|  |  |
| --- | --- |
| **bỘ GIÁO DỤC VÀ ĐÀO TẠO** | **BỘ NÔNG NGHIỆP VÀ PTNT** |

TRƯỜNG ĐẠI HỌC THỦY LỢI



ngô việt hoàng

GIẢI THUẬT tối ưu hóa ĐÀN KIẾN ĐỂ LẬP LỊCH ĐƯỜNG ĐI cho ROBOT DI ĐỘNG TRONG MÔI TRƯỜNG TĨNH CÓ CHƯỚNG NGẠI VẬT

ĐỒ ÁN TỐT NGHIỆP

HÀ NỘI, NĂM 2021

|  |  |
| --- | --- |
| **bỘ GIÁO DỤC VÀ ĐÀO TẠO** | **BỘ NÔNG NGHIỆP VÀ PTNT** |

TRƯỜNG ĐẠI HỌC THỦY LỢI

NGÔ VIỆT HOÀNG

GIẢI THUẬT TỐI ƯU HÓA ĐÀN KIẾN ĐỂ LẬP LỊCH ĐƯỜNG ĐI CHO ROBOT DI ĐỘNG TRONG MÔI TRƯỜNG TĨNH CÓ CHƯỚNG NGẠI VẬT

|  |  |
| --- | --- |
| Ngành : | Công nghệ thông tin |
| Mã số: |  |

|  |  |
| --- | --- |
| NGƯỜI HƯỚNG DẪN | 1. ThS. Trần Thị Cẩm Giang |

HÀ NỘI, NĂM 2021

**NGÔ VIỆT HOÀNG ĐỒ ÁN TỐT NGHIỆP HÀ NỘI, NĂM 2021 ……**

|  |  |
| --- | --- |
| Logo-WRU | CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM  **Độc lập  - Tự do  - Hạnh phúc**  ----------★----------  **NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP** |

**Họ tên sinh viên:** Ngô Việt Hoàng **Hệ đào tạo**: Đại học chính quy

**Lớp**: 59TH3 **Ngành**: Công nghệ thông tin

**Khoa**: Công nghệ thông tin

1- TÊN ĐỀ TÀI:

GIẢI THUẬT TỐI ƯU HÓA ĐÀN KIẾN ĐỂ LẬP LỊCH ĐƯỜNG ĐI CHO ROBOT DI ĐỘNG TRONG MÔI TRƯỜNG TĨNH CÓ CHƯỚNG NGẠI VẬT.

2- CÁC TÀI LIỆU CƠ BẢN:

|  |  |
| --- | --- |
| [1] | Alatortsev, S., Stellmacher, S., & Ortmeier, F. (2015). Robotic task sequencing problem: A survey. *Journal of intelligent & robotic systems*, *80*(2), 279-298. |
| [2] | Han-ye Zhang, Wei-ming Lin, and Ai-xia Chen. Path planning for the mobile robot: A review. Symmetry, 10(10):450, 2018. |
| [3] | Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. *IEEE computational intelligence magazine*, *1*(4), 28-39. |
| [4] | Canny, J. (1988). *The complexity of robot motion planning*. MIT press. |
| [5] | Ajeil, F. H., Ibraheem, I. K., Azar, A. T., & Humaidi, A. J. (2020). Grid-based mobile robot path planning using aging - based ant colony optimization algorithm in static and dynamic environments. *Sensors*, *20*(7), 1880. |
| [6] | Ning, J., Zhang, C., Sun, P., & Feng, Y. (2019). Comparative study of ant colony algorithms for multi-objective optimization. *Information*, 10(1), 11. |
| [7] | Sariff, N., & Buniyamin, N. (2006). An Overview of Autonomous Mobile Robot Path Planning Algorithms. *2006 4th Student Conference on Research and Development*, 183-188. |
| [8] | Wikipedia contributors. Motion planning Wikipedia, the free encyclopedia, 2021. [Online; accessed 11-June-2021] |
| [9] | Cai, K., Wang, C., Cheng, J., De Silva, C. W., & Meng, M. Q. H. (2020). Mobile Robot Path Planning in Dynamic Environments: A Survey. *arXiv preprint arXiv:2006.14195.* |
| [10] | BK Patle, Anish Pandey, DRK Parhi, A Jagadeesh, et al. A review: On path planning strategies for navigation of mobile robot. Defence Technology, 15(4):582606, 2019. |
| [11] | Trần Thị Cẩm Giang (2020). *Trí tuệ nhân tạo* (Curriculum). |
| [12] | Khaled Akka and Farid Khaber (2018). Mobile robot path planning using an improved ant colony optimization. *International Journal of Advanced Robotic Systems*. 1-7. |
| [13] | D R Parhi and J K Pothal (2011). Intelligent navigation of multiple mobile robots using an ant colony optimization technique in a highly cluttered environment. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*. |

3 - NỘI DUNG CÁC PHẦN THUYẾT MINH VÀ TÍNH TOÁN:

|  |  |
| --- | --- |
| **Nội dung các phần** | **Tỷ lệ** |
| Chương 1: Cơ sở lý thuyết | 30% |
| Chương 2: Bài toán lập kế hoạch đường đi cho robot di động trong môi trường tĩnh có chướng ngại vật | 20% |
| Chương 3: Đề xuất giải pháp cải tiến thuật toán đàn kiến kết hợp thuật toán tìm kiếm theo chiều rộng để giải bài toán lập kế hoạch đường đi cho robot di động | 30% |
| Chương 4: Kết quả thử nghiệm | 20% |

4. GIÁO VIÊN HƯỚNG DẪN TỪNG PHẦN

|  |  |
| --- | --- |
| **Phần** | **Họ tên giáo viên hướng dẫn** |
| Chương 1: Cơ sở lý thuyết | ThS. Trần Thị Cẩm Giang |
| Chương 2: Bài toán lập kế hoạch đường đi cho robot di động trong môi trường tĩnh có chướng ngại vật | ThS. Trần Thị Cẩm Giang |
| Chương 3: Đề xuất giải pháp cải tiến thuật toán đàn kiến kết hợp thuật toán tìm kiếm theo chiều rộng để giải bài toán lập kế hoạch đường đi cho robot di động | ThS. Trần Thị Cẩm Giang |
| Chương 4: Kết quả thử nghiệm | ThS. Trần Thị Cẩm Giang |

5. NGÀY GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Ngày ............  tháng .........  năm 2020

|  |  |
| --- | --- |
| **Trưởng Bộ môn**  *(Ký và ghi rõ Họ tên)* | **Giáo viên hướng dẫn chính**  *(Ký và ghi rõ Họ tên)* |

  ThS. Trần Thị Cẩm Giang

Nhiệm vụ Đồ án tốt nghiệp đã được Hội đồng thi tốt nghiệp của Khoa thông qua

                                           Ngày. . . . .tháng. . . . .năm 2020                                                       **Chủ tịch Hội đồng**

*(Ký và ghi rõ Họ tên)*

Sinh viên đã hoàn thành và nộp bản Đồ án tốt nghiệp cho Hội đồng thi ngày...  tháng... năm 2020

**Sinh viên làm Đồ án tốt nghiệp**

*(Ký và ghi rõ Họ tên)*

     Ngô Việt Hoàng

|  |  |
| --- | --- |
| logo | TRƯỜNG ĐẠI HỌC THUỶ LỢI  **KHOA CÔNG NGHỆ THÔNG TIN**    **BẢN TÓM TẮT ĐỀ CƯƠNG ĐỒ ÁN TỐT NGHIỆP** |

**TÊN ĐỀ TÀI: Giải thuật tối ưu hóa đàn kiến để lập lịch đường đi cho Robot di động trong môi trường tĩnh có chướng ngại vật.**

*Sinh viên thực hiện*:   Ngô Việt Hoàng

*Lớp*:  59TH3

*Giáo viên hướng dẫn*:   ThS. Trần Thị Cẩm Giang

**TÓM TẮT ĐỀ TÀI**

Bước vào cuộc cách mạng công nghiệp lần thứ tư, khoa học công nghệ tiếp tục cho thấy sự phát triển nhanh chóng mạnh mẽ. Sức lao động của con người dần được thay thế bằng thế bằng những robot di động (mobile robot). Tìm kiếm đường đi cho robot thì cần phải có kế hoạch. Vì vậy, việc lập kế hoạch đường đi cho robot đã trở thành một trong những đề tài dành được nhiều sự chú ý của các nhà nghiên cứu. Đặc biệt, bài toán lập kế hoạch đường đi cho robot di động trong môi trường tĩnh có chướng ngại vật cũng là bài toán được quan tâm trong việc tạo ra các robot di động. Trong đồ án này, tôi cải tiến giải thuật tối ưu hóa đàn kiến trong môi trường tĩnh 2D có chướng ngại vật.

Giải thuật tối ưu đàn kiến (Ant Colony Optimization - ACO) là cách tiếp cận metaheuristic được Marco Dorigo giới thiệu vào năm 1991 và liên tục được phát triển cho đến nay. Giải thuật được lấy cảm hứng từ cách loài kiến di chuyển tìm kiếm thức ăn trong thực tế. Trong thế giới tự nhiên, những con kiến di chuyển và sẽ tiết ra một chất hóa học có tên là pheromone, lần theo vết pheromone này các con kiến khác trong đàn có thể tìm đến chỗ thức ăn, vì vậy thuật toán tối ưu đàn kiến là một loại thuật toán tối ưu hóa phỏng sinh học. Thành công đầu tiên của các thuật toán ACO là giải quyết bài toán Người chào hàng nổi tiếng với số đỉnh lên tới hơn 2000 với kết quả thu được là tốt, hiệu quả của nó được chứng minh bằng thực nghiệm. Để giải quyết hiệu quả bài toán lập kế hoạch đường đi cho robot di động trong môi trường tránh chướng ngại vật, trong khuôn khổ đồ án này, tôi sử dụng thuật toán tối ưu hóa đàn kiến ​​cải tiến.

Trong phạm vi đề tài này, tôi thực hiện nghiên cứu giải thuật đàn kiến để tìm đường đi tối ưu nhất trong môi trường tĩnh có chướng ngại vật. Để hoàn thành đề tài, tôi đã tham khảo bài báo khoa học: "**Mobile robot path planning using an improved ant colony optimization**". Đồng thời tìm hiểu và cải tiến thuật toán giải quyết các trường hợp khó hơn và đem lại hiệu quả cao hơn trong việc tìm đường đi tối ưu nhất.

**CÁC MỤC TIÊU CHÍNH**

Tìm hiểu rõ về giải thuật tối ưu đàn kiến (ACO) để giải các bài toán lập kế hoạch di chuyển cho robot di động.

Cài đặt môi trường và sử dụng giải thuật tối ưu đàn kiến (ACO) để tìm được đường đi tối ưu để giải quyết vấn đề bài toán.

Tìm hiểu và phân tích những trường hợp chưa tối ưu và đưa ra phương hướng giải quyết.

Cài đặt và thử nghiệm thuật toán lập kế hoạch đường đi cho robot dựa trên thuật toán tối ưu hóa đàn kiến (ACO) bằng ngôn ngữ lập trình Java.

**KẾT QUẢ DỰ KIẾN**

Sử dụng thuật toán tối ưu hóa đàn kiến và ngôn ngữ lập trình Java để tìm được đường đi tối ưu cho robot di động trong môi trường tĩnh 2D có chướng ngại vật thỏa mãn được các tiêu chí đã nêu ở trên.

LỜI CAM ĐOAN

Tác giả xin cam đoan đây là Đồ án tốt nghiệp của bản thân tác giả. Các kết quả trong Đồ án tốt nghiệp này là trung thực, và không sao chép từ bất kỳ một nguồn nào và dưới bất kỳ hình thức nào. Việc tham khảo các nguồn tài liệu (nếu có) đã được thực hiện trích dẫn và ghi nguồn tài liệu tham khảo đúng quy định.

|  |  |
| --- | --- |
|  | **Tác giả ĐATN**  *Chữ ký*  **Ngô Việt Hoàng** |

LỜI CÁM ƠN

Em xin bày tỏ lời cảm ơn chân thành tới tập thể các thầy cô giáo trường Đại học Thủy Lợi đã dạy dỗ chúng em trong suốt quá trình học tập chương trình cao học tại trường.

Đặc biệt em xin bày tỏ lòng biết ơn sâu sắc tới cô giáo ThS. Trần Thị Cẩm Giang, Trường Đại học Thủy Lợi đã quan tâm, định hướng và đưa ra những góp ý, gợi ý, chỉnh sửa quý báu cho em trong quá trình làm đồ án tốt nghiệp.

Cuối cùng, em xin chân thành cảm ơn các bạn bè đồng nghiệp, gia đình và người thân đã quan tâm, giúp đỡ và chia sẻ với em trong suốt quá trình làm đồ án tốt nghiệp.

*Em xin chân thành cảm ơn!*

PHỤ LỤC

[PHỤ LỤC](#_Toc92276485)

[DANH MỤC CÁC HÌNH ẢNH](#_Toc92276486)

[DANH MỤC BẢNG BIỂU](#_Toc92276487)

[DANH MỤC CÁC TỪ VIẾT TẮT VÀ GIẢI THÍCH CÁC THUẬT NGỮ](#_Toc92276488)

[MỞ ĐẦU 1](#_Toc92276489)

[CHƯƠNG 1 CƠ SỞ LÝ THUYẾT 4](#_Toc92276490)

[1.1 Robot di động 4](#_Toc92276491)

[1.1.1 Robot di động là gì? 4](#_Toc92276492)

[1.1.2 Ứng dụng của robot di động 4](#_Toc92276493)

[1.1.3 Những thách thức cần vượt qua 6](#_Toc92276494)

[1.2 Mô hình hoá môi trường 7](#_Toc92276495)

[1.3 Thuật toán tìm kiếm 9](#_Toc92276496)

[1.3.1 Chiến lược tìm kiếm không có thông tin 10](#_Toc92276497)

[1.3.2 Chiến lược tìm kiếm có thông tin 10](#_Toc92276498)

[1.4 Thuật toán tìm kiếm theo chiều rộng (Breadth First Search – BFS) 11](#_Toc92276499)

[1.5 Thuật toán tối ưu hoá đàn kiến (Ant Colony Optimization - ACO) 12](#_Toc92276500)

[1.5.1 Giới thiệu chung 12](#_Toc92276501)

[1.5.2 Thí nghiệm trên cây cầu đôi 15](#_Toc92276502)

[1.5.3 Tối ưu hóa đàn kiến tiêu chuẩn 17](#_Toc92276503)

[1.5.4 Các yếu tố quyết định đến hiệu quả của thuật toán 19](#_Toc92276504)

[1.5.5 So sánh thuật toán ACO với các thuật toán tối ưu khác 20](#_Toc92276505)

[1.5.6 Ưu điểm và nhược điểm của thuật toán ACO 20](#_Toc92276506)

[CHƯƠNG 2 BÀI TOÁN LẬP KẾ HOẠCH ĐƯỜNG ĐI CHO ROBOT DI ĐỘNG TRONG MÔI TRƯỜNG TĨNH CÓ CHƯỚNG NGẠI VẬT 21](#_Toc92276507)

[2.1 Tổng quan bài toán 21](#_Toc92276508)

[2.1.1 Giới thiệu bài toán 21](#_Toc92276509)

[2.1.2 Phân loại bài toán 24](#_Toc92276510)

[2.1.3 Định hướng bài toán 25](#_Toc92276511)

[2.2 Phát biểu bài toán 28](#_Toc92276512)

[2.3 Các nghiên cứu liên quan 30](#_Toc92276513)

[CHƯƠNG 3 ĐỀ XUẤT GIẢI PHÁP CẢI TIẾN THUẬT TOÁN ĐÀN KIẾN KẾT HỢP THUẬT TOÁN TÌM KIẾM THEO CHIỀU RỘNG ĐỂ GIẢI BÀI TOÁN LẬP KẾ HOẠCH ĐƯỜNG ĐI CHO ROBOT DI ĐỘNG 31](#_Toc92276514)

[3.1 Đề xuất giải pháp cải tiến thuật toán đàn kiến kết hợp thuật toán tìm kiếm theo chiều rộng (ACO-BFS) 31](#_Toc92276515)

[3.2 Tối ưu hóa đàn kiến sử dụng xác suất kích thích (Ant Colony Optimization using Stimulating Probability – ACO-SP) 34](#_Toc92276516)

[3.3 Kết hợp thuật toán ACO-BFS với ACO-SP (ACO-BFS-SP) 35](#_Toc92276517)

[3.4 Tối ưu hóa đàn kiến thông qua điều hướng mục tiêu (Ant Colony Optimization Navigation - ACO-N) 36](#_Toc92276518)

[3.5 Cải tiến thuật toán ACO-BFS-SP dựa trên ACO-N (ACO-BFS-SP-N) 37](#_Toc92276519)

[CHƯƠNG 4 CÀI ĐẶT THỰC NGHIỆM 39](#_Toc92276520)

[4.1 Mô hình hóa môi trường dựa trên lưới 39](#_Toc92276521)

[4.2 Các hàm mục tiêu 41](#_Toc92276522)

[4.3 Cài đặt thực nghiệm 41](#_Toc92276523)

[4.4 Dữ liệu thực nghiệm 41](#_Toc92276524)

[4.5 Kết quả thực nghiệm 44](#_Toc92276525)

[4.5.1 Thực nghiệm 1: Phân tích và đánh giá kết quả đối với sự thay đổi số lượng chướng ngại vật của giải thuật ACO-BFS-SP-N so với giải thuật ACO, ACO-BFS và ACO-BFS-SP trên các bộ dữ liệu khác nhau. 44](#_Toc92276526)

[4.5.2 Thực nghiệm 2: Phân tích sự ảnh hưởng hình dạng chướng ngại vật lên kết quả trong giải thuật ACO-BFS-SP-N 49](#_Toc92276527)

[4.5.3 Thực nghiệm 3: Phân tích và đánh giá kết quả của giải thuật ACO-BFS-SP-N trong các bộ dữ liệu mô phỏng không gian trong phòng thí nghiệm 54](#_Toc92276528)

[Kết luận 59](#_Toc92276529)

[Tài liệu tham khảo 61](#_Toc92276530)

DANH MỤC CÁC HÌNH ẢNH

[Hình 1.1. Robot xe tự vận hành 5](#_Toc92276558)

[Hình 1.2. Robot tự vận hành thăm dò sao Hỏa của NASA 5](#_Toc92276559)

[Hình 1.3. Robot thay nhân viên y tế giao thuốc và nhu yếu phẩm cho bệnh nhân trong tình hình dịch Covid 6](#_Toc92276560)

[Hình 1.4. Phân loại các phương pháp mô hình hóa môi trường 7](#_Toc92276561)

[Hình 1.5. Các phương pháp tiếp cận theo khung không gian 8](#_Toc92276562)

[Hình 1.6. Phương pháp lộ trình xác suất 9](#_Toc92276563)

[Hình 1.7. Các loại thuật toán tìm kiếm có thông tin 10](#_Toc92276564)

[Hình 1.8. Đàn kiến kiếm ăn 12](#_Toc92276565)

[Hình 1.9. Mô tả tổng quan về kiến và pheromone 13](#_Toc92276566)

[Hình 1.10. Thí nghiệm trên cây cầu đôi 16](#_Toc92276567)

[Hình 1.11. Thí nghiệm ban đầu chỉ một nhánh dài và sau 30 phút thêm nhánh ngắn 17](#_Toc92276568)

[Hình 2.1 Sơ đồ các phương pháp giải bài toán lập lịch đường đi cho robot di động 24](#_Toc92276569)

