về SuKoBan và các ứng dụng trong AI:** 4](#_Toc182263448)

[**1.1.2 Ý nghĩa và tính ứng dụng của bài toán:** 4](#_Toc182263449)

[**1.1.3 Lợi ích học thuật và phát triển kỹ năng giải quyết vấn đề:** 4](#_Toc182263450)

[**1.2** **Mục tiêu của đề tài** 4](#_Toc182263451)

[**1.2.1 Mục tiêu học thuật:** 4](#_Toc182263452)

[**1.2.2 Mục tiêu thực tiễn:** 4](#_Toc182263453)

[**2.1 Phát biểu bài toán:** 5](#_Toc182263454)

[**2.1.1 Khái niệm thuật toán BFS:** 5](#_Toc182263455)

[**2.1.2 Đặc điểm của BFS khi áp dụng cho bài toán SuKoBan:** 5](#_Toc182263456)

[**2.2 Hướng giải quyết bài toán** 5](#_Toc182263457)

[**2.2.1 Xác định không gian trạng thái** 5](#_Toc182263458)

[**2.2.2 Biểu diễn trạng thái và các hành động** 6](#_Toc182263459)

[**2.2.3 Triển khai thuật toán DFS** 6](#_Toc182263460)

[**2.3 Mã nguồn** 7](#_Toc182263461)

[**2.3.1** **Mã nguồn** 7](#_Toc182263462)

[**2.3.2 Kết Luận- Nhận xét** 8](#_Toc182263463)

[**3 KẾT LUẬN** 9](#_Toc182263464)

[**3.1.1 Tóm tắt quá trình thực hiện:** 9](#_Toc182263465)

[**3.1.2 Kết quả đạt được:** 9](#_Toc182263466)

[**3.1.3 Nhận xét và hạn chế:** 9](#_Toc182263467)

**CHƯƠNG 1. CƠ SỞ LÝ THUYẾT – BÀI TOÁN BREADTH**

**FIRST SEARCH**

# **1.1** Lý do chọn đề tài

## **1.1.1 Giới thiệu về SuKoBan và các ứng dụng trong AI:**

- Trò chơi SuKoBan, ban đầu là một trò chơi Nhật Bản với nguyên tắc đơn giản là đẩy các hộp vào vị trí đã định trong một mê cung, đã trở thành một trong những bài toán thú vị trong lĩnh vực trí tuệ nhân tạo. Việc giải quyết SuKoBan không chỉ là giải một trò chơi mà còn đòi hỏi kỹ năng lập trình, tối ưu thuật toán, và quản lý không gian trạng thái, tạo ra một môi trường lý tưởng để áp dụng các thuật toán tìm kiếm trong AI.

## **1.1.2 Ý nghĩa và tính ứng dụng của bài toán:**

- SuKoBan được sử dụng rộng rãi trong giảng dạy và nghiên cứu AI vì nó cung cấp một ví dụ điển hình cho việc giải quyết vấn đề tìm kiếm trong không gian trạng thái phức tạp, giúp sinh viên hiểu sâu hơn về các thuật toán tìm kiếm và các kỹ thuật tối ưu hóa.

## **1.1.3 Lợi ích học thuật và phát triển kỹ năng giải quyết vấn đề:**

- Đề tài này sẽ giúp người thực hiện củng cố các kỹ năng lập trình và thuật toán, đặc biệt là trong việc triển khai và tinh chỉnh các thuật toán tìm kiếm như BFS. Ngoài ra, việc nghiên cứu SuKoBan cũng giúp rèn luyện khả năng phân tích và giải quyết các bài toán trong lĩnh vực AI.

# **1.2** Mục tiêu của đề tài

## **1.2.1 Mục tiêu học thuật:**

- Đề tài này nhằm giúp người thực hiện hiểu sâu về thuật toán tìm kiếm, đặc biệt là thuật toán tìm kiếm theo chiều rộng (Breadth-First Search - BFS), và cách áp dụng nó vào bài toán SuKoBan. Người thực hiện sẽ nghiên cứu cách BFS có thể tìm ra giải pháp tối ưu cho bài toán SuKoBan, bao gồm cách BFS duyệt qua không gian trạng thái để tìm ra chuỗi hành động ngắn nhất, đồng thời hiểu rõ những ưu điểm và nhược điểm của thuật toán này trong các bài toán tìm kiếm.

## **1.2.2 Mục tiêu thực tiễn:**

- Mục tiêu của đề tài là xây dựng một chương trình giải quyết bài toán SuKoBan, có khả năng tìm ra chuỗi hành động tối ưu hoặc hợp lý nhất để đưa các hộp vào vị trí đích. Chương trình này sẽ sử dụng thuật toán BFS để đảm bảo tìm được giải pháp ngắn nhất về số bước di chuyển.