[Hình 2.2. Phân loại bài toán lập kế hoạch đường đi cho robot 25](#_Toc92276570)

[Hình 2.3. Định hướng giải quyết bài toán lập kế hoạch đường đi cho robot trong môi trường [8] 26](#_Toc92276571)

[Hình 2.4. Một số phương pháp mô hình hóa môi trường phân rã ô [10] 27](#_Toc92276572)

[Hình 2.5. Mô tả môi trường tĩnh đã biết trước chướng ngại vật 29](#_Toc92276573)

[Hình 3.1. Đường đi trước và sau khi loại bỏ các đoạn đường không cần thiết 31](#_Toc92276574)

[Hình 4.1. Môi trường lưới [5] 39](#_Toc92276575)

[Hình 4.2. Vị trí có thể di chuyển của robot trong môi trường lưới [5] 40](#_Toc92276576)

[Hình 4.3. Tọa độ vị trí mà kiến có thể di chuyển trong môi trường 40](#_Toc92276577)

[Hình 4.4. Hình minh họa kết quả thực nghiệm 1:  Phân tích và đánh giá kết quả của thuật toán ACO-BFS-SP-N đối với thuật toán ACO, ACO-BFS và ACO-BFS-SP trên các bộ dữ liệu khác nhau. 46](#_Toc92276578)

[Hình 4.5. Hình minh họa kết quả thực nghiệm 2: phân tích sự ảnh hưởng hình dạng chướng ngại vật lên kết quả giải thuật ACO-BFS-SP-N 51](#_Toc92276579)

[Hình 4.6. Hình minh họa kết quả thực nghiệm 3: Phân tích và đánh giá kết quả của giải thuật ACO-BFS-SP-N trong các bộ dữ liệu mô phỏng không gian trong phòng thí nghiệm 55](#_Toc92276580)

DANH MỤC BẢNG BIỂU

[Bảng 1.1. Thuật toán tìm kiếm theo chiều rộng 12](#_Toc92276615)

[Bảng 1.2. Thuật toán tối ưu đàn kiến 19](#_Toc92276616)

[Bảng 1.3. Ưu điểm và nhược điểm của thuật toán ACO [12] [16]. 20](#_Toc92276617)

[Bảng 2.1. Phát biểu bài toán lập kế hoạch đường đi cho robot trong môi trường tĩnh đã biết trước chướng ngại vật 29](#_Toc92276618)

[Bảng 3.1. Thuật toán ACO-BFS 33](#_Toc92276619)

[Bảng 3.2. Ưu điểm và nhược điểm của thuật toán ACO-BFS 33](#_Toc92276620)

[Bảng 3.3. Ưu điểm và nhược điểm của ACO-SP: 35](#_Toc92276621)

[Bảng 3.4. Thuật toán ACO-BFS-SP 36](#_Toc92276622)

[Bảng 4.1. Các tiêu chí đánh giá 42](#_Toc92276623)

[Bảng 4.2. Chú thích giao diện khi chạy thực nghiệm 43](#_Toc92276624)

[Bảng 4.3. Tham số cài đặt trên từng bộ dữ liệu được cài đặt trong thực nghiệm 1 44](#_Toc92276625)

[Bảng 4.4. Kết quả thực nghiệm 1: phân tích và đánh giá kết quả của thuật toán ACO-BFS-SP-N đối với thuật toán ACO, ACO-BFS và ACO-BFS-SP trên các bộ dữ liệu khác nhau. 47](#_Toc92276626)

[Bảng 4.5. Tham số cài đặt trên từng bộ dữ liệu được cài đặt trong thực nghiệm 2 49](#_Toc92276627)

[Bảng 4.6. Kết quả thực nghiệm 2: phân tích sự ảnh hưởng hình dạng chướng ngại vật lên kết quả giải thuật ACO-BFS-SP-N 52](#_Toc92276628)

[Bảng 4.7. Tham số cài đặt trên từng bộ dữ liệu được cài đặt trong thực nghiệm 3 54](#_Toc92276629)

[Bảng 4.8. Kết quả thực nghiệm 3: phân tích và đánh giá kết quả của giải thuật 56](#_Toc92276630)

[ACO-BFS-SP-N trong các bộ dữ liệu mô phỏng không gian phòng thí nghiệm 56](#_Toc92276631)

DANH MỤC CÁC TỪ VIẾT TẮT VÀ GIẢI THÍCH CÁC THUẬT NGỮ

|  |  |  |
| --- | --- | --- |
| Từ viết tắt | Viết đầy đủ | Ý nghĩa |
| ACO | Ant Colony Optimization | Tối ưu hóa đàn kiến |
| BFS | Breadth First Search | Tìm kiếm theo chiều rộng |
| ACO-BFS | Ant Colony Optimization Breadth First Search | Tối ưu hóa đàn kiến kết hợp thuật toán tìm kiếm theo chiều rộng |
| ACO-SP | Ant Colony Optimization using Stimulating Probability | Tối ưu hóa đàn kiến sử dụng xác suất kích thích |
| ACO-N | Ant Colony Optimization Navigation | Tối ưu hóa đàn kiến thông qua điều hướng mục tiêu |
| ACO-BFS-SP | Ant Colony Optimization Breadth First Search using Stimulating Probability | Tối ưu hóa đàn kiến kết hợp giải thuật tìm kiếm theo chiều rộng sử dụng xác suất kích thích |
| ACO-BFS-SP-N | Ant Colony Optimization Breadth First Search using Stimulating Probability Navigation | Tối ưu hóa đàn kiến thông qua điều hướng mục tiêu kết hợp thuật toán tìm kiếm theo chiều rộng sử dụng xác suất kích thích |

MỞ ĐẦU

Trong thời đại cách mạng công nghiệp, những thành tựu về khoa học công nghệ đã được ứng dụng trong nhiều lĩnh vực khác nhau giúp cho việc sản xuất đạt năng suất cao bằng việc sử dụng các công cụ hỗ trợ, thay thế sức lao động con người. Một trong số những thành tựu trong sự tiến bộ vượt bậc của công nghệ là robot. Robot thay con người làm những công việc có hại cho sức khỏe, giúp tăng năng suất công việc. Robot không chỉ xuất hiện trong các khu công nghiệp, robot còn phổ biến trong nhiều gia đình để làm những công việc đơn giản như: lau dọn, cắt cỏ,… Robot di động tự động (mobile robot) là sự kết hợp robot với trí tuệ nhân tạo (Artificial Intelligence – AI), có khả năng di chuyển tự động trong môi trường. Robot di động tự động đã giúp giải phóng sức lao động cho con người trong nhiều lĩnh vực khác nhau như công nghiệp, nông nghiệp, thăm dò, cứu hộ cứ nạn,… Nhưng do sự hạn chế của robot làm ảnh hưởng lớn đến hành trình di chuyển và nhiệm vụ của robot. Vì vậy, bài toán tối ưu hóa sự hạn chế cho robot luôn là một bài toán nhận được nhiều sự quan tâm của các nhà nghiên cứu. Một trong các bài toán tối ưu hóa quan trọng cho robot chính là bài toán tối ưu tìm đường đi cho robot di động. Đây là bài toán thuộc lớp bài toán NP-khó.

Để giải quyết các bài toán thuộc dạng NP-khó, nhiều các giải thuật khác nhau đã được đề xuất để có thể tìm lời giải tối ưu như: thuật toán tham lam, thuật toán di truyền, thuật toán tối ưu hóa bầy đàn,… Tuy nhiên, các giải thuật trên đều tốn chi phí về thời gian hoặc không gian lớn.

Tối ưu hóa đàn kiến (Ant Colony Optimization - ACO) là cách tiếp cận metaheuristic tương đối mới, do Marco Dorigo giới thiệu vào năm 1991 và liên tục được phát triển cho đến nay. Thành công đầu tiên của các thuật toán ACO là giải quyết bài toán Người chào hàng nổi tiếng với số đỉnh lên tới hơn 2000 với kết quả thu được là tốt, hiệu quả của nó được chứng minh bằng thực nghiệm.

Từ đó trong đề tài này, đề xuất giải thuật đàn kiến (Ant Colony Optimization - ACO) vào bài toán tối ưu hóa tìm đường đi cho robot trong môi trường tĩnh có chướng ngại vật để đạt được tiêu chí: robot vượt qua các chướng ngại vật di chuyển từ điểm xuất phát đến điểm đích, đường đi tìm được phải là đường đi ngắn nhất và tốt nhất.

Đầu tiên, đề tài đã hệ thống các kiến thức cơ sở về lý thuyết độ phức tạp thuật toán, thách thức và ứng dụng của robot di động. Sau đó, đề tài trình bày về giải thuật tìm kiếm theo chiều rộng, giải thuật tối ưu đàn kiến (Ant Colony Optimization - ACO) và bài toán tìm đường đi cho robot trong môi trường tĩnh có chướng ngại vật. Tiếp theo, đề xuất giải pháp cải tiến thuật toán đàn kiến (Ant Colony Optimization - ACO) và giải thuật tối ưu hóa đàn kiến sử dụng xác suất kích thích (ACO-SP), giải thuật tối ưu hóa đàn kiến thông qua điều hướng (ACO-N) và so sánh kết quả thu được với giải thuật đàn kiến (Ant Colony Optimization - ACO) nhằm rút ra được ưu điểm và nhược điểm của thuật toán.

Nội dung chính của đề tài được chia thành 4 phần như sau:

**Chương 1:  Cơ sở lý thuyết**

Tìm hiểu tổng quan kiến thức cơ sở về robot di động, mô hình hóa môi trường, thuật toán tìm kiếm, trình bày chi tiết vềgiải thuật đàn kiến (Ant Colony Optimization - ACO).

**Chương 2: Bài toán lập kế hoạch đường đi cho robot di động trong môi trường tĩnh có chướng ngại vật**

Giới thiệu bài toán lập kế hoạch đường đi robot, định hướng giải quyết bài toán và các nghiên cứu liên quan.

**Chương 3: Đề xuất giải pháp cải tiến thuật toán đàn kiến kết hợp thuật toán tìm kiếm theo chiều rộng để giải bài toán lập kế hoạch đường đi cho robot di động**

Giới thiệu, trình bày chi tiết về phương pháp cải tiến giải thuật đàn kiến kết hợp giải thuật tìm kiếm theo chiều rộng, giải thuật tối ưu đàn kiến sử dụng xác suất kích thích (ACO-SP) và giải thuật tối ưu hóa đàn kiến thông qua điều hướng mục tiêu (ACO-N) và giải thuật cải tiến kết hợp.

**Chương 4: Cài đặt thực nghiệm**

Trình bày về nội dung phương pháp phân rã môi trường dạng ô lưới, kết quả thực nghiệm, đưa ra đánh giá, nhận xét.

**Kết luận.**

**Tài liệu tham khảo.**

CHƯƠNG 1 CƠ SỞ LÝ THUYẾT

1.1 Robot di động

1.1.1 Robot di động là gì?

Cụm từ “Robot di động” không còn quá xa lạ với chúng ta trong thời đại cách mạng công nghiệp lần thứ tư. Trong tương lai, mỗi người sẽ có nhu cầu sử dụng một robot như cần một máy tính cá nhân, lúc đó robot sẽ trở thành tâm điểm của một cuộc cách mạng công nghệ lớn sau Internet.

Robot là một loại thiết bị có thể thực hiện những công việc một cách tự động bằng sự điều khiển của máy tính hoặc chương trình được lập trình sẵn. Robot di động là robot có khả năng di chuyển quay hoặc tịnh tiến theo một hay nhiều chiều một cách khéo léo trong môi trường. Robot di động hoạt động một cách tự động theo những trình tự đã được lập trình trước và có thể điều khiển được bằng các lệnh thay đổi tùy theo yêu cầu của người sử dụng, có khả năng nhận biết môi trường xung quanh, tương tác với những vật thể trong môi trường. Từ đó, robot di động có thể đưa ra các lựa chọn dựa trên môi trường [1].

1.1.2 Ứng dụng của robot di động

Ngày nay, robot di động đang được sử dụng trong nhiều lĩnh vực của cuộc sống. Chúng ta có thể bắt gặp sự xuất hiện của các robot dọn dẹp nhà cửa như robot hút bụi, robot cắt cỏ,… trong sinh hoạt đời sống. Robot di động cũng được áp dụng trong nhiều ngành nghề khác nhau. Ở các bệnh viện sử dụng robot di động tự động để chào đón, hướng dẫn và giải đáp các thắc mắc của bệnh nhân hay di chuyển vật liệu y tế. Tại các khu công nghiệp, robot được lắp đặt để di chuyển và phân phối vật tư, hàng hóa từ các kệ hàng đến khu vực thực hiện đơn hàng một cách hiệu quả (*Hình 1.1*).



Hình 1.1. Robot xe tự vận hành

Trong các nhà máy, robot giúp giảm sức lao động con người và tăng cao năng suất lao động lên rất nhiều, chúng thay thế cho con người trong nhiều công việc đòi hỏi sự chính xác cao. Điều đó cho thấy sức ảnh hưởng của robot trong thời đại công nghiệp 4.0. Ngoài ra, robot còn thay thế con người làm những công việc có tính chất khó khăn, nặng nhọc hoặc những công việc có môi trường làm việc nguy hiểm, độc hại ảnh hưởng đến sức khỏe con người. Bên cạnh đó, robot cũng hỗ trợ con người thăm dò, tiếp cận những môi trường có khí hậu, thời tiết khắc nghiệt và địa hình khó di chuyển (vùng rừng núi sâu, sa mạc, vùng sâu đáy đại dương, nơi có động đất, khu vực cháy rừng, vùng nhiễm phóng xạ, khu vực bị khủng bố...) (*Hình 1.2*).



Hình 1.2. Robot tự vận hành thăm dò sao Hỏa của NASA

Đặc biệt, khi tình hình dịch Covid-19 có diễn biến phức tạp như hiện nay, nhiều quốc gia trên thế giới đã triển khai robot di động để phòng dịch, hỗ trợ các nhân viên y tế. Nhiệm vụ của chúng là hỗ trợ đo thân nhiệt; vận chuyển thức ăn, thuốc men, nhu yếu phẩm,… cho người bệnh. Từ đó tạo điều kiện để các nhân viên y tế tập trung thời gian, công sức chăm sóc và điều trị cho bệnh nhân(*Hình 1.3*).



Hình 1.3. Robot thay nhân viên y tế giao thuốc và nhu yếu phẩm cho bệnh nhân trong tình hình dịch Covid

1.1.3 Những thách thức cần vượt qua

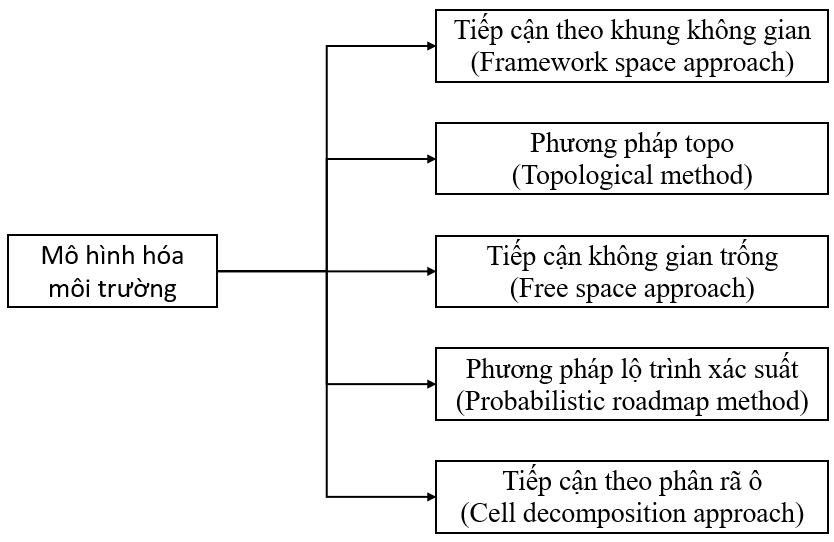
Bên cạnh những thành tựu nghiên cứu đã đạt được và nhu cầu sử dụng robot di động trong đời sống. Robot di động vẫn còn nhiều hạn chế và thách thức phải kể đến như vấn đề thao tác của robot, phương thức để robot có thể thể nhận biết được hành vi và đưa ra quyết định trong những tình huống bất ngờ… Ngoài việc cần phải khéo léo và hoàn thiện hơn, robot di động còn phải đối mặt với vấn đề hao mòn năng lượng khi di chuyển. Do robot sử dụng một nguồn năng lượng hữu hạn, robot cần phải di chuyển trên một đường đi ngắn, tối ưu nhất để tránh tiêu hao quá nhiều năng lượng. Vì vậy, lập lịch đường đi cho robot di động trở thành vấn đề ảnh hưởng trực tiếp đến hiệu quả hoạt động của robot.

1.2 Mô hình hoá môi trường

Khi robot di chuyển trong môi trường có chướng ngại vật biết trước, chúng cần xây dựng một hay nhiều đường đi khả thi, không có va chạm, sau đó robot sẽ di chuyển theo đường đi đấy để đến được vị trí mục tiêu. Trước khi bắt đầu điều hướng di chuyển, các thông tin của môi trường cần được xác định để đường đi tránh các chướng ngại vật và không xảy ra va chạm.

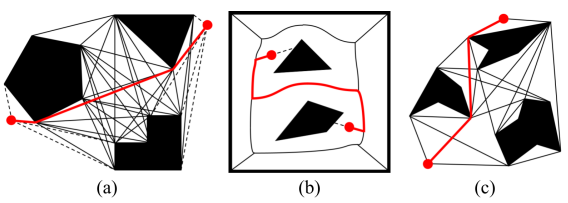
Mô hình hóa môi trường hay phân rã môi trường là việc dựa trên các thông tin môi trường làm việc của robot trong không gian thực tế đã biết trước để xây dựng một đồ thị hay một bản đồ di chuyển cho robot. Một môi trường được phân rã phù hợp sẽ giúp giảm số lần tính toán và giảm các thao tác không cần thiết của robot. Mô hình hóa môi trường là một trong những vấn đề đang được nhiều nhà nghiên cứu chú ý trong lĩnh vực phát triển robot, nó ảnh hưởng rất nhiều trong việc lập lịch đường đi cho robot [2].

Đã có rất nhiều bài báo nghiên cứu về các phương pháp mô hình hóa môi trường. Trong đó, các phương pháp mô hình hóa môi trường phổ biến được trình bày như *Hình 1.4*:



Hình 1.4. Phân loại các phương pháp mô hình hóa môi trường

Phương pháp tiếp cận theo khung không gian (Framework space approach) là một phương pháp cổ điển, gồm đồ thị hiển thị (visibility graph - *Hình 1.5.a*), đồ thị Voronoi (*Hình 1.5.b*), đồ thị tiếp tuyến (tangent graph - *Hình 1.5.c*),... Đặc điểm chung của các đồ thị này là được xây dựng dựa trên tọa độ điểm xuất phát, điểm đích và các đỉnh của chướng ngại vật. Các đồ thị này coi robot chỉ là một điểm, robot có thể di chuyển tự do sát với chướng ngại vật mà không xảy ra va chạm. Nhưng trong thực tế, robot có kích thước tương đối với nhiều bộ phận khác nhau như cánh tay, ăng-ten,… Vì vậy, việc coi robot là một điểm trở nên không khả thi. Ngoài ra, việc di chuyển sát các đỉnh chướng ngại vật làm giảm độ an toàn của đường đi, gây va chạm. Do đó, hiện nay các phương pháp mô hình hóa môi trường trên không còn được sử dụng nhiều nữa.