**CHƯƠNG 2. ÁP DỤNG BÀI TOÁN MINIMAX DEPTH FIRST SEACH**

# **2.1 Phát biểu bài toán:**

## **2.1.1 Khái niệm thuật toán BFS:**

- Thuật toán tìm kiếm theo chiều rộng (Breadth-First Search - BFS) là một thuật toán tìm kiếm trong cây hoặc đồ thị, bắt đầu từ nút gốc và duyệt qua tất cả các nút ở cùng độ sâu trước khi chuyển sang độ sâu tiếp theo. BFS đảm bảo tìm kiếm tất cả các trạng thái trong phạm vi gần nhất trước khi mở rộng ra các trạng thái xa hơn. Thuật toán này thường được sử dụng khi cần tìm kiếm giải pháp tối ưu, vì BFS sẽ tìm ra giải pháp có số bước ngắn nhất đầu tiên, miễn là không gian trạng thái không quá lớn.

## **2.1.2 Đặc điểm của BFS khi áp dụng cho bài toán SuKoBan:**

- Trong trò chơi SuKoBan, không gian trạng thái có thể được mô phỏng như một cây hoặc đồ thị, trong đó mỗi trạng thái đại diện cho một tình huống của trò chơi. Mỗi trạng thái bao gồm vị trí của người chơi và các hộp trên bản đồ. Thuật toán BFS sẽ duyệt qua tất cả các trạng thái từ trạng thái ban đầu, kiểm tra các hành động có thể thực hiện và tìm kiếm đến trạng thái đích, nơi tất cả các hộp đã được đẩy vào các vị trí đích.

# **2.2 Hướng giải quyết bài toán**

## **2.2.1 Xác định không gian trạng thái**

**- Định nghĩa trạng thái của trò chơi:** Mỗi trạng thái bao gồm vị trí của người chơi và vị trí của tất cả các hộp trên bản đồ. Trạng thái có thể được biểu diễn dưới dạng ma trận 2D, trong đó các ký hiệu thể hiện vị trí người chơi, hộp, tường và các ô đích.

**- Xác định trạng thái hợp lệ:** Một trạng thái hợp lệ là trạng thái trong đó người chơi và hộp không vượt ra ngoài bản đồ và không đẩy hộp vào các vị trí không hợp lệ (như tường hoặc hộp khác mà không có ô trống phía sau).

**- Định nghĩa điều kiện đích:** Trạng thái đích là trạng thái mà tất cả các hộp đã được đặt vào đúng vị trí đích quy định trên bản đồ.

## **2.2.2 Biểu diễn trạng thái và các hành động**

**- Cấu trúc dữ liệu cho trạng thái:** Sử dụng một ma trận hoặc danh sách hai chiều để lưu trữ thông tin về vị trí người chơi, vị trí các hộp, các ô trống, và các bức tường.

**- Các hành động hợp lệ:** Mỗi hành động sẽ bao gồm các bước di chuyển của người chơi (lên, xuống, trái, phải). Nếu có hộp ở vị trí kế bên và ô phía sau hộp trống, người chơi có thể đẩy hộp vào ô trống đó.

**- Hành động và trạng thái kế tiếp:** Mỗi khi thực hiện một hành động, chương trình sẽ tạo ra một trạng thái mới. Trạng thái này sẽ được lưu trữ và kiểm tra xem có đạt điều kiện đích hay không.

## **2.2.3 Triển khai thuật toán DFS**

**- Khởi tạo thuật toán DFS:** Bắt đầu từ trạng thái ban đầu, DFS sẽ thực hiện các hành động khả thi từ vị trí hiện tại của người chơi và đẩy hộp khi cần.

**- Duyệt theo chiều sâu:** DFS sẽ chọn một hành động, chuyển trạng thái và tiếp tục đi sâu theo nhánh đó. Quá trình này sẽ lặp lại cho đến khi đạt trạng thái đích hoặc đạt độ sâu giới hạn.

**- Kiểm tra điều kiện chiến thắng:** Sau mỗi lần thay đổi trạng thái, kiểm tra xem các hộp đã đạt vị trí đích chưa. Nếu đạt, thuật toán sẽ dừng lại và trả về chuỗi hành động từ trạng thái khởi đầu đến trạng thái đích.

# **2.3 Mã nguồn**

## **2.3.1** **Mã nguồn**

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, màn hình

Mô tả được tạo tự động

Ảnh có chứa ảnh chụp màn hình, văn bản, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động

## **2.3.2 Kết Luận- Nhận xét**

**Kết Luận**

Trong đồ án này, chúng ta đã nghiên cứu và áp dụng thuật toán tìm kiếm theo chiều rộng (Breadth-First Search - BFS) để giải quyết bài toán SuKoBan – một trò chơi giải đố yêu cầu người chơi đẩy các hộp vào đúng vị trí đích. Qua quá trình thực hiện, từ xác định không gian trạng thái, biểu diễn trạng thái và hành động, đến triển khai và tối ưu hóa thuật toán, bài toán đã được giải quyết một cách hệ thống và hiệu quả.