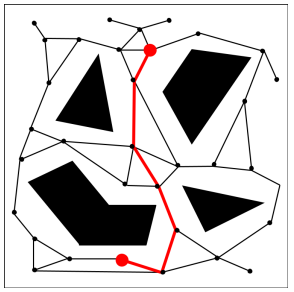


Hình 1.5. Các phương pháp tiếp cận theo khung không gian

Phương pháp topo (Topological method) lập lịch đường đi trong môi trường đa chiều bằng cách giảm số chiều tạo thành mạng topology. Nhược điểm của phương pháp này là chỉ phù hợp với môi trường có đặc điểm rõ ràng và thưa thớt chướng ngại vật. Khi số lượng chướng ngại vật thay đổi, việc hình thành một mạng topo mới là khó khăn và mất nhiều công sức.

Phương pháp tiếp cận không gian trống (Free space approach) tạo ra đồ thị bằng cách sử dụng trung điểm của các đường nối giữa các đỉnh chướng ngại vật làm thành các điểm để robot có thể đi qua. Nhưng phương pháp này khó có thể tìm được đường đi phù hợp trong những môi trường có mật độ chướng ngại vật cao.

Phương pháp lộ trình xác suất (Probabilistic roadmap method) gieo các điểm ngẫu nhiên trong khu vực không có chướng ngại vật, các cạnh đô thị tạo thành bởi đường nối các điểm khả thi không va chạm chướng ngại vật (*Hình 1.6*). Tuy nhiên, việc gieo ngẫu nhiên như vậy không đảm bảo đường đi là tối ưu.

**

Hình 1.6. Phương pháp lộ trình xác suất

Phương pháp tiếp cận theo phân rã ô (Cell decomposition approach) chia môi trường làm việc thành các ô vùng (Cell). Ví dụ như phân rã ô chính xác (exact cell decomposition), phân rã ô thích ứng (adaptive cell decomposition), phân rã ô xấp xỉ (approximate cell decomposition). Xoay trục tọa độ (the Coordinate transformation method) .... Phương pháp này được sử dụng rộng rãi hiện nay giúp việc di chuyển trở nên mượt mà hơn, tìm ra được các đường đi có độ dài tối ưu hơn.

1.3 Thuật toán tìm kiếm

Sau khi mô hình hóa môi trường, ta chuyển sang bước xây dựng thuật toán tìm kiếm. Thuật toán sẽ tìm đường đi từ trạng thái xuất phát tới một trạng thái đích trong không gian trạng thái. Mỗi thuật toán sẽ có thứ tự phát triển các nút để duyệt khác nhau. Kết quả của quá trình này là tập các phép chuyển gắn với đường đi tìm được [11].

Các thuật toán tìm kiếm có thể chia thành hai loại chiến lược:

* + Chiến lược tìm kiếm không có thông tin (tìm kiếm mù).
  + Chiến lược tìm kiếm có thông tin.

Không gian trạng thái là tập hợp tất cả các trạng thái có thể đi đến từ trạng thái ban đầu bởi một dãy các phép biến đổi trạng thái. Ta có thể biểu diễn không gian trạng thái bằng đồ thị có hướng trong đó mỗi nút là một trạng thái và cạnh ứng với một phép chuyển đổi trạng thái.

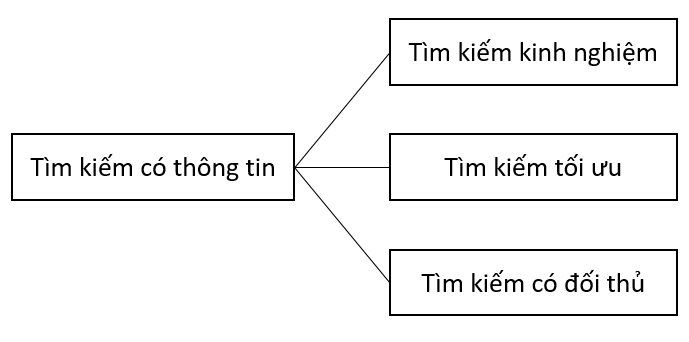
1.3.1 Chiến lược tìm kiếm không có thông tin

Tìm kiếm không có thông tin (tìm kiếm mù) là những thuật toán không có thông tin hướng dẫn nào cho việc tìm kiếm, mà ta chỉ phát triển từ vị trí xuất phát cho tới khi gặp vị đích. Các thuật toán tìm kiếm mù phổ biến như: tìm kiếm theo chiều rộng, tìm kiếm theo chiều sâu,…

Nhược điểm của nhóm thuật toán này là kém hiệu quả, tốc độ tìm kiếm chậm trong những không gian tìm kiếm lớn.

1.3.2 Chiến lược tìm kiếm có thông tin

Tìm kiếm có thông tin là những thuật toán tìm kiếm dựa trên những thông tin đánh giá thông qua hàm đánh giá. Hàm đánh giá là hàm ước lượng, đánh giá mức độ tốt hoặc xấu, khả năng về đích của mỗi trạng thái. Các thuật toán tìm kiếm có thông tin được chia làm ba loại như *Hình 1.7*:



Hình 1.7. Các loại thuật toán tìm kiếm có thông tin

Thuật toán tìm kiếm kinh nghiệm (tìm kiếm heuristic) là kỹ thuật tìm kiếm dựa vào kinh nghiệm và sự hiểu biết của chúng ta về vấn đề cần giải quyết để xây dựng nên hàm đánh giá hướng dẫn sự tìm kiếm. Những thuật toán phổ biến của nhóm thuật toán này là: thuật toán tốt nhất đầu tiên, thuật toán nhánh cận,…

Trong thực tế, ta không chỉ quan tâm đến việc tìm ra kết quả mà còn phải quan tâm đến việc kết quả đó có tối ưu hay không. Các thuật toán thuộc nhóm thuật toán tìm kiếm mù và tìm kiếm kinh nghiệm chưa quan tâm đến độ dài hay chi phí của đường đi của kết quả. Thuật toán tìm kiếm tối ưu sẽ giải quyết vấn đề này. Hàm đánh giá của nhóm thuật toán này sẽ tối ưu hơn so với hàm đánh giá của tìm kiếm kinh nghiệm. Các thuật toán tiêu biểu là: thuật toán nhánh cận, thuật toán A\*,…

Thuật toán tìm kiếm có đối thủ (tìm kiếm đối kháng) là nhóm thuật toán thường được sử dụng trong các trò chơi đơn giản dạng cờ vua, caro, tictactoe,… Không gian trạng thái của thuật toán tìm kiếm có đối thủ là cây trò chơi bao gồm các nước đi có thể của cả hai đối thủ. Thuật toán tiêu biểu của nhóm thuật toán này là thuật toán minimax.

1.4 Thuật toán tìm kiếm theo chiều rộng (Breadth First Search – BFS)

Thuật toán tìm kiếm theo chiều rộng (Breadth First Search – BFS) thuộc nhóm thuật toán có chiến lược tìm kiếm không có thông tin (tìm kiếm mù). Thuật toán BFS là một thuật toán đệ quy để duyệt qua tất cả các phần tử của cây hoặc đồ thị, sử dụng hàng đợi (queue) để ghi nhớ phần tử liền kề. Tìm kiếm theo chiều rộng có thể được sử dụng cho hai mục đích: tìm kiếm đường đi từ một phần tử xuất phát cho trước tới một phần tử mục tiêu và tìm kiếm đường đi từ phần tử xuất phát tới tất cả các phần tử khác [11].

Ý tưởng của thuật toán BFS như sau: từ một đỉnh xuất phát *v* lần lượt duyệt các phần tử kề xung quanh của đỉnh xuất phát và cho vào hàng đợi. Tiếp tục xét phần tử đầu tiên *u* của hàng đợi ra và duyệt các phần tử *t* kề với phần tử *u* vừa xét, rồi lại cho tất cả các phần tử *t* (chưa xét) vào hàng đợi. Thuật toán cứ lặp lại như vậy cho tới khi hàng đợi rỗng hoặc xét đến phần tử mục tiêu.

Các bước thực hiện thuật toán tìm kiếm theo chiều rộng:

Bảng 1.1. Thuật toán tìm kiếm theo chiều rộng

|  |
| --- |
| **Procedure Breadth\_First\_Search;**  **Begin**   Khởi tạo danh sách QUEUE = {trạng thái ban đầu};  **loop do**  **if** (QUEUE rỗng) **then** {                     Thông báo tìm kiếm thất bại; **stop**};              Loại phần tử u ở đầu danh sách QUEUE;  **if** (u là phần tử mục tiêu) **then** {                     Thông báo tìm kiếm thành công; **stop**};              Thêm các phần tử kề với u vào cuối danh sách QUEUE;  **End;** |

1.5 Thuật toán tối ưu hoá đàn kiến (Ant Colony Optimization - ACO)

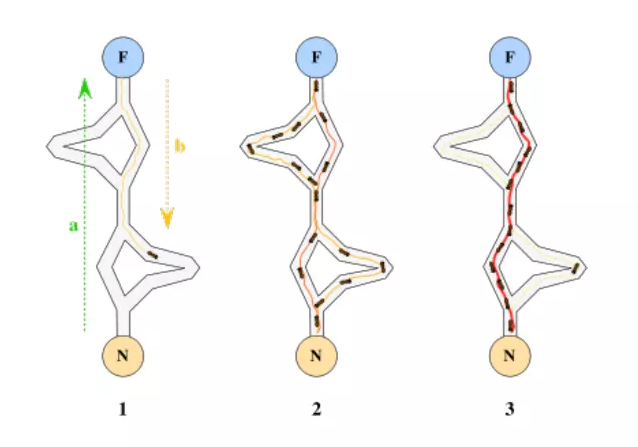
1.5.1 Giới thiệu chung

Trong tự nhiên, kiến là loài sinh vật sống theo bày đàn với số lượng lớn và tính tổ chức rất cao bởi vậy việc di chuyển và phối hợp giữa các thành viên trong đàn đều rất quan trọng. Nhờ sự chặt chẽ trong việc tổ chức đó mà những con kiến nối tiếng với khả năng tìm kiếm đường đi ngắn nhất từ tổ tới nguồn thức ăn mà không cần dùng đến thị giác vì hầu hết loài kiến đều có thị giác kém và một số loài kiến bị mù.



Hình 1.8. Đàn kiến kiếm ăn

Ban đầu, những con kiến đều đi di chuyển ngẫu nhiên để tìm kiếm nguồn thức ăn (*Hình 1.9.1*). Trong khi di chuyển, những con kiến sẽ để lại một chất hóa học đặc biệt, được gọi là pheromone trên đường đi để đánh dấu. Từ đó, đoạn đường nào có nhiều pheromone hơn sẽ được nhiều con kiến đi hơn. Trong khi di chuyển, những con kiến đi sau sẽ cảm nhận nồng độ pheromone trên đoạn đường mà những con kiến đi trước để lại và lựa chọn đoạn đường có nồng độ pheromone cao hơn, từ đó lần theo đến nguồn thức ăn mà không cần phải đi lang thang (*Hình 1.9.2*). Bên cạnh đó, theo thời gian, lượng pheromone lưu lại trên đường đi sẽ bắt đầu bay hơi, do vậy làm giảm đi nồng độ của nó. Thời gian càng dài lượng pheromone mà những con kiến đi trước để lại trên đoạn đường sẽ càng giảm, lâu dần pheromone cũng sẽ bốc hơi hoàn toàn. Nhờ đó, những con đường ngắn hơn sẽ có nồng độ pheromone lưu lại cao hơn so với các con đường dài vì thời gian thực hiện để bao phủ quãng đường ngắn nhất sẽ là tối thiểu so với các quãng đường khác. Lâu dần, đường đi dài hơn của những con kiến đầu tiên sẽ bốc hơi hoàn toàn và những đường đi đó sẽ không còn thu hút những con kiến đi sau. Ngoài ra, nồng độ pheromone phụ thuộc vào số lượng kiến di chuyển trên quãng đường, nồng độ pheromone càng cao thì số lượng kiến di chuyển trên quãng đường đó là càng nhiều và ngược lại (*Hình 1.9.3*).



Hình 1.9. Mô tả tổng quan về kiến và pheromone

Lấy cảm hứng từ cách tìm kiếm nguồn thức ăn của loài kiến trong tự nhiên, nhiều nhà nghiên cứu đã đặt những viên gạch đầu tiên cho thuật toán tối ưu đàn kiến (Ant colony optimization algorithms). Trước tiên, ta cùng điểm qua những dấu mốc trong sự phát triển đó: năm 1991, Ant System (AS) được phát triển trong luận án tiến sĩ của Marco Dorigo. Đến năm 1996, Max-Min Ant System được phát triển bởi Hoos và Stutzle. Giải thuật Ant Colony được phát triển bởi Gambardella Dorigo năm 1997 [3].

Thuật toán tối ưu đàn kiến là thuật toán tìm kiếm đường đi tốt nhất được lấy ý tưởng dựa trên hành vi của loài kiến lưu dấu vết bằng pheromone để đánh dấu các đoạn đường đã đi, dựa vào những đoạn đường đã được lưu để phân tích tìm đường tốt nhất. Vì vậy, thuật toán này còn được biết tới là một thuật toán tối ưu sinh học.

Thuật toán ACO được ứng dụng cho một số lượng lớn bài toán tối ưu tổ hợp. Những ứng dụng hiện nay của ACO chia thành hai lớp ứng dụng:

* + Lớp bài toán tối ưu tổ hợp NP-khó cho công nghệ cũ. Đặc tính thành công nhất của ứng dụng ACO tới những bài toán mà ở đó kiến kết hợp với vùng tìm kiếm để có cách giải quyết tốt.
  + Lớp bài toán tìm đường đi ngắn nhất. Lớp ứng dụng này của ACO được dùng để kết nối đường thông tin.

Sau đây là lịch sử phát triển của ứng dụng, tổng quan mở rộng trong ứng dụng của ACO.

Bài toán tổ hợp đầu tiên được giải quyết bởi thuật toán ACO là bài toán người chào hàng (Traveling Salesman Problem - TSP) bởi vì bài toán này được biết đến nhiều nhất của lớp bài toán NP-khó và lớp bài toán tìm đường đi ngắn nhất, là bài toán chuẩn để đánh giá hiệu quả của các thuật toán giải các bài toán tổ ưu tổ hợp. Ngoài ra, bài toán này còn phù hợp để dễ dàng mô phỏng sự sinh hoạt của loài kiến [4]. Từ khi ứng dụng đầu tiên của hệ thống kiến (Ant System - AS) trong luận án của Marco Dorigo năm 1991, nó trở thành một đánh giá chung của vài đóng góp tốt hơn trong mô hình thực thi ACO hơn AS.

Năm 1994, thuật toán được ứng dụng vào bài toán lập lịch cho cửa hàng và bài toán phân định bậc hai (Quadratic Assignment Problem - QAP). Năm 1997, sau một năm công bố của nhà báo đầu tiên cho ACO năm 1996, số ứng dụng ACO bắt đầu tăng nhanh. Sau đó, nhiều nhà nghiên cứu khác nhau đã sử dụng ACO để giải quyết số lượng lớn bài toán tối ưu tổ hợp như tổng quát nhiệm vụ, chuỗi chung ngắn nhất, tập che phủ,…

Từ phần ứng dụng trước đó, ACO gần như được sử dụng cho mục đích kỹ thuật, cụ thể để thiết kế nghiên cứu thuật toán cho độc giả biểu diễn cấu trúc như tập quy tắc logic classical, quy tắc logic fuzzy và thông tin suy luận Bayes đã đưa ra những kết quả hứa hẹn.

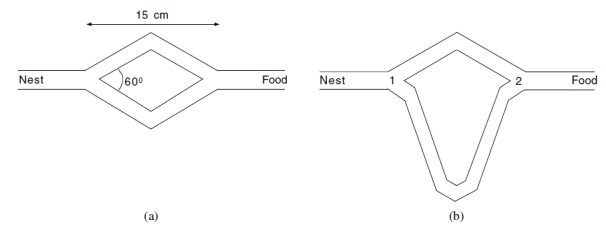
Ngày nay, ACO được ứng dụng cho nhiều bài toán khác nhau và đưa ra kết quả tốt như: trình tự chuỗi, đường đi, lập lịch và thông tin gói điều khiển. Bên cạnh đó, ACO còn được sử dụng để giải quyết các bài toán mới của thế giới với kết quả thu được đầy hứa hẹn.

1.5.2 Thí nghiệm trên cây cầu đôi

Trên quãng đường di chuyển, những con kiến sẽ để lại pheromone để đánh dấu đường đi. Bằng cách cảm nhận nồng độ pheromone trên đoạn đường, những con kiến đi sau có thể lần theo đường đi đến nguồn thức ăn được các con kiến đi trước khám phá bằng cách di chuyển ngẫu nhiên. Các con kiến chịu ảnh hưởng của các vết mùi của các con kiến khác là ý tưởng để thiết kế thuật toán ACO.

Có nhiều thực nghiệm nghiên cứu về hành vi để lại pheromone và đi theo pheromone của loài kiến. Một thực nghiệm được thiết kế bởi Deneubourg và các đồng nghiệp nhờ dùng một chiếc cầu đôi nối từ tổ của loài kiến tới nguồn thức ăn [14]. Họ đã thực nghiệm với tỉ lệ độ dài đường 𝑟 = giữa hai nhánh của chiếc cầu đôi khác nhau, trong đó là độ dài của nhánh dài còn là độ dài của nhánh ngắn.

Trong thực nghiệm thứ nhất, chiếc cầu đôi có hai nhánh bằng nhau (*Hình 1.10.a*). Ban đầu, kiến lựa chọn đường đi một cách ngẫu nhiên đi từ tổ đến nguồn thức ăn, kiến trên cả hai nhánh, nhưng sau một thời gian các con kiến này tập trung đi theo cùng một nhánh. Kết quả có thể được giải thích như sau: ban đầu không có vết pheromone nào trên cả hai nhánh, do đó các kiến lựa chọn nhánh bất kỳ với xác suất như nhau. Thực tế, theo một cách ngẫu nhiên sẽ có một nhánh có số lượng kiến lựa chọn nhiều hơn nhánh kia. Do kiến để lại pheromone trong quá trình di chuyển, nhánh có nhiều kiến lựa chọn sẽ có nồng độ pheromone lớn hơn nồng độ pheromone của nhánh còn lại. Dẫn đến dần dần các con kiến sẽ chọn cạnh có nồng độ pheromone cao hơn, từ đó nồng độ pheromone trên cạnh lớn hơn sẽ ngày càng lớn hơn vì ngày càng có nhiều kiến lựa chọn. Cuối cùng, hầu như tất cả các kiến sẽ tập trung trên cùng một nhánh. Thực nghiệm này cho thấy là sự tương tác cục bộ giữa các con kiến nhờ thông tin gián tiếp qua pheromone để lại mà có thể điều chỉnh hoạt động toàn cục của đàn kiến.



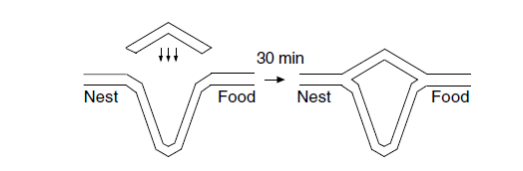
Hình 1.10. Thí nghiệm trên cây cầu đôi

*(a) Hai nhánh có độ dài bằng nhau.*

*(b) Hai nhánh có độ dài khác nhau.*

Trong thực nghiệm thứ hai (*Hình 1.10.b*), độ dài giữa nhánh dài hơn gấp đôi nhánh ngắn hơn (tỉ lệ 𝑟 = 2). Trong trường hợp này, sau một thời gian tất cả các kiến chọn đi duy nhất theo đường ngắn hơn. Cũng như thực nghiệm thứ nhất, ban đầu kiến lựa chọn đường đi theo hai nhánh như nhau, ta có thể giả sử một nửa kiến đi theo nhánh ngắn và một nửa đi theo nhánh dài (mặc dù ngẫu nhiên nhưng vẫn có thể một nhánh sẽ có nhiều lựa chọn hơn nhánh khác). Tuy nhiên thực nghiệm này khác với thực nghiệm đầu tiên bởi vì một nhánh ngắn hơn nhánh kia nên các kiến lựa chọn theo nhánh ngắn hơn sẽ nhanh chóng quay trở lại từ nguồn thức ăn về tổ và phải lựa chọn giữa nhánh ngắn và nhánh dài, nồng độ pheromone trên nhánh ngắn đang cao hơn nồng độ mùi trên nhánh dài, do đó nó ưu tiên lựa chọn đi theo nhánh ngắn hơn. Từ kết quả này, chúng ta có thể quan sát thấy rằng, với nhánh dài gấp đôi nhánh ngắn thì không phải tất cả các kiến cũng đều đi theo nhánh ngắn hơn và phải mất một thời gian nữa thì các kiến này sẽ đi theo nhánh ngắn hơn. Điều này được hiểu là một loại thăm dò, tìm đường mới.

Để kiểm nghiệm việc tạo pheromone và sự bay hơi của chúng, một thực nghiệm mới được thực hiện như sau. Ban đầu, từ tổ đến nguồn thức ăn chỉ có một nhánh dài và sau 30 phút thì lắp thêm một nhánh ngắn. Trong trường hợp này, nhánh ngắn thường không được kiến chọn mà chúng tập trung đi trên nhánh dài. Điều này có thể giải thích bởi nồng độ pheromone trên cạnh dài cao hơn, cùng với sự bay hơi chậm của pheromone nên đại đa số các con kiến lựa chọn nhánh dài và hành vi này tiếp tục củng cố trên nhánh dài, ngay cả khi một nhánh ngắn xuất hiện. Việc bay hơi pheromone có thể có lợi cho việc tìm đường mới, nghĩa là việc bay hơi có thể giúp kiến quên đi đường đi tối ưu cục bộ để tìm đường đi mới tốt hơn có thể được khám phá (*Hình 1.11*).



Hình 1.11. Thí nghiệm ban đầu chỉ một nhánh dài và sau 30 phút thêm nhánh ngắn

1.5.3 Tối ưu hóa đàn kiến tiêu chuẩn

Khi giải các bài toán thuộc loại NP-khó, người ta thường sử dụng các phương pháp heuristic để tìm kiếm lời giải đủ tốt cho bài toán. Thuật toán ACO kết hợp thông tin heuristic này với phương pháp học tăng cường nhờ mô phỏng hành vi của đàn kiến để tìm lời giải tốt hơn.

Với điểm bắt đầu và điểm đích biết trước, dùng đàn kiến m con thực hiện lời giải. Khi di chuyển, kiến sẽ xem xét chiều dài và lượng pheromone của cạnh từ vị trí hiện tại đến vị trí tiếp theo. Ở mỗi bước di chuyển, mỗi con kiến k di chuyển từ đỉnh hiện tại *i* đến một đỉnh chưa được đi qua *j* trong tập láng giềng của *i*. Nếu có nhiều hơn một đỉnh chưa được đi qua trong trong tập láng giềng của *i*, kiến k sẽ chọn đỉnh tiếp theo sẽ di chuyển đến theo công thức tính xác suất như sau [5]:

(1-1)

Trong đó:

* 𝜏i,j : lượng pheromone trên cạnh *i*,*j*
* ηi,j : hàm heuristic ()
* α: tham số kiểm soát ảnh hưởng của 𝜏i,j
* β: tham số kiểm soát ảnh hưởng của ηi,j
* allowedi: các đỉnh trong tập làng giềng của i chưa được đi qua.
* di,j : khoảng cách giữa hai đỉnh *i*,*j*

Sau khi tất cả các con kiến hoàn thành lời giải của mình, lượng pheromone trên tất cả các đoạn đường sẽ được cập nhật bằng cách làm bay hơi lượng pheromone cũ tại thời điểm t và thêm lượng pheromone được tiết ra bởi mỗi con kiến. Tùy theo đường đi mà đàn kiến tìm được mà vết pheromone trên mỗi cạnh sẽ được điều chỉnh tăng hoặc giảm. Ta có công thức cập nhật mùi như sau [5]:

(1-2)

Trong đó:

* 𝜌: tốc độ bay hơi của pheromone.
* *Lk*: độ dài quãng đường di chuyển của kiến k.
* *Q*: tham số điều chỉnh lượng pheromone.

Các bước thực hiện để giải bài toán ACO:

Bảng 1.2. Thuật toán tối ưu đàn kiến

|  |
| --- |
| **Input:** điểm bắt đầu, điểm kết thúc, môi trường hoạt động của Robot  **Procedure Thuật toán ACO;**  **Begin**   Khởi tạo tham số, ma trận mùi, khởi tạo m con kiến;  **repeat**  **for** k = 1 **to** m **do**                     Kiến k xây dựng lời giải;  **end-for**  Cập nhật mùi;              Cập nhật lời giải tốt nhất;  **until(**Điều kiện kết thúc**);**  Đưa ra lời giải tốt nhất;  **End;**  **Output:** Đường đi tối ưu mà thuật toán tìm được |

1.5.4 Các yếu tố quyết định đến hiệu quả của thuật toán

Khi áp dụng thuật toán ACO cho một bài toán cụ thể, có ba yếu tố ảnh hưởng đến hiệu quả của thuật toán [16]:

* + Xây dựng môi trường thích hợp: việc xây dựng môi trường thích hợp là một trong những yếu tố quan trọng ảnh hưởng đến hiệu quả của thuật toán. Khi môi trường quá rộng, kiến sẽ khó khăn trong việc chọn đường đi tới đích do khả năng di chuyển ngẫu nhiên của kiến, dẫn đến đường đi không phải là đường đi tối ưu nhất.
  + Chọn quy tắc cập nhật mùi: quy tắc cập nhật mùi cho thấy chiến lược học của thuật toán, giúp bài toán thông minh hơn
  + Chọn thông tin heuristic: Thông tin heuristic tốt sẽ tăng hiệu quả thuật toán. Tuy nhiên, nhiều bài toán không có thông tin này nên ta có thể đánh giá chúng như nhau. Khi đó ban đầu, thuật toán chỉ đơn thuần tìm kiếm theo phương thức ngẫu nhiên nên vết mùi thể hiện định hướng của học tăng cường và thuật toán vẫn được thực hiện..

1.5.5 So sánh thuật toán ACO với các thuật toán tối ưu khác

Thuật toán ACO có những điểm khác biệt so với các thuật toán tối ưu khác [6]:

* + Trong thuật toán ACO, với mỗi cá thể kiến nhân tạo di chuyển để lại một lượng pheromone nhất định và từ đó nồng độ pheromone để lại quyết định chính xác đường đi tối ưu nhất.
  + Thuật toán ACO sử dụng kết hợp thông tin kinh nghiệm (heuristic) và học tăng cường qua các vết pheromone của các con kiến nhân tạo để giải các bài toán tối ưu tổ hợp bằng cách đưa về bài toán tìm đường đi tối ưu trên đồ thị cấu trúc tương ứng của bài toán.

1.5.6 Ưu điểm và nhược điểm của thuật toán ACO

Bảng 1.3. Ưu điểm và nhược điểm của thuật toán ACO [12] [16].

|  |  |
| --- | --- |
| Ưu điểm | * + Thuật toán ACO có cơ chế phản hồi tích cực và hiệu quả trong việc giải quyết các vấn đề phân tán.   + Thuật toán ACO có thể dễ dàng kết hợp với các thuật toán khác. |
| Nhược điểm | * + Trong giai đoạn đầu của quá trình tối ưu hóa, thuật toán dễ rơi vào trạng thái tối ưu cục bộ.   + Thuật toán ACO có tốc độ hội tụ chậm dẫn đến mất nhiều thời gian hơn để tìm kiếm, không phù hợp với các bài toán có quy mô lớn. |

CHƯƠNG 2 BÀI TOÁN LẬP KẾ HOẠCH ĐƯỜNG ĐI CHO ROBOT DI ĐỘNG TRONG MÔI TRƯỜNG TĨNH CÓ CHƯỚNG NGẠI VẬT

2.1 Tổng quan bài toán

2.1.1 Giới thiệu bài toán

Ngày nay, robot đã trở thành một phần quan trọng trong nhiều lĩnh vực khác nhau. Robot được sử dụng rộng rãi vì nhiều lý do như độ tin cậy cao, nâng cao hiệu quả công việc và có thể sử dụng liên tục. Một đánh giá quan trọng trong sự thành công của hệ thống robot là tính tối ưu trong toàn bộ chuyển động của robot, ví dụ như robot vừa phải tìm ra đường đi ngắn nhất, vừa đảm bảo robot di chuyển an toàn trong không gian làm việc. Hiện tại, các chuyển động này được tối ưu hóa một cách thủ công trong nhiều kịch bản công nghiệp. Điều này dễ xảy ra sai sót và gây tốn kém. Do đó, vấn đề điều hướng chuyển động cho robot vẫn luôn được nhiều nhà nghiên cứu quan tâm.

Trong thực tế, năng lượng của robot không phải là vô hạn, điều này tác động rất lớn đến chu trình đường đi của robot. Vì vậy, bài toán tối ưu hóa năng lượng cho robot là một trong những bài toán nhận được nhiều quan tâm trong nhiều năm gần đây. Một trong các bài toán tối ưu hoá cho robot chính là bài toán lập lịch đường đi cho robot di động. Đây là một bài toán NP-khó. Lập lịch đường đi cho robot di động phải đảm bảo ba mục tiêu: đường đi ngắn nhất, đường đi an toàn không va chạm chướng ngại vật và đường đi trơn tru nhất.

Điều hướng là quá trình lên kế hoạch và điều khiển robot di chuyển một cách chính xác trên tuyến đường đi đã được lập kế hoạch trước đó. Ba yếu tố cần quan tâm để điều hướng robot là định vị, lập kế hoạch đường đi và điều khiển robot di chuyển. Trong đó, yếu tố quan trọng nhất trong quá trình điều hướng robot là lập kế hoạch đường đi (path planning), yếu tố này giúp cho robot xác định đường đi phù hợp và an toàn trong không gian [7]. Một kế hoạch đường đi tốt phải có khả năng tìm ra một đường đi tối ưu về nhiều mặt như thời gian di chuyển, độ dài đường đi, độ an toàn, độ mượt,... giúp robot di chuyển từ vị trí này đến vị trí khác mà không xảy ra va chạm với chướng ngại vật trong không gian môi trường.

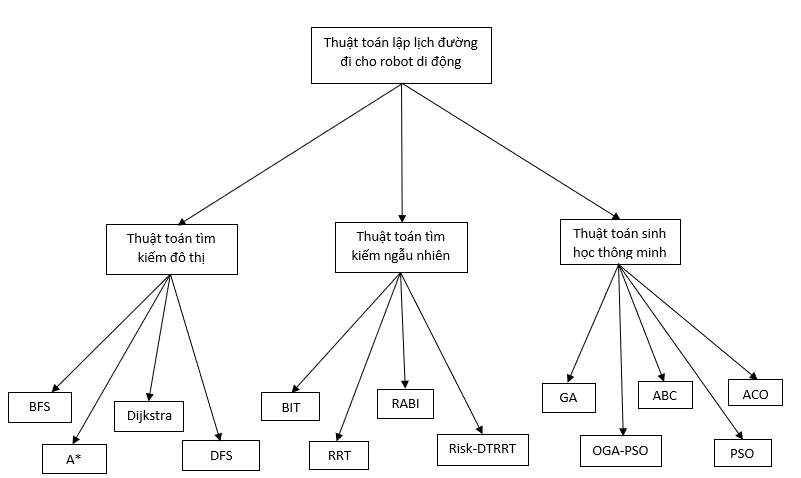
Ngoài ra, đối với các môi trường động có chướng ngại vật không cố định, việc lập kế hoạch đường đi cho robot còn phải xử lý việc cảm biến được các thay đổi của môi trường, giảm thiểu sự tác động của các chướng ngại vật trong môi trường lên robot, tìm ra đường đi tối ưu trong thời gian phù hợp để tương tác thường xuyên với robot... Do nhiều khó khăn như vậy, bài toán lập kế hoạch đường đi cho robot được xác định thuộc lớp bài toán NP-khó (Non-deterministic Polynomial-time Hard) [4].

Ngoài ra, việc lập kế hoạch đường đi cho robot còn cần phải đáp ứng giải quyết được các vấn đề khác như robot có số lượng khớp nối nhiều hơn (robot trong công nghiệp...), các ràng buộc khác nhau (một robot chỉ có thể đi về phía trước...), các nhiệm vụ phức tạp hơn (robot thao tác với các đối tượng khác...) và môi trường không hoàn hảo... [8].

* Phân loại điều hướng:
* Điều hướng phản ứng: robot không có bản đồ hoặc bất kỳ ý tưởng nào về vị trí của nó trong môi trường. Robot di chuyển ngẫu nhiên và thu thập thông tin từ môi trường thông qua cảm biến tiếp xúc để nhận biết đường đi hay chướng ngại vật, robot có khả năng cảm nhận các sự vật trong môi trường từ đó di chuyển.
* Điều hướng dựa trên bản đồ: là quá trình tạo ra một chu trình đường đi để robot di chuyển từ nơi này đến nơi khác theo một số tiêu chí như thời gian tốt nhất hoặc với chi phí thấp nhất.

Lập lịch kế hoạch đường đi cho robot di động là bài toán NP-khó. Vì vậy, các nhà nghiên cứu đã đưa ra rất nhiều hướng giải quyết nhằm giúp giải bài toán lập lịch kế hoạch đường đi robot di động và cho ra kết quả tốt nhất. Từ các nghiên cứu thu được rất nhiều thuật toán. Các thuật toán này được chia thành [9]:

* + Thuật toán tìm kiếm đồ thị [9]: các thuật toán cổ điển để tìm kiếm đồ thị bao gồm thuật toán Dijkstra, thuật toán A\*, Thuật toán DFS, thuật toán BFS,... Thuật toán Dijkstra và các thuật toán A\* được triển khai rộng rãi trong hệ thống Hệ điều hành Robot (ROS). Các phương pháp này được cải tiến thông qua ước tính heuristic, cải thiện hiệu quả tìm kiếm và giúp giảm số lượng lưới tìm kiếm. Tuy nhiên, trong các môi trường phức tạp thì cho ra hiệu quả sẽ thấp.
  + Thuật toán sinh học thông minh: là một thuật toán thông minh để mô phỏng hành vi của côn trùng tiến hóa và phỏng sinh học, trong đó các nghiên cứu chủ yếu sử dụng Thuật toán di truyền (Genetic Algorithm - GA), Thuật toán tối ưu đàn kiến ​​(Ant Colony Optimization - ACO), Thuật toán đàn ong nhân tạo (Artificial Bee Colony - ABC), Thuật toán tối ưu hóa bầy đàn (Particle Swarm Optimization - PSO). Liu và cộng sự đã kết hợp trường tiềm năng nhân tạo và phương pháp tối ưu hóa cục bộ hình học để tìm kiếm con đường tối ưu toàn cục bằng cách tối ưu hóa thuật toán di truyền - thuật toán tối ưu hóa bầy đàn (OGA-PSO) nhằm tăng tốc độ tính toán nhiều nhất có thể và giải quyết các thiếu sót của phương pháp tối ưu cục bộ trong việc lập kế hoạch đường đi ngắn nhất và không va chạm cho robot. Mac và cộng sự đề xuất tối ưu hóa bầy hạt được gọi là tối ưu hóa bầy hạt đa mục tiêu bị ràng buộc với phương pháp cập nhật tăng tốc cung cấp các đường dẫn robot toàn cầu tối ưu về độ dài và độ trơn của đường dẫn [9].
  + Thuật toán tìm kiếm ngẫu nhiên: bao gồm thuật toán Batch Informed Trees (BIT), Regionally Accelerated Batch Informed Trees (RABIT), Rapidly exploring Random Tree (RRT), Risk-based Dual-Tree Rapidly exploring Random Tree (Risk-DTRRT). Các thuật toán này được sử dụng rộng rãi trong các môi trường động hoặc nhiều chiều [9].



Hình 2.1 Sơ đồ các phương pháp giải bài toán lập lịch đường đi cho robot di động

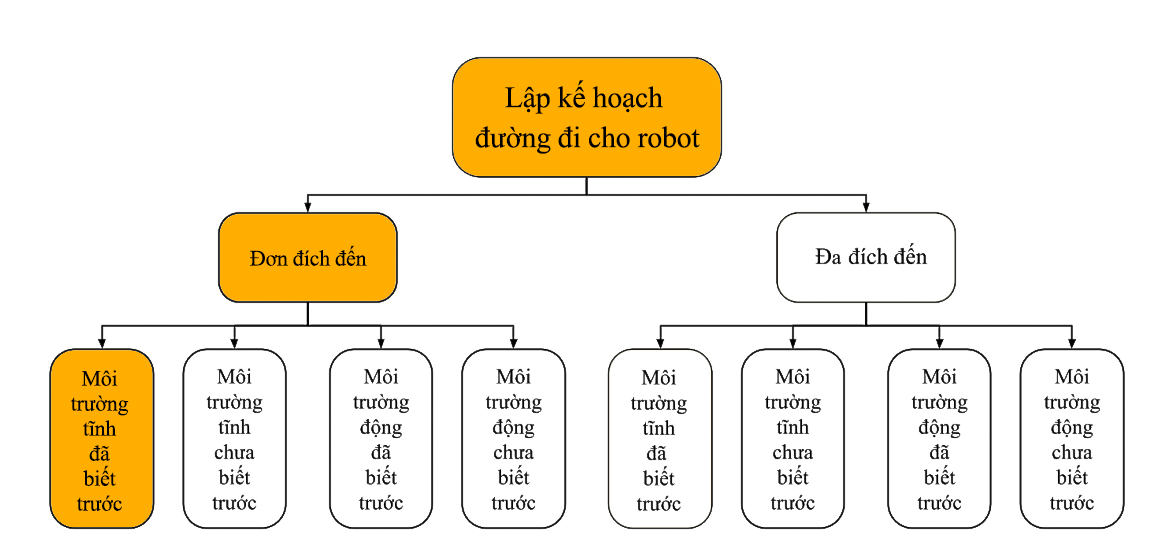
2.1.2 Phân loại bài toán

Dựa trên số đích cần đến của robot di động, bài toán lập kế hoạch đường đi cho robot có thể được chia thành hai loại (*Hình 2.2*):

* + Đơn mục tiêu: robot bắt đầu tại điểm xuất phát và di chuyển tới một đích đến.
  + Đa mục tiêu: robot bắt đầu tại điểm xuất phát và di chuyển qua tập hợp nhiều đích đến.