Kết quả cho thấy thuật toán BFS có khả năng tìm ra giải pháp tối ưu với số bước đi ngắn nhất cho các cấp độ đơn giản và vừa phải của SuKoBan. Với BFS, việc kiểm tra toàn bộ các trạng thái ở cùng mức độ sâu trước khi mở rộng các trạng thái ở độ sâu lớn hơn giúp đảm bảo không bỏ sót bất kỳ giải pháp ngắn hơn nào. Tuy nhiên, BFS yêu cầu lưu trữ nhiều trạng thái trong bộ nhớ, đặc biệt khi không gian trạng thái của trò chơi mở rộng, điều này có thể làm tăng nhanh nhu cầu về bộ nhớ trong các cấp độ phức tạp.

**Nhận xét – Ưu điểm và Nhược điểm của phương pháp BFS:**

* **Ưu điểm:**

- BFS đảm bảo tìm ra giải pháp tối ưu về số bước đi, phù hợp với bài toán SuKoBan khi yêu cầu đưa ra lộ trình ngắn nhất để giải quyết.

- Dễ dàng triển khai và dễ hiểu, đặc biệt phù hợp cho các bài toán có không gian trạng thái không quá lớn.

**Nhược điểm:**

- BFS tiêu tốn nhiều bộ nhớ, vì tất cả các trạng thái ở cùng mức độ sâu cần được lưu trữ trong hàng đợi để duyệt.

- Khi không gian trạng thái lớn, nhu cầu bộ nhớ tăng nhanh, khiến BFS khó áp dụng hiệu quả trong các cấp độ trò chơi phức tạp hoặc các bản đồ lớn.

**CHƯƠNG 3. KẾT LUẬN**

# 3 KẾT LUẬN

## **3.1.1 Tóm tắt quá trình thực hiện:**

Báo cáo đã trình bày quá trình áp dụng thuật toán tìm kiếm theo chiều rộng (Breadth-First Search - BFS) vào bài toán SuKoBan. Các bước thực hiện bao gồm:

* **Xác định vấn đề**: Đề xuất mô hình bài toán SuKoBan dưới dạng một không gian trạng thái, trong đó người chơi phải di chuyển và đẩy các hộp vào các vị trí đích.
* **Phân tích thuật toán BFS**: Trình bày lý thuyết về thuật toán BFS và lý do tại sao nó phù hợp với bài toán SuKoBan, đặc biệt là khả năng tìm ra giải pháp tối ưu về số bước đi.
* **Triển khai thuật toán**: Cung cấp mã nguồn và các bước triển khai thuật toán BFS để giải quyết bài toán SuKoBan.
* **Đánh giá kết quả**: Phân tích hiệu quả của thuật toán, bao gồm việc so sánh số bước tìm kiếm tối ưu với các phương pháp khác.

## **3.1.2 Kết quả đạt được:**

Chương trình đã được triển khai và có thể giải quyết các cấp độ đơn giản của SuKoBan. Kết quả cho thấy thuật toán BFS có thể tìm ra chuỗi hành động tối ưu cho các trạng thái ban đầu của trò chơi, đảm bảo tìm ra đường đi ngắn nhất. Điều này chứng minh được tính khả thi và hiệu quả của việc áp dụng thuật toán BFS vào bài toán SuKoBan.

## **3.1.3 Nhận xét và hạn chế:**

* **Nhận xét**: Thuật toán BFS là một công cụ mạnh mẽ để giải quyết bài toán SuKoBan, đặc biệt khi yêu cầu tìm kiếm giải pháp tối ưu. BFS đảm bảo rằng nếu một giải pháp tồn tại, thuật toán sẽ tìm ra nó với số bước di chuyển ít nhất, giúp giải quyết bài toán một cách hiệu quả ở các cấp độ đơn giản.

**- Hạn chế**: BFS gặp phải một số hạn chế khi áp dụng cho các bài toán có không gian trạng thái lớn. Cụ thể, khi không gian trạng thái mở rộng, số lượng trạng thái phải lưu trữ và duyệt sẽ tăng rất nhanh, dẫn đến việc tiêu tốn nhiều bộ nhớ và làm giảm hiệu suất của thuật toán. Do đó, BFS chỉ thực sự hiệu quả với các bài toán có không gian trạng thái không quá lớn.

TÀI LIỆU THAM KHẢO

[1] Hoang S, Nguyen K, Huynh T, et al., “Chest X-ray Severity Score as a Putative  
Predictor of Clinical Outcome in Hospitalized Patients: An Experience From a Vietnamese Covid-19 Field Hospital”, 10.7759/cureus.23323, 2022.

[2] L. J. Muhammad, Ebrahem A. Algehyne, et al., “Supervised Machine Learning  
Models for Prediction of COVID-19 Infection using Epidemiology Dataset”, Advances in Computational Approaches for Artificial Intelligence, Image Processing, IoT and Cloud Applications, 2020.

[3] Zoabi, Y., Deri-Rozov, S. & Shomron, N, “Machine learning-based prediction of COVID-19 diagnosis based on symptoms”, 2021.