Ngoài ra, ta còn có thể phân loại bài toán chi tiết hơn dựa trên các tiêu chí về môi trường hoạt động:

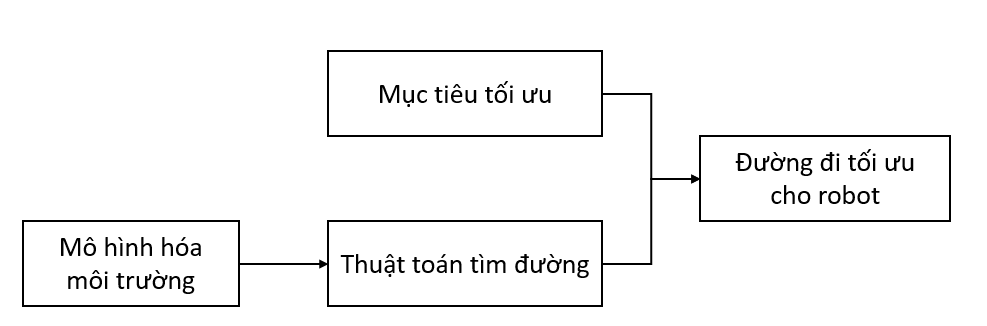
* **Kiểu chướng ngại vật**:
* Môi trường tĩnh: môi trường chỉ bao gồm các chướng ngại vật tĩnh không thay đổi.
* Môi trường động: môi trường chứa cả các chướng ngại vật tĩnh và các chướng ngại vật có thay đổi vị trí theo thời gian.
* **Lượng thông tin robot biết được về toàn bộ môi trường hoạt động:**
* Môi trường chưa biết trước: robot chỉ biết được một phần thông tin, hoặc chỉ biết được các thông tin không chắc chắn về môi trường hoạt động. Trường hợp này yêu cầu robot phải được trang bị các thiết bị cảm biến để thu thập được các thông tin của chướng ngại vật như vị trí, hình dạng, kích thước, … dùng các thông tin đó để tìm ra đường đi tối ưu cục bộ.
* Môi trường đã biết trước: robot đã biết rõ toàn bộ vị trí của các chướng ngại vật trong môi trường trước khi bắt đầu hoạt động. Đường đi của robot tìm được có thể là đường đi tối ưu toàn cục do toàn bộ thông tin đã được biết trước.



Hình 2.2. Phân loại bài toán lập kế hoạch đường đi cho robot

2.1.3 Định hướng bài toán

Định hướng chung để giải quyết bài toán lập kế hoạch đường đi cho robot trong môi trường đã biết trước thường được thực hiện theo quy trình như *Hình 2.3*:

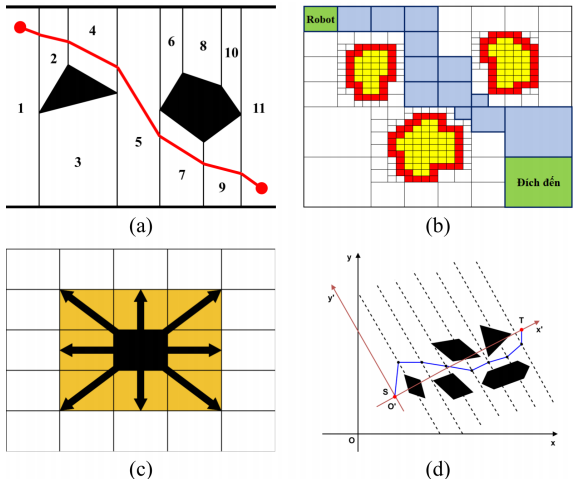


Hình 2.3. Định hướng giải quyết bài toán lập kế hoạch đường đi cho robot trong môi trường [8]

Định hướng giải quyết bài toán lập kế hoạch đường đi cho robot di động trong môi trường bao gồm ba phần:

* + Lựa chọn phương pháp mô hình hóa môi trường
  + Lựa chọn thuật toán tìm đường đi
  + Nhận thức và tính toán các mục tiêu tối ưu từ đó đưa ra đường đi tối ưu nhất.

Một mô hình môi trường phù hợp giúp hiểu rõ hơn về các thành phần của môi trường, từ đó giảm các thao tác không cần thiết và giảm số lần tính toán của robot. Các phương pháp khác nhau để mô hình hóa môi trường đã được liệt kê ở *mục 1.2*. Các phương pháp mô hình hóa cổ điển thường không đảm bảo tìm ra đường đi tối ưu nhất. Trong khi đó, phương pháp phân rã ô (cell decomposition approach) ứng dụng các thuật toán dựa trên trí tuệ nhân tạo, đang được sử dụng phổ biến hiện nay. Các phương pháp phân rã ô thường gặp là: phân rã ô chính xác (*Hình 2.4.a*), phân rã ô thích ứng (*Hình 2.4.b*), phân rã ô xấp xỉ (*Hình 2.4.c*), xoay trục tọa độ phân rã ô trục dọc (*Hình 2.4.d*),… Ứng dụng vào việc mô hình hóa môi trường phân rã theo ô như trên các giải thuật xấp xỉ giúp các nghiên cứu có thể tìm ra được các đường đi có độ dài tối ưu hơn nhiều so với các phương pháp cổ điển, ngoài ra còn có thể tối ưu thêm nhiều mục tiêu khác giúp bài toán đáp ứng được tốt hơn các yêu cầu của robot trong thực tế. Mỗi phương pháp phân rã ô sẽ phù hợp với các mô hình bài toán đầu vào khác nhau và có các thuật toán phù hợp để tìm ra đường đi tối ưu riêng.



Hình 2.4. Một số phương pháp mô hình hóa môi trường phân rã ô [10]

Từ các phương pháp mô hình hóa môi trường, sử dụng các thuật toán tìm đường trong đồ thị để tìm ra đường đi tối ưu cho robot. Có nhiều thuật toán tìm đường khác nhau được nêu trong như:

* **Thuật toán cổ điển:** Rolling windows algorithm, Artificial potential field (APF), Cased-based learning method...
* **Thuật toán heuristic:** A\* algorithm, Dijkstra's algorithm, D\* algorithm…
* **Thuật toán dựa trên trí tuệ nhân tạo:**
* Thuật toán lấy cảm hứng từ tự nhiên (Nature inspired algorithms): Differential evolution (DE), Genetic algorithm (GA), Partide swarm optimization (PSO),...
* Thuật toán lấy cảm hứng từ con người (Human inspired algorithms): Neural Network, Fuzzy logic,...

Phương pháp mô hình hóa môi trường và thuật toán tìm đường phải phù hợp với nhau, vì hai phần này có liên quan mật thiết với nhau và ảnh hưởng rất lớn đến kết quả tìm kiếm đường đi tối ưu.

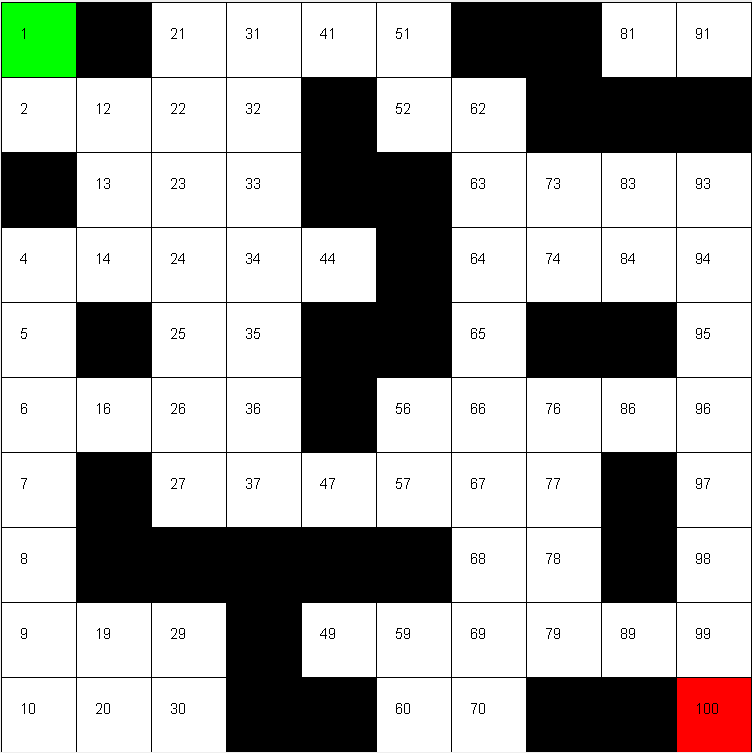
Ngoài ra, các mục tiêu khác cũng cần tối ưu để bài toán có thể đáp ứng được các yêu cầu trong thực tế. Có ba mục tiêu tối ưu thường được xét đến là:

* + **Độ dài đường đi (length):** đường đi càng ngắn, robot càng tiết kiệm thời gian và năng lượng.
  + **Độ an toàn đường đi (safety):** đường đi càng an toàn, robot càng tránh xa chướng ngại vật, giúp đảm bảo an toàn cho robot.
  + **Độ mượt đường đi (smoothness):** đường đi càng mượt trơn tru robot càng không phải xoay rẽ nhiều, giúp tiết kiệm thời gian và năng lượng trong việc robot chuyển hướng.

2.2 Phát biểu bài toán

Bài toán lập kế hoạch đường đi cho robot trong môi trường tĩnh đã biết trước được phát biểu như sau:

Trong một môi trường tĩnh 2D robot di chuyển có m chướng ngại vật, tọa độ vị trí các chướng ngại vật đã biết trước và không thay đổi trong cả quá trình xử lý bài toán (*Hình 2.5*). Robot được xem như là một điểm. Vị trí xuất phát () và vị trí đích () của robot được đánh màu lần lượt là màu xanh và màu đỏ. Các chướng ngại vật trong môi trường được đánh dấu bằng màu đen. Mục tiêu của bài toán là tìm ra đường đi khả thi ngắn nhất không va chạm chướng ngại vật, đi qua mỗi điểm một lần và tìm đến điểm đích tạo thành một chu trình.



Hình 2.5. Mô tả môi trường tĩnh đã biết trước chướng ngại vật

*(Chú thích: ô màu xanh – điểm xuất phát, ô màu đỏ - điểm đích, ô màu đen – chướng ngại vật )*

Bảng 2.1. Phát biểu bài toán lập kế hoạch đường đi cho robot trong môi trường tĩnh đã biết trước chướng ngại vật

|  |  |
| --- | --- |
| **Đầu vào** | * Điểm bắt đầu và điểm kết thúc với vị trí và * Tập m chướng ngại vật là các ô được xác định tọa độ |
| **Đầu ra** | * Đường đi mà robot cần tìm và độ dài đường đi đó |
| **Ràng buộc** | * Đường đi không va chạm với chướng ngại vật |
| **Mục tiêu** | * Độ dài đường đi ngắn nhất:   (2-1) |

2.3 Các nghiên cứu liên quan

Ngày nay, công nghệ ngày càng được sử dụng nhiều hơn trong cuộc sống của con người. Trong đó, robot di động đã được ứng dụng ở nhiều lĩnh vực khác nhau và mang lại những thành tựu nhất định. Tuy nhiên, vì robot di động có nguồn năng lượng hữu hạn đã làm ảnh hưởng đến quá trình di chuyển cũng như làm nhiệm vụ của nó. Vì vậy, rất nhiều giải thuật được đề xuất để giải quyết bài toán lập lịch đường đi nhằm giảm bớt lượng tiêu thụ năng lượng của robot trong khi di chuyển:

Giải thuật GA (Genetic Algorithm): thuật toán di truyền là thuật toán tối ưu ngẫu nhiên dựa trên cơ chế chọn lọc tự nhiên và tiến hóa di truyền. Nguyên lý cơ bản của thuật toán di truyền đã được Holland giới thiệu vào năm 1962. Thuật toán di truyền gồm 4 quá trình là: chọn lọc (selection), lai ghép (crossover) và đột biến (mutation). Thuật toán di truyền được ứng dụng đầu tiên trong hai lĩnh vực chính: tối ưu hóa và học máy. Trong đó, lĩnh vực tối ưu hóa thuật toán di truyền được phát triển nhanh chóng và ứng dụng trong nhiều lĩnh vực nhỏ hơn như tối ưu hàm, bài toán người chào hàng (TSP), xử lý ảnh và nhận dạng hệ thống.

Giải thuật PSO (Particle Swarm Optimization): là một thuật toán tiến hóa quần thể, được phát triển bởi James Kennedy và Russell C. Eberhart vào năm 1995. Thuật toán mô phỏng đàn chim bay đi kiếm thức ăn.

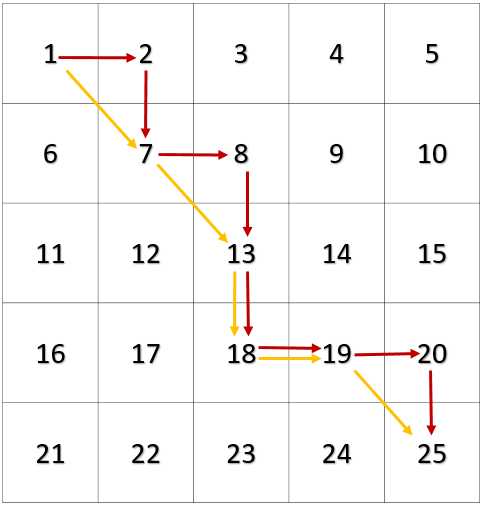
Giải thuật ABC (Artificial Bee Colony):  là giải thuật mô phỏng theo hành vi kiếm ăn của bầy ong trong tự nhiên và đã được áp dụng vào nhiều bài toán thực tế. Thuật toán được đề xuất bởi Karaboga vào năm 2005 [7]. Ý tưởng chính của thuật toán là từ một quần thể nguồn thức ăn, trong quá trình chạy thuật toán những con ong sẽ thay đổi vị trí nguồn thức ăn thành vị trí nguồn thức ăn tốt nhất [7].

CHƯƠNG 3 ĐỀ XUẤT GIẢI PHÁP CẢI TIẾN THUẬT TOÁN ĐÀN KIẾN KẾT HỢP THUẬT TOÁN TÌM KIẾM THEO CHIỀU RỘNG ĐỂ GIẢI BÀI TOÁN LẬP KẾ HOẠCH ĐƯỜNG ĐI CHO ROBOT DI ĐỘNG

3.1 Đề xuất giải pháp cải tiến thuật toán đàn kiến kết hợp thuật toán tìm kiếm theo chiều rộng (ACO-BFS)

Trong quá trình xem xét và tìm hiểu về thuật toán đàn kiến nhận thấy trong chu trình đường đi mà thuật toán ACO tìm được chưa phải là đường đi ngắn nhất. Vậy vấn đề đặt ra là liệu có cách nào để cải tiến thuật toán ACO để đường đi tìm được của các con kiến trong trong thuật toán ngắn hơn không?

Sau khi thực nghiệm và cho ra kết quả, nhận ra rằng trên chu trình đường đi mà thuật toán Ant Colony tìm được có những đoạn đường đi không cần thiết khiến đường đi tìm được dài hơn. Dựa vào đó, ta có thể làm cho đường đi này ngắn hơn bằng cách loại bỏ những những đoạn đường đi thừa này trong quá trình đường đi được tạo ra bởi thuật toán ACO hoặc có thể tìm ra đoạn đường đi ngắn hơn bằng cách loại bỏ các đoạn đường không cần thiết trong chu trình đường đi tìm ra được tạo ra.



Hình 3.1. Đường đi trước và sau khi loại bỏ các đoạn đường không cần thiết

*(Chú thích: Đường màu đỏ là đường trước loại bỏ các đoạn đường không cần thiết, đường màu vàng là đường sau khi loại bỏ các đoạn đường không cần thiết theo ý tưởng)*

Để hiểu rõ hơn ý tưởng thực hiện, ta giả sử có một bản đồ như *Hình 3.1*, con kiến cần di chuyển từ điểm xuất phát 1 đến điểm đích 25. Sau khi di chuyển, con kiến tìm được một đường đi với chu trình 1→2→7→8→13→18→19→20→25. Ta thấy trên bản đồ, điểm có thể đi đến điểm 7 mà không cần thiết đi qua điểm 2, tương tự như vậy với các điểm 7 và 13, 19 và 25. Để làm cho quãng đường di chuyển của kiến ngắn hơn, ta có thể loại bỏ các điểm không cần thiết trong chu trình cũ như 2, 8, 20. Như vậy, chu trình đường đi mới sẽ là 1→7→13→18→19→25.

Nắm bắt được ý tưởng đó, ta có thể sử dụng thuật toán tìm kiếm theo chiều rộng để thực hiện loại bỏ các đoạn không cất thiết sau khi để tạo thành một chu trình mới có độ dài ngắn hơn chu trình mà thuật toán ACO đã tìm ra.

Thuật toán tìm kiếm theo chiều rộng (Breadth First Search – BFS) thuộc nhóm chiến lược tìm kiếm mù. Thuật toán tìm kiếm theo chiều rộng là một thuật toán đệ quy để duyệt qua tất cả các phần tử của cây hoặc đồ thị, sử dụng hàng đợi (queue) để ghi nhớ phần tử liền kề. Từ đó có thể tìm ra đường đi ngắn nhất từ điểm xuất phát đến điểm đích. Ý tưởng và các bước thực hiện của thuật toán tìm kiếm theo chiều rộng đã được phân tích rõ ở *mục 1.4* của chương 1.

Đây là phần cải tiến cơ bản của tôi nhằm giúp tìm ra chu trình đường đi ngắn hơn so với thuật toán ACO. Cải tiến sử dụng thuật toán tối ưu đàn kiến kết hợp với thuật toán tìm kiếm theo chiều rộng.

Bảng 3.1. Thuật toán ACO-BFS

|  |
| --- |
| **Input:** điểm bắt đầu, điểm kết thúc, môi trường hoạt động của Robot  **Procedure Thuật toán ACO-BFS;**  **Begin**   Khởi tạo tham số, ma trận mùi, khởi tạo m con kiến;  **repeat**  **for** k = 1 **to** m **do**                     Kiến k xây dựng lời giải;  **If (**chu trình chứa đoạn đường không cần thiết**)**  Thuật toán BFS                               Cập nhập lại chu trình  **end**  **end-for**  Cập nhật mùi;              Cập nhật lời giải tốt nhất;  **Until (**Điều kiện kết thúc**);**  Đưa ra lời giải tốt nhất;  **End;**  **Output:** Đường đi tối ưu mà thuật toán tìm được |

Ưu điểm và nhược điểm của thuật toán ACO-BFS:

Bảng 3.2. Ưu điểm và nhược điểm của thuật toán ACO-BFS

|  |  |
| --- | --- |
| Ưu điểm | Đường đi tìm được đi qua ít điểm nhất khiến độ dài của đường đi là ngắn nhất. |
| Nhược điểm | * + Do phải duyệt tất cả các điểm, dẫn đến thời gian thực hiện thuật toán tăng lên.   + Tốn nhiều bộ nhớ để lưu trữ các nút. |

3.2 Tối ưu hóa đàn kiến sử dụng xác suất kích thích (Ant Colony Optimization using Stimulating Probability – ACO-SP)

Sau khi xem xét kết quả thực nghiệm, nhận thấy trong quá trình tìm kiếm đường đi, có những con kiến lựa chọn những đoạn đường đi vào ngõ cụt khiến cho chúng không thể đi tiếp hoặc chúng phải quay lại để thoát ra khỏi ngõ cụt, điều đó sẽ làm tăng thêm độ dài của đường đi tìm thấy và giảm đi tốc độ hội tụ của thuật toán khiến cho thời gian thực hiện thuật toán tăng lên. Vậy làm thế nào để kiến biết trước trong đó là ngõ cụt để không đi vào? Để giải quyết vấn đề này, xác suất chọn là thứ cần phải cải tiến để con kiến có thể lựa chọn chính xác hơn, ta sẽ sử dụng một xác suất kích thích (stimulating probability) để con kiến dễ dàng lựa chọn điểm tiếp theo tốt nhất với điều kiện điểm này phải có nhiều hơn một lối ra. Xác suất này phụ thuộc số lượng chướng ngại xung quanh điểm xét *j*. Vì vậy, càng ít chướng ngại vật xung quanh điểm *j* thì xác suất càng lớn và điểm j càng thu hút con kiến [12]. Để xác định được xác suất kích thích của điểm *j*, ta cần có:

* Số cách phân bố của các chướng ngại vật xung quanh điểm *j*, được tính bằng công thức tổ hợp:

(3-1)

Với: là số lượng chướng ngại vật xung quanh j

* Số cách con kiến có thể thoát khỏi điểm *j*, được tính bằng công thức tổ hợp:

(3-2)

Trong công thức (3-2), ta phải loại bỏ đường thoát bằng điểm hiện tại kiến đang đứng *i* để xét điểm *j*.

Công thức tính xác suất kích thích làm kiến k đang đứng ở điểm *i* bị thu hút đến điểm *j* là:

(3-3)

Xác suất chọn sau khi kết hợp với xác suất kích thích sẽ được thay đổi như sau:

(3-4)

Ưu điểm và nhược điểm của ACO-SP:

Bảng 3.3. Ưu điểm và nhược điểm của ACO-SP:

|  |  |
| --- | --- |
| Ưu điểm | * + Giúp con kiến có thể lựa chọn đoạn đường tiếp theo dễ dàng hơn tránh các ngõ cụt (bẫy) hoặc nhanh chóng thoát khỏi ngõ cụt trong lúc tìm kiếm đường đi.   + Đường đi tìm được có thể giữ khoảng cách với chướng ngại vật giúp những robot di động có thể vận chuyển những kiện hàng cồng kềnh một cách an toàn, tránh va đập khi di chuyển trong thực tế. |
| Nhược điểm | Những con kiến có xu hướng chọn những đường đi có ít chướng ngại vật hơn dẫn đến đường đi tìm được có độ dài lớn. |

3.3 Kết hợp thuật toán ACO-BFS với ACO-SP (ACO-BFS-SP)

Thấy được khả năng loại bỏ những đường đi không cần thiết của ACO-BFS và lợi ích trong việc sử dụng xác suất kích thích của ACO-SP để kiến có thể lựa chọn dễ dàng hơn. Ta thực hiện kết hợp hai thuật toán này với nhau, khi đó, xác suất chọn điểm tiếp theo của thuật toán được xác định như sau:

(3-5)

Trong đó:

* : xác suất kích thích trên cạnh 𝑖𝑗

Các bước thực hiện ACO-BFS-SP:

Bảng 3.4. Thuật toán ACO-BFS-SP

|  |
| --- |
| **Input:** điểm bắt đầu, điểm kết thúc, môi trường hoạt động của Robot  **Procedure Thuật toán ACO-BFS-SP;**  **Begin**   Khởi tạo tham số, ma trận mùi, khởi tạo m con kiến;  **repeat**  **for** k = 1 **to** m **do**                     Kiến k xây dựng lời giải;  **If (**chu trình chứa đoạn đường không cần thiết**)**  Thuật toán BFS                               Cập nhập lại chu trình  **end**  **end-for**  Cập nhật mùi;              Cập nhật lời giải tốt nhất;  **Until (**Điều kiện kết thúc**);**  Đưa ra lời giải tốt nhất;  **End;**  **Output:** Đường đi tối ưu mà thuật toán tìm được |

3.4 Tối ưu hóa đàn kiến thông qua điều hướng mục tiêu (Ant Colony Optimization Navigation - ACO-N)

Trong thuật toán ACO, việc lựa chọn theo xác suất có thể khiến những con kiến chọn những điểm tối ưu cục bộ mà không chọn những điểm tối ưu toàn cục. Điều này dẫn đến con kiến có xu hướng đi lang thang và mắc kẹt hoặc làm tăng độ dài của đường đi tìm thấy. Để giải quyết vấn đề này, ta cần điều hướng con kiến đi về phía điểm đích. Nghĩa là con kiến k đứng tại điểm *i*, sẽ chọn một trong các điểm *j* ở xung quanh i làm điểm tiếp theo sẽ đi đến với điều kiện điểm đó phải hướng đến điểm đích. Từ ý tưởng đó, ta có thể cải tiến xác suất chọn của thuật toán ACO bằng số hạng thu hút toàn cục. Mục đích của số hàng này là điều hướng kiến đi về hướng mong muốn để nó có thể đến đích với số bước đi ít nhất [13].

Xác suất chọn khi thêm số hạng sức hút toàn cục là:

(3-6)

Trong đó:

* là tích vô hướng của vector từ *i* đến *j* và vector từ *i* đến điểm kết thúc

Ưu điểm và nhược điểm của ACO-N:

*Bảng 3.5. Ưu điểm và nhược điểm của ACO-N*

|  |  |
| --- | --- |
| Ưu điểm | Khắc phục được tình trạng tối ưu cục bộ khi tìm kiếm đường đi, giúp những con kiến không lang thang nữa. Điều này khiến tốc độ hội tụ của thuật toán tăng lên làm cho thời gian thực hiện thuật toán giảm xuống, đường đi tìm được tối ưu hơn và có độ dài ngắn hơn so với thuật toán ACO tiêu chuẩn. |
| Nhược điểm | Do ảnh hưởng của điều hướng mục tiêu (số hạng thu hút toàn cục), những con kiến có thể đi lọt một số đường đi bẫy và không thể thoát ra hoặc mất thời gian thoát khỏi bẫy, làm thuật toán chậm đi và đường đi tìm được dài hơn. |

3.5 Cải tiến thuật toán ACO-BFS-SP dựa trên ACO-N (ACO-BFS-SP-N)

Khả năng định hướng của ACO-N được thêm vào để khắc phục nhược điểm của xác suất kích thích trong ACO-BFS-SP, đồng thời khả năng đánh giá các điểm lân cận nhờ xác suất kích thích cũng hạn chế nhược điểm của ACO-N trong xử lý các ngõ cụt, bẫy. Ngoài ra, nhờ ưu điểm của khả năng định hướng của ACO-N và xác suất kích thích trong việc tìm đường đi sẽ khắc phục nhược điểm về thời gian thực hiện thuật toán của BFS. Điều này được chứng minh ở các thực nghiệm của chương 4.

Sau khi cải tiến thuật toán ACO-BFS-SP dựa trên ACO-N, xác suất chọn sẽ được xác định như sau:

(3-7)

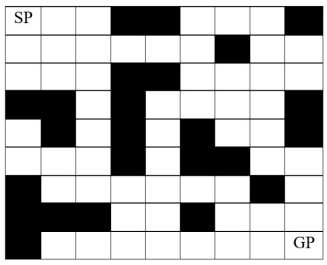
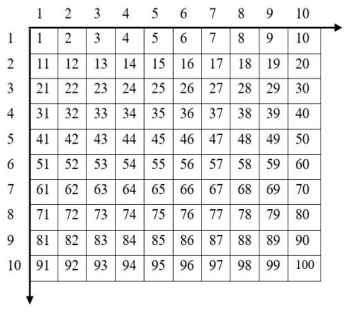
Các bước thực hiện của thuật toán ACO-BFS-SP được giữ nguyên trong thuật toán cải tiến ACO-BFS-SP-N.

CHƯƠNG 4 CÀI ĐẶT THỰC NGHIỆM

4.1 Mô hình hóa môi trường dựa trên lưới

Bước đầu tiên của việc lập kế hoạch đường đi robot di động là thiết lập môi trường cho không gian làm việc 2 chiều (2D) của robot di động. Phương pháp lưới được sử dụng để biểu thị không gian làm việc của robot di động dưới dạng các ô vuông bằng nhau. Mỗi ô lưới chứa một thông tin nhị phân cho biết các ô bị cản trở bằng chướng ngại vật là “1” và các ô trống là “0” như trong *Hình 4.1.a.*

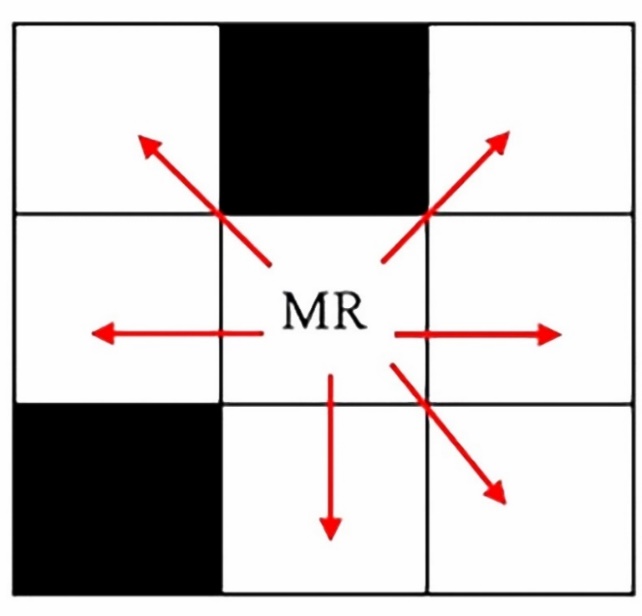
Mỗi ô được xác định bởi một số được gọi là “địa chỉ”. Địa chỉ của ô có thể được xác định bằng phương pháp: tọa độ lưới 2D (r, c), trong đó gốc tọa độ gốc là ô trên cùng bên trái của lưới với vị trí đầu tiên có tọa độ là (1,1) như *Hình 4.1.b.*

(a) (b)

Hình 4.1. Môi trường lưới [5]

Mục tiêu chính là khoảng cách ngắn nhất để robot di chuyển từ vị trí xuất phát đến mục tiêu với điều kiện đường đi đó là an toàn. Tức là robot di chuyển không va chạm chướng ngại vật. Trong mỗi bước robot di động có thể di chuyển từ vị trí hiện tại đến một vị trí khác trong các ô trống xung quanh của nó như *Hình 4.2* [4].

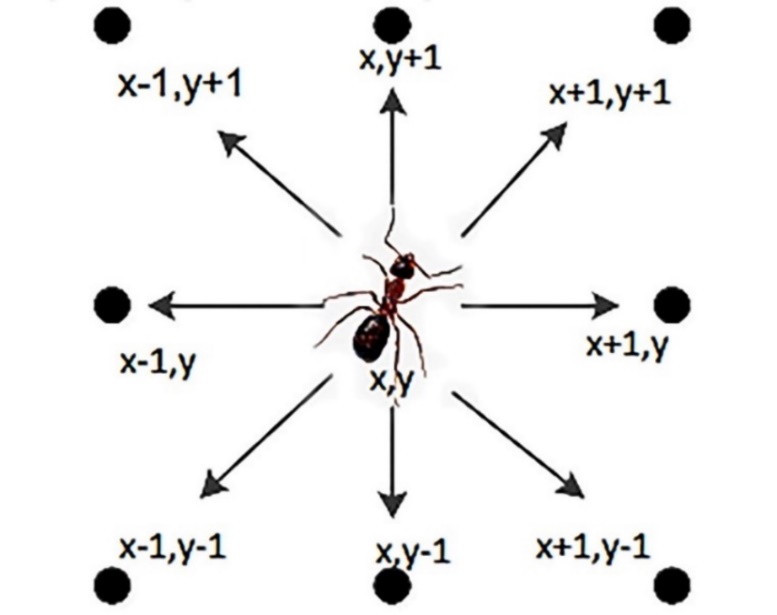


Hình 4.2. Vị trí có thể di chuyển của robot trong môi trường lưới [5]

Môi trường lưới của robot di động được biểu diễn bằng các ô vuông được đánh trọng số, các ô màu trắng là các ô mà robot có thể đi, những ô màu đen biểu diễn cho chướng ngại vật (vật cản) được biểu diễn như *Hình 4.2*.

Giả sử kiến đang đứng ở vị trí (x, y), kiến có thể di chuyển được 8 vị trí xung quanh đó

(x, y+1), (x+1, y+1), (x+1, y), (x+1, y-1), (x, y-1), (x-1, y-1), (x-1, y), (x-1, y+1).



Hình 4.3. Tọa độ vị trí mà kiến có thể di chuyển trong môi trường

4.2 Các hàm mục tiêu

Đồ án này quan tâm đến đường đi có độ dài đường đi là nhỏ nhất và không bị va chạm chướng ngại vật. Độ dài đường đi *Pa* được tính như sau:

Với: Vị trí bắt đầu (), vị trí mục tiêu () của robot

4.3 Cài đặt thực nghiệm

Đề tài thực hiện cài đặt giải thuật ACO-BFS-SP-N để giải quyết bài toán lập kế hoạch đường đi cho robot di động trong môi trường tĩnh có chướng ngại vật.

Để đánh giá giải thuật ACO-BFS-SP-N, đề tài thực hiện hai thực nghiệm sau:

1. Thực nghiệm 1: Phân tích và đánh giá kết quả của giải thuật ACO-BFS-SP-N đối với giải thuật ACO, ACO-BFS và ACO-BFS-SP trên các bộ dữ liệu khác nhau.
2. Thực nghiệm 2: Phân tích sự ảnh hưởng của hình dạng chướng ngại vật lên kết quả của giải thuật ACO-BFS-SP-N
3. Thực nghiệm 3: Phân tích và đánh giá kết quả của giải thuật ACO-BFS-SP-N trong các bộ dữ liệu mô phỏng không gian trong phòng thí nghiệm.

4.4 Dữ liệu thực nghiệm

Dữ liệu được sử dụng trong quá trình thực nghiệm có đặc điểm chung sau:

* Môi trường: Mô hình hóa dạng lưới có kích thước 100 ô được đánh dấu số thứ tự lần lượt.
* Vị trí toạ độ điểm xuất phát và điểm đích.
* Vị trí tọa độ những chướng ngại vật trong môi trường.

Ngoài ra dữ liệu của mỗi thực nghiệm có các đặc điểm riêng để phù hợp với tính chất của thực nghiệm:

* Thực nghiệm 1 gồm 5 bộ dữ liệu có chung điểm xuất phát và điểm đích, trong khi đó số lượng và hình dạng của chướng ngại vật là khác nhau.
* Thực nghiệm 2 gồm 5 bộ dữ liệu có điểm điểm xuất phát và điểm đích khác nhau. Mỗi bộ dữ liệu có một chướng ngại vật có hình dạng đặc biệt.
* Thực nghiệm 3 gồm 3 bộ dữ liệu, mỗi bộ dữ liệu mô phỏng một không gian trong phòng thí nghiệm.

Đề tài tiến hành các thực nghiệm cho từng kịch bản khác nhau, chạy thực nghiệm 30 lần trên từng bộ dữ liệu.

Đề tài đánh giá kết quả thực nghiệm dựa trên các tiêu chí như *Bảng 4.1.*

Bảng 4.1. Các tiêu chí đánh giá

|  |  |  |
| --- | --- | --- |
| Tiêu chí | Đơn vị | Ý nghĩa |
| Độ dài | Đơn vị chiều dài | Độ dài đường đi tìm được |
| Thời gian | Mili giây (ms) | Thời gian chạy giải thuật |
| Tỉ lệ | Phần trăm | Tỷ lệ số lần tìm ra đường đi/số lần chạy |

Thực nghiệm được chạy trên máy tính xách tay có thông số:

Hệ điều hành Windows 10.

Bô vi xử lý: Chip Intel®, Core i5-7200U, CPU 2.50GHz

Dung lượng Ram: 8GB

Các thuật toán được cài đặt và viết bằng ngôn ngữ java với môi trường phát triển tích hợp Eclipse.

Bảng 4.2. Chú thích giao diện khi chạy thực nghiệm

|  |  |
| --- | --- |
| Điểm xuất phát |  |
| Điểm đích |  |
| Chướng ngại vật |  |
| Đường đi của giải thuật ACO |  |
| Đường đi của giải thuật ACO-BFS |  |
| Đường đi của giải thuật ACO-BFS-SP |  |
| Đường đi của giải thuật ACO-BFS-SP-N |  |

4.5 Kết quả thực nghiệm

4.5.1 Thực nghiệm 1: Phân tích và đánh giá kết quả đối với sự thay đổi số lượng chướng ngại vật của giải thuật ACO-BFS-SP-N so với giải thuật ACO, ACO-BFS và ACO-BFS-SP trên các bộ dữ liệu khác nhau.

Đề tài thực hiện kịch bản thực nghiệm 1 để phân tích kết quả của giải thuật ACO-BFS-SP-N ở các bộ dữ liệu có môi trường với số lượng chướng ngại vật khác nhau, từ đó so sánh với giải thuật ACO, ACO-BFS và ACO-BFS-SP.

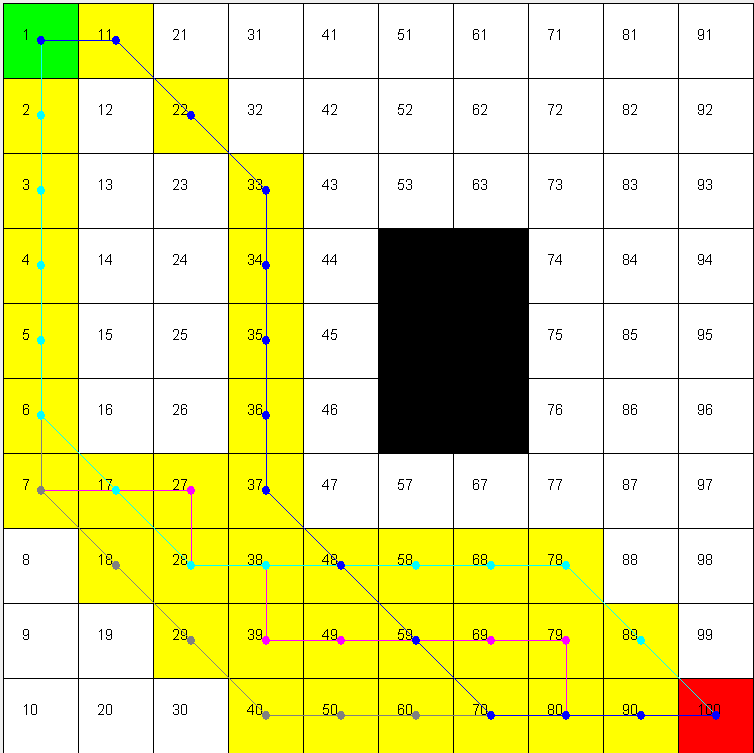
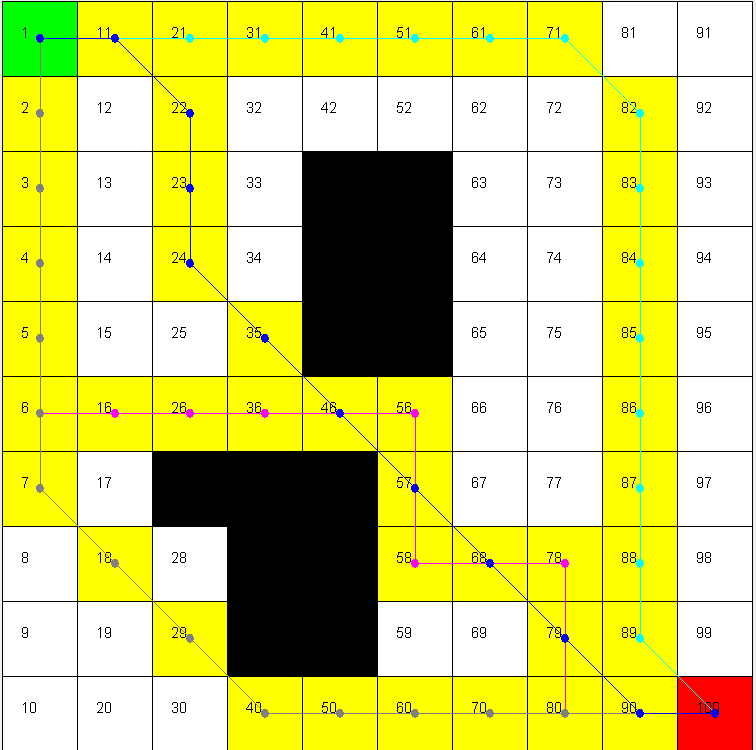
Dữ liệu của thực nghiệm 1 bao gồm 5 bộ dữ liệu, trong đó 5 bộ dữ liệu có chung điểm xuất phát và điểm đích nhưng khác nhau về số lượng và vị trí chướng ngại vật.

Các tham số của giải thuật ACO-BFS-SP-N sử dụng trong thực nghiệm 1 được cài đặt và trình bày trong *Bảng 4.3.*

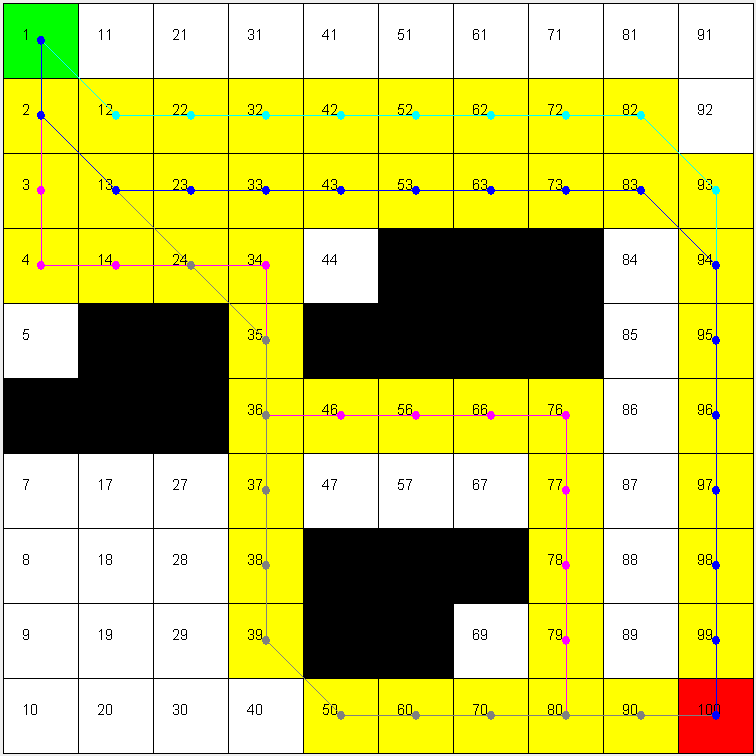
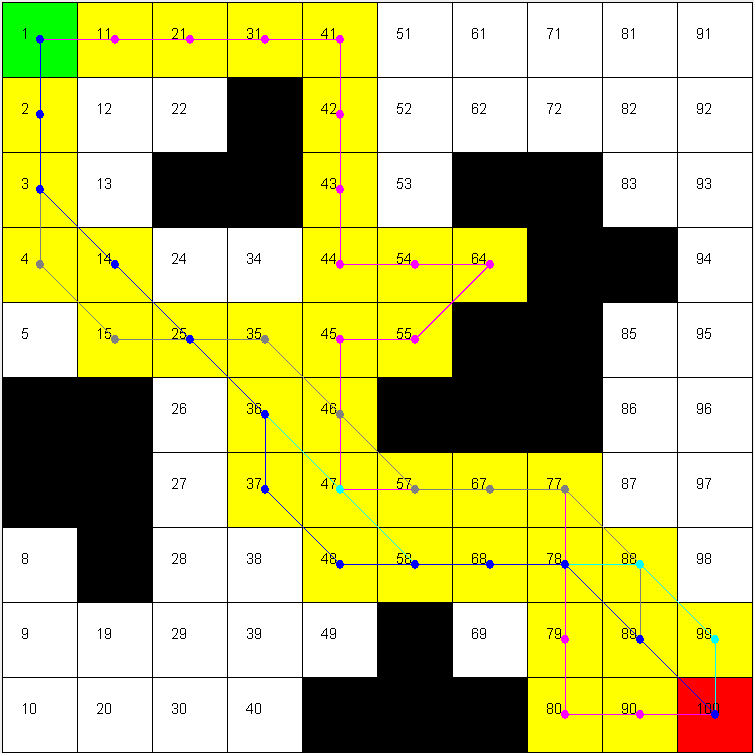
Bảng 4.3. Tham số cài đặt trên từng bộ dữ liệu được cài đặt trong thực nghiệm 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tham số thực nghiệm | Bộ dữ liệu 1 | Bộ dữ liệu 2 | Bộ dữ liệu 3 | Bộ dữ liệu 4 | Bộ dữ liệu 5 |
| Số con kiến tham gia | 50 | 50 | 50 | 50 | 50 |
| Kích thước ma trận | 10x10 | 10x10 | 10x10 | 10x10 | 10x10 |
| Số lượng chướng ngại vật | 1 | 2 | 3 | 5 | 10 |
| 𝜌: tốc độ bay hơi pheromone | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 |
| *Q*: tham số điều chỉnh lượng pheromone | 1 | 1 | 1 | 1 | 1 |
| α: tham số kiểm soát ảnh hưởng của 𝜏i,j | 1 | 1 | 1 | 1 | 1 |
| β: tham số kiểm soát ảnh hưởng của của ηi,j | 5 | 5 | 5 | 5 | 5 |

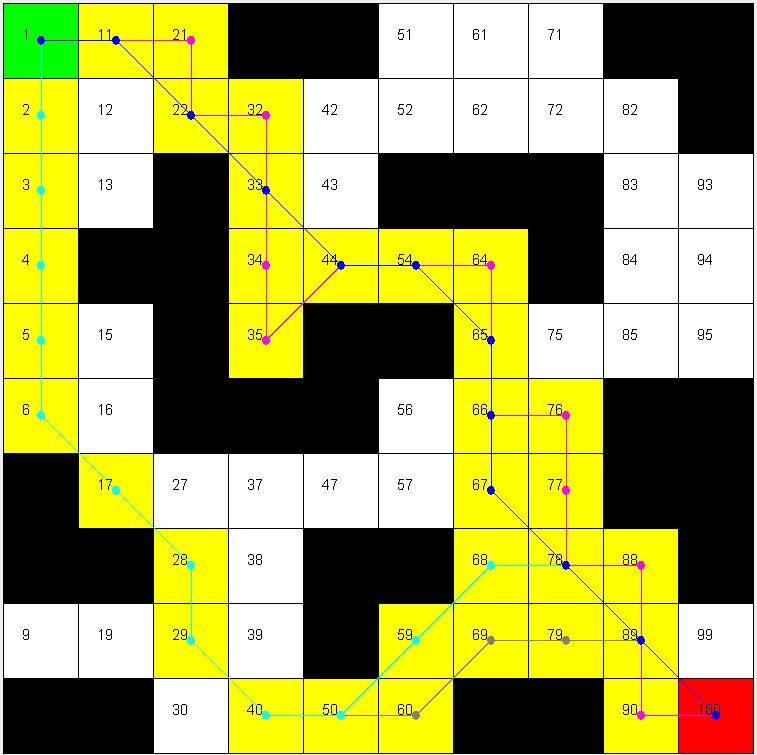
Chạy thực nghiệm 1 thu được kết quả của từng bộ dữ liệu và mô tả qua *Hình 4.4.*

1. (b)

(c) (d)



(e)

Hình 4.4. Hình minh họa kết quả thực nghiệm 1:  Phân tích và đánh giá kết quả của thuật toán ACO-BFS-SP-N đối với thuật toán ACO, ACO-BFS và ACO-BFS-SP trên các bộ dữ liệu khác nhau.

*(Chú thích: Đường màu xanh dương – Giải thuật ACO-BFS-SP-N, đường màu hồng – Giải thuật ACO, đường màu xám – Giải thuật ACO-BFS, đường màu xanh lơ – Giải thuật ACO-BFS-SP)*

1. *Bộ dữ liệu 1: 1 chướng ngại vật*
2. *Bộ dữ liệu 2: 2 chướng ngại vật*
3. *Bộ dữ liệu 3: 3 chướng ngại vật*
4. *Bộ dữ liệu 4: 4 chướng ngại vật*
5. *Bộ dữ liệu 5: 9 chướng ngại vật*

Kết quả chạy thực nghiệm 1 của thuật toán ACO-BFS-SP-N được trình bày trong *Bảng 4.4.*

Bảng 4.4. Kết quả thực nghiệm 1: phân tích và đánh giá kết quả của thuật toán ACO-BFS-SP-N đối với thuật toán ACO, ACO-BFS và ACO-BFS-SP trên các bộ dữ liệu khác nhau.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bộ dữ liệu | Thuật toán | Thời gian chạy trung bình | Độ dài đường đi trung bình tìm được | Tỉ lệ thành công |
| Bộ dữ liệu 1  (1 chướng ngại vật – chiếm 6% môi trường) | ACO | 36.2 | 18.5333 | 100.00% |
| ACO-BFS | 39.4667 | 16.8069 | 100.00% |
| ACO-BFS-SP | 38.7333 | 16.6965 | 100.00% |
| ACO-BFS-SP-N | 28.3667 | 15.0711 | 100.00% |
| Bộ dữ liệu 2  (2 chướng ngại vật – chiếm 13% môi trường) | ACO | 46.2667 | 19 | 100.00% |
| ACO-BFS | 48.8 | 16.8479 | 100.00% |
| ACO-BFS-SP | 48.9667 | 17.1098 | 100.00% |
| ACO-BFS-SP-N | 37.8667 | 13.8995 | 100.00% |
| Bộ dữ liệu 3  (3 chướng ngại vật – chiếm 17% môi trường) | ACO | 60.5 | 20.6966 | 100.00% |
| ACO-BFS | 56.6667 | 15.9498 | 100.00% |
| ACO-BFS-SP | 15.4333 | 17.1846 | 100.00% |
| ACO-BFS-SP-N | 9.3667 | 16.8284 | 100.00% |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bộ dữ liệu 4  (4 chướng ngại vật – chiếm 21% môi trường) | ACO | 15.1333 | 19.1805 | 100.00% |
| ACO-BFS | 15.7 | 15.1391 | 100.00% |
| ACO-BFS-SP | 8.8 | 17.6521 | 100.00% |
| ACO-BFS-SP-N | 7.7667 | 14.4853 | 100.00% |
| Bộ dữ liệu 5  (10 chướng ngại vật – chiếm 33% môi trường) | ACO | 14.9333 | 20.0219 | 100.00% |
| ACO-BFS | 19.6 | 16.562 | 100.00% |
| ACO-BFS-SP | 14.6667 | 17.8995 | 100.00% |
| ACO-BFS-SP-N | 11.4333 | 14.9238 | 100.00% |

Kết quả từ *Bảng 4.4.* cho thấy thuật toán ACO-BFS-SP-N có thể chạy tốt, ổn định trong các môi trường có cùng điểm xuất phát, điểm đích và khác nhau về số lượng và vị trí của chướng ngại vật.

Dựa vào thời gian chạy thực, độ dài đường đi tìm được, trong 30 lần chạy, trung bình độ dài đường đi tại các bộ dữ liệu có sự chênh lệch rõ rệt, tương đồng với sự chênh lệch thời gian chạy thực nghiệm. Từ đó, nhận xét thuật toán ACO-BFS-SP-N cho thấy sự cải tiến về cả thời gian chạy thực và độ dài đường đi tìm được.

Thay đổi số lượng chướng ngại vật trong thuật toán:

* + Với số lượng chướng ngại vật ít: làm tăng phạm vi tìm kiếm đường đi, khiến số lượng điểm cần xét để lựa chọn tăng lên, dẫn đến, thời gian chạy thực chậm. Tuy nhiên, đường đi tìm được của thuật toán ACO-BFS-SP-N vẫn tối ưu hơn và có thời gian chạy thực nhanh hơn so với các thuật toán còn lại.
  + Với số lượng chướng ngại vật nhiều: làm giảm phạm vi tìm kiếm đường đi, xét về mặt tổng thể, các thuật toán đều tìm được đường đi có đội dài ngắn và thời gian chạy thực nhanh. Nhưng, đường đi tìm được của thuật toán ACO-BFS-SP-N vẫn tối ưu hơn, nổi trội hơn về độ dài đường đi tìm được và thời gian chạy thực so với các thuật toán còn lại.

Tỷ lệ lần tìm ra đường đi khả thi, tránh được các chướng ngại vật trong tổng số lần chạy của các thuật toán ở mức cao, luôn được giữ ở mức 100%.

4.5.2 Thực nghiệm 2: Phân tích sự ảnh hưởng hình dạng chướng ngại vật lên kết quả trong giải thuật ACO-BFS-SP-N

Đề tài thực hiện thực nghiệm 2: phân tích sự ảnh hưởng hình dạng chướng ngại vật lên kết quả thuật toán ACO-BFS-SP-N, với các hình dạng cụ thể như chướng ngại vật hình lồi, hình lõm.

Dữ liệu thực nghiệm 2 bao gồm các hình dạng chướng ngại vật đặc biệt (U, E, J, H, G)

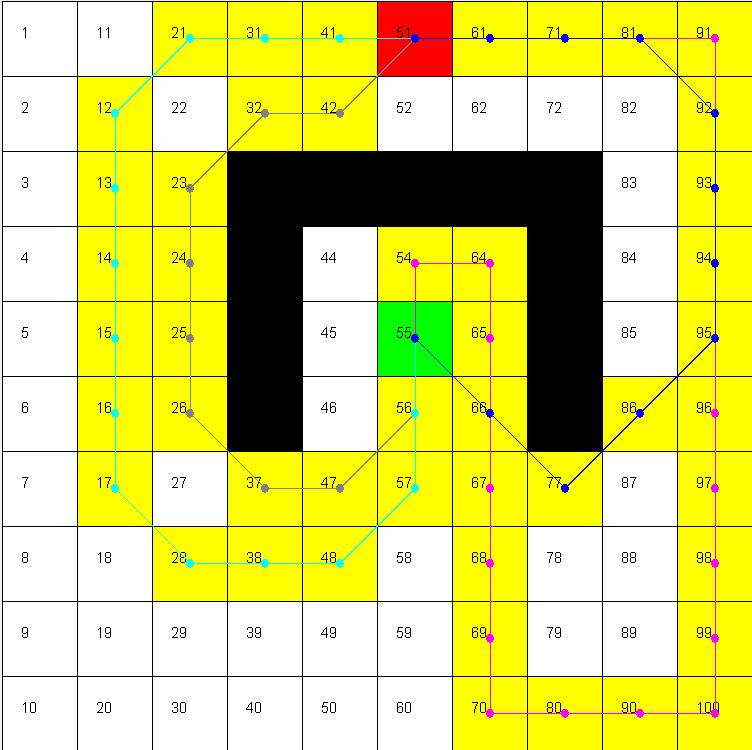
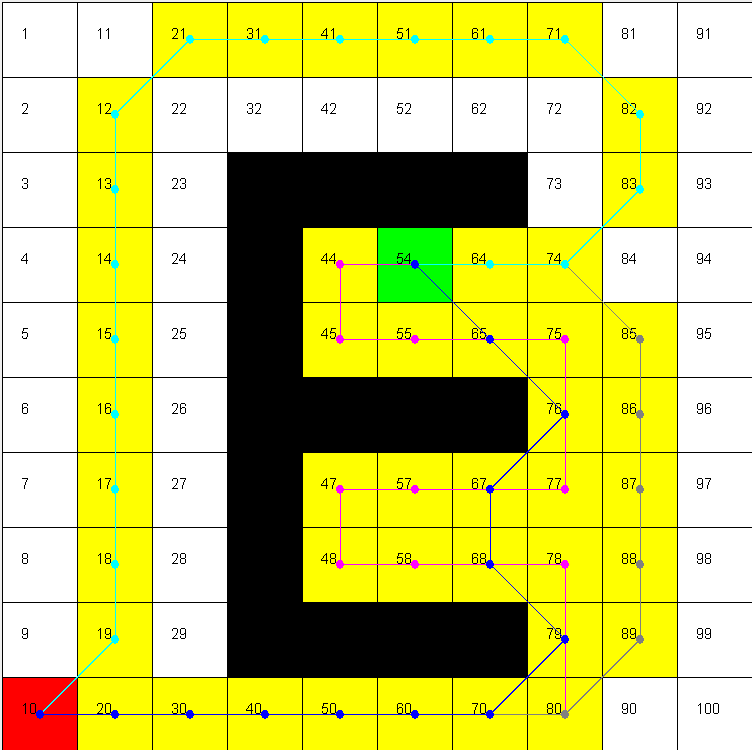
có điểm xuất phát và điểm đích khác nhau.

Các thông số thực nghiệm được thể hiện qua *Bảng 4.5.*

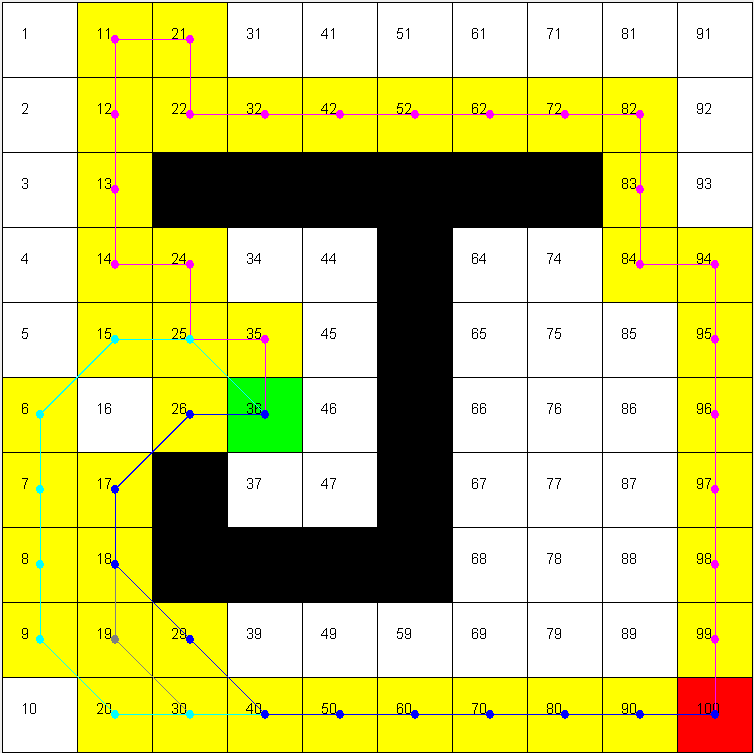
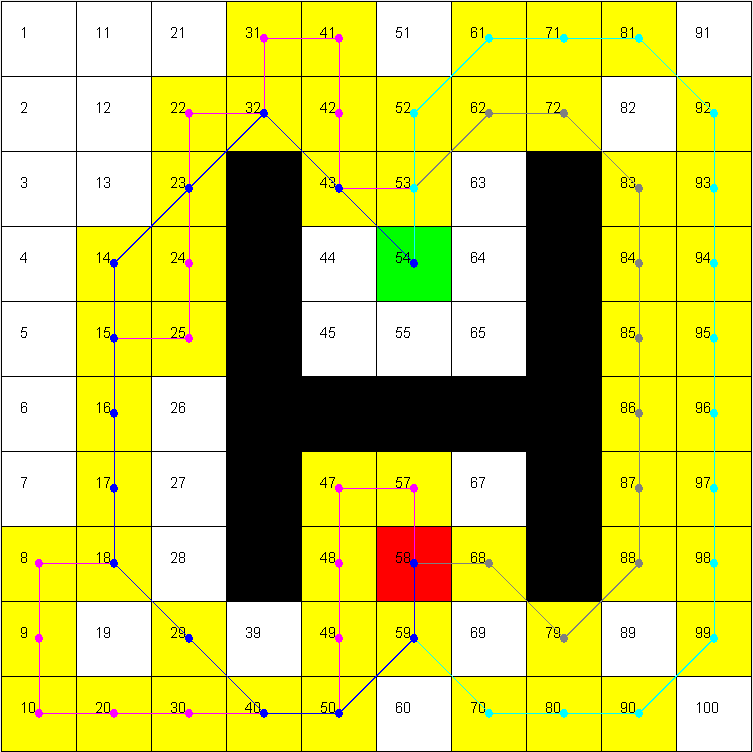
Bảng 4.5. Tham số cài đặt trên từng bộ dữ liệu được cài đặt trong thực nghiệm 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tham số thực nghiệm | Bộ dữ liệu 1 | Bộ dữ liệu 2 | Bộ dữ liệu 3 | Bộ dữ liệu 4 | Bộ dữ liệu 5 |
| Số con kiến tham gia | 50 | 50 | 50 | 50 | 50 |
| Kích thước ma trận | 10x10 | 10x10 | 10x10 | 10x10 | 10x10 |
| Hình dạng chướng ngại vật | U | E | J | H | G |
| 𝜌: tốc độ bay hơi pheromone | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 |
| *Q*: tham số điều chỉnh lượng pheromone | 1 | 1 | 1 | 1 | 1 |
| α: parameter kiểm soát ảnh hưởng của 𝜏i,j | 1 | 1 | 1 | 1 | 1 |
| β: tham số kiểm soát ảnh hưởng của của ηi,j | 5 | 5 | 5 | 5 | 5 |

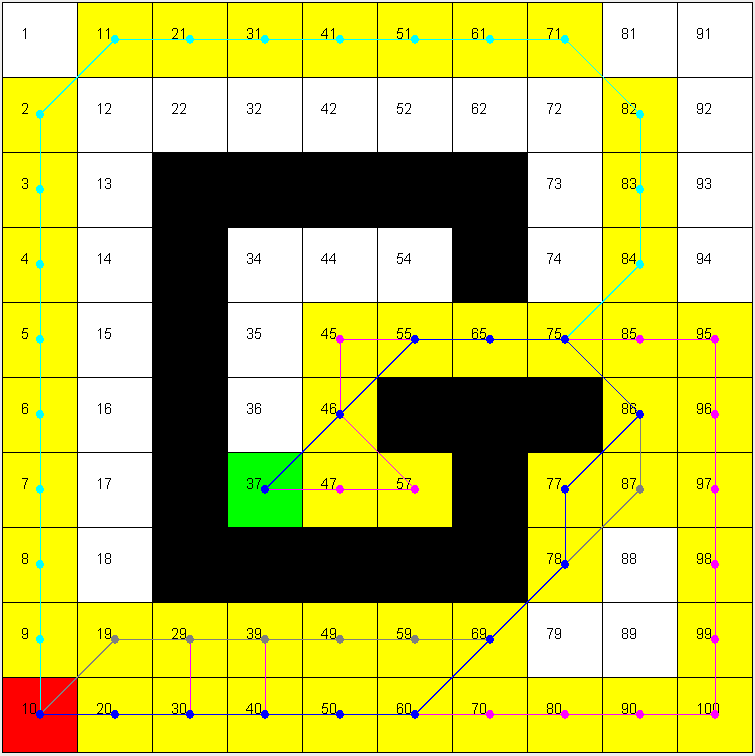
Chạy thực nghiệm 2 thu được kết quả thể hiện qua *Hình 4.5.*

1. (b)

(c) (d)



(e)

Hình 4.5. Hình minh họa kết quả thực nghiệm 2: phân tích sự ảnh hưởng hình dạng chướng ngại vật lên kết quả giải thuật ACO-BFS-SP-N

*(Chú thích: Đường màu xanh dương – Giải thuật ACO-BFS-SP-N, đường màu hồng – Giải thuật ACO, đường màu xám – Giải thuật ACO-BFS, đường màu xanh lơ – Giải thuật ACO-BFS-SP)*

1. *Bộ dữ liệu 1 chướng ngại vật hình chữ U*
2. *Bộ dữ liệu 2 chướng ngại vật hình chữ E*
3. *Bộ dữ liệu 3 chướng ngại vật hình chữ J*
4. *Bộ dữ liệu 4 chướng ngại vật hình chữ H*
5. *Bộ dữ liệu 4 chướng ngại vật hình chữ G*

Kết quả thực nghiệm 2 được thể hiện qua *Bảng 4.6.*

Bảng 4.6. Kết quả thực nghiệm 2: phân tích sự ảnh hưởng hình dạng chướng ngại vật lên kết quả giải thuật ACO-BFS-SP-N

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bộ dữ liệu | Giải thuật | Thời gian chạy trung bình | Độ dài đường đi trung bình tìm được | Tỉ lệ thành công |
| Bộ dữ liệu 1 (chướng ngại vật hình chữ U) | ACO | 14.6667 | 17.0667 | 100.00% |
| ACO-BFS | 22.5333 | 14.4731 | 100.00% |
| ACO-BFS-SP | 11.2 | 18.1188 | 100.00% |
| ACO-BFS-SP-N | 5.4667 | 13.0711 | 100.00% |
| Bộ dữ liệu 2 (chướng ngại vật hình chữ E) | ACO | 13.1936 | 20.2903 | 100.00% |
| ACO-BFS | 17.6667 | 17.0164 | 100.00% |
| ACO-BFS-SP | 17.3571 | 21.7405 | 100.00% |
| ACO-BFS-SP-N | 5.6667 | 14.0711 | 100.00% |
| Bộ dữ liệu 3 (chướng ngại vật hình chữ J) | ACO | 16.4 | 18.6667 | 100.00% |
| ACO-BFS | 17.7333 | 12.3598 | 100.00% |
| ACO-BFS-SP | 11.2 | 16.2426 | 100.00% |
| ACO-BFS-SP-N | 5.9333 | 12.2426 | 100.00% |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bộ dữ liệu 4 (chướng ngại vật hình chữ H) | ACO | 14.7333 | 20 | 100.00% |
| ACO-BFS | 19.3333 | 13.8455 | 100.00% |
| ACO-BFS-SP | 11.1333 | 19.6569 | 100.00% |
| ACO-BFS-SP-N | 8.2667 | 16.0977 | 100.00% |
| Bộ dữ liệu 5 (chướng ngại vật hình chữ G) | ACO | 17.3333 | 23.0828 | 100.00% |
| ACO-BFS | 18.0667 | 19.6206 | 100.00% |
| ACO-BFS-SP | 7.8 | 25.0711 | 100.00% |
| ACO-BFS-SP-N | 6.2 | 16.4853 | 100.00% |

Kết quả thực nghiệm 2 trên 5 bộ dữ liệu có chướng ngại vật với các hình dạng đặc biệt (U, E, J, H, G) chạy trên thuật toán ACO-BFS-SP-N cho thấy:

Trong môi trường, chướng ngại vật có bẫy với mức độ lõm và độ phức tạp khác nhau: đường đi tìm được của thuật toán ACO-BFS-SP-N có thời gian thoát khỏi bẫy nhanh và độ dài ngắn hơn so với các thuật toán còn lại. Ta có thể thấy rõ, thuật toán tìm ra đường đi tối ưu hơn so với thuật toán ACO,

Thông qua bảng dữ liệu và hình ảnh kết quả thu được, nhận xét với các hình dạng đặc biệt như hình chứng ngại vật U, J, E, H, G: với các chướng ngại vật có bẫy, thuật toán xử lý và chạy tương đối tốt với tỷ lệ chạy thực nghiệm thành công cao, có thời gian chạy thực và độ dài đường đi trung bình chênh lệch khá rõ ràng tại giải thuật ACO-BFS-SP-N và giải thuật ACO. Tuy nhiên, mức độ ảnh hưởng của ACO-SP trong thuật toán ACO-BFS-SP-N thấp hơn so với thuật toán ACO-BFS-SP do mức độ ảnh hưởng của sự định hướng của ACO-N cao hơn, ta có thấy rõ trong *Hình 4.5.b*. Nhưng ACO-SP lại giúp thuật toán ACO-BFS-SP-N khắc phục nhược điểm của ACO-N. Vì vậy, thuật toán cải tiến vẫn chưa thể cân bằng cùng lúc độ định hướng của ACO-N và khả năng lựa chọn điểm có ít chướng ngại vật xung quanh nhất của ACO-SP. Tuy vậy, kết quả của của thuật toán ACO-BFS-SP-N vẫn có thể chấp nhận được.

Chạy thực nghiệm 5 bộ dữ liệu, chạy mỗi bộ dữ liệu 30 lần với tỉ lệ chạy thực nghiệm thành công là 100%.

4.5.3 Thực nghiệm 3: Phân tích và đánh giá kết quả của giải thuật ACO-BFS-SP-N trong các bộ dữ liệu mô phỏng không gian trong phòng thí nghiệm

Đề tài thực hiện thực nghiệm 3: phân tích và đánh giá kết quả của giải thuật ACO-BFS-SP-N trong các bộ dữ liệu mô phỏng không gian phòng thí nghiệm

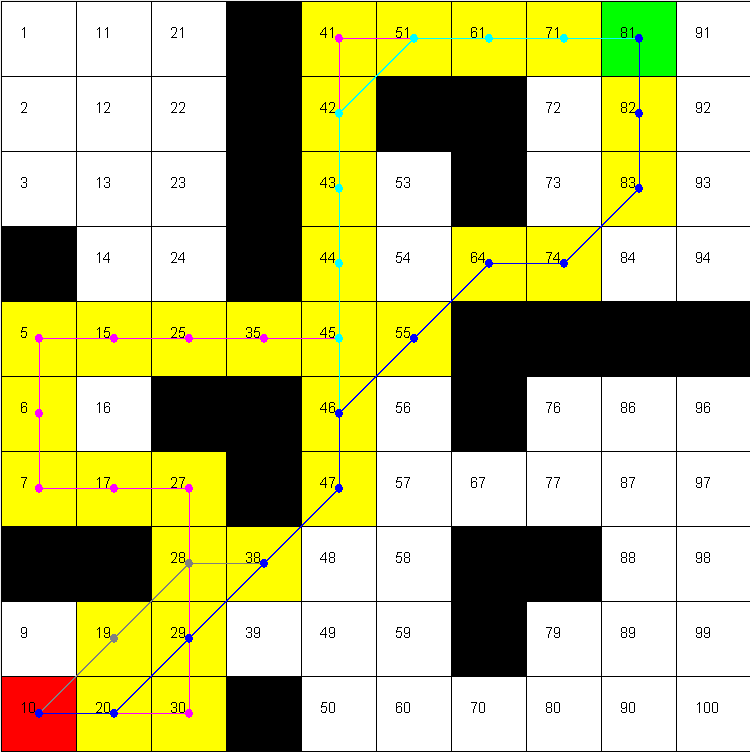
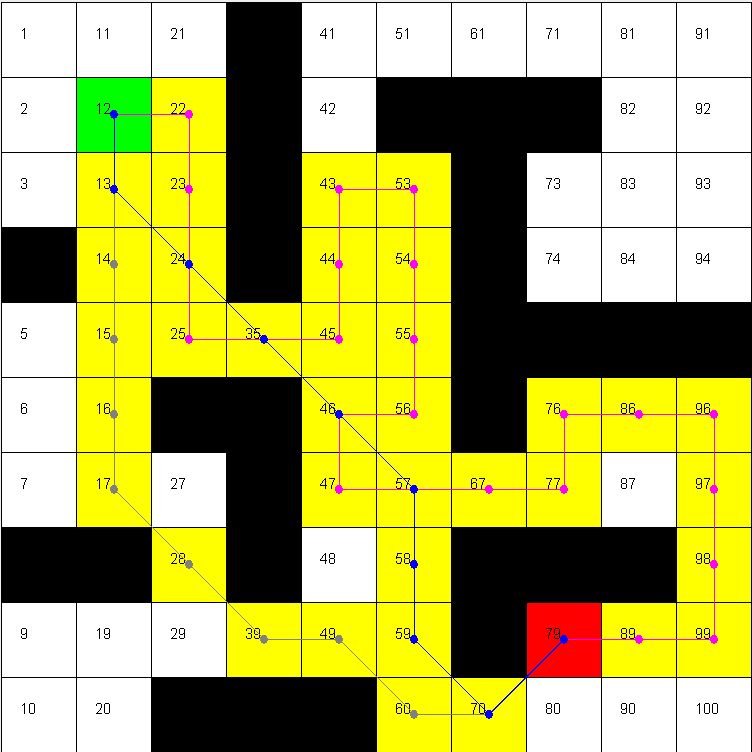
Dữ liệu thực nghiệm 3 bao gồm 3 bộ dữ liệu khác nhau, mỗi bộ dữ liệu mô phỏng một không gian trong phòng thí nghiệm có điểm xuất phát, điểm đích, số lượng chướng ngại vật và hình dạng chướng ngại vật khác nhau.

Các thông số thực nghiệm được thể hiện qua *Bảng 4.7.*

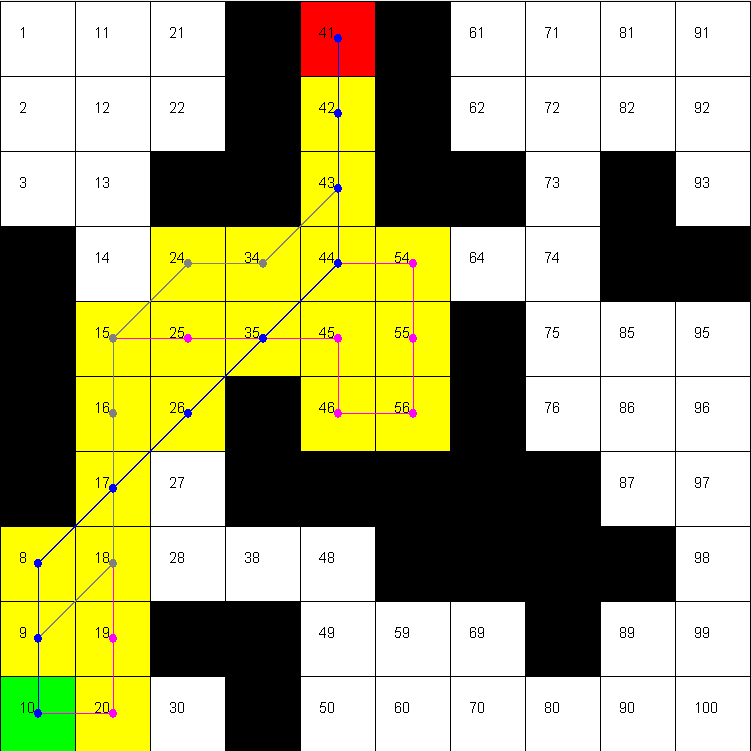
Bảng 4.7. Tham số cài đặt trên từng bộ dữ liệu được cài đặt trong thực nghiệm 3

|  |  |  |  |
| --- | --- | --- | --- |
| Tham số thực nghiệm | Bộ dữ liệu 1 | Bộ dữ liệu 2 | Bộ dữ liệu 3 |
| Số con kiến tham gia | 50 | 50 | 50 |
| Kích thước ma trận | 10x10 | 10x10 | 10x10 |
| Số lượng chướng ngại vật | 8 | 7 | 6 |
| 𝜌: tốc độ bay hơi pheromone | 0.5 | 0.5 | 0.5 |
| *Q*: tham số điều chỉnh lượng pheromone | 1 | 1 | 1 |
| α: tham số kiểm soát ảnh hưởng của 𝜏i,j | 1 | 1 | 1 |
| β: tham số kiểm soát ảnh hưởng của của ηi,j | 5 | 5 | 5 |

Chạy thực nghiệm 3 thu được kết quả của từng bộ dữ liệu và mô tả qua *Hình 4.6.*

1. (b)



(c)

Hình 4.6. Hình minh họa kết quả thực nghiệm 3: Phân tích và đánh giá kết quả của giải thuật ACO-BFS-SP-N trong các bộ dữ liệu mô phỏng không gian trong phòng thí nghiệm

*(Chú thích: Đường màu xanh dương – Giải thuật ACO-BFS-SP-N, đường màu hồng – Giải thuật ACO, đường màu xám – Giải thuật ACO-BFS, đường màu xanh lơ – Giải thuật ACO-BFS-SP)*

1. *Bộ dữ liệu 1: phòng 1 với 8 chướng ngại vật*
2. *Bộ dữ liệu 2: phòng 2 với 7 chướng ngại vật*
3. *Bộ dữ liệu 3: phòng 3 với 6 chướng ngại vật*

Kết quả thực nghiệm 3 được thể hiện qua *Bảng 4.8.*

Bảng 4.8. Kết quả thực nghiệm 3: phân tích và đánh giá kết quả của giải thuật

ACO-BFS-SP-N trong các bộ dữ liệu mô phỏng không gian phòng thí nghiệm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bộ dữ liệu | Giải thuật | Thời gian chạy trung bình | Độ dài đường đi trung bình tìm được | Tỉ lệ thành công |
| Bộ dữ liệu 1 (8 chướng ngại vật – chiếm 21% môi trường) | ACO | 16.6333 | 21.9333 | 100.00% |
| ACO-BFS | 18.3667 | 14.6845 | 100.00% |
| ACO-BFS-SP | 18.2667 | 14.6569 | 100.00% |
| ACO-BFS-SP-N | 8.7667 | 13.48537 | 100.00% |
| Bộ dữ liệu 2 (7 chướng ngại vật – chiếm 28% môi trường) | ACO | 13.6 | 18.2667 | 100.00% |
| ACO-BFS | 13.7 | 12.6569 | 100.00% |
| ACO-BFS-SP | Ø | Ø | 0.00% |
| ACO-BFS-SP-N | 10.5667 | 11.4853 | 100.00% |
| Bộ dữ liệu 3 (6 chướng ngại vật – chiếm 31% môi trường) | ACO | 14.0795 | 16.5667 | 100.00% |
| ACO-BFS | 20.6333 | 11.0669 | 100.00% |
| ACO-BFS-SP | Ø | Ø | 0.00% |
| ACO-BFS-SP-N | 7.2667 | 10.6569 | 100.00% |

Từ kết quả của thực nghiệm 3, ta thấy thuật toán ACO-BFS-SP-N có thể chạy tốt, ổn định trong mỗi bộ dữ liệu mô phỏng một không gian trong phòng thí nghiệm có điểm xuất phát, điểm đích, số lượng chướng ngại vật, vị trí và hình dạng chướng ngại vật khác nhau. Đường đi mà giải thuật ACO-BFS-SP-N tìm ra tối ưu hơn, độ dài ngắn hơn và tốn ít thời gian thực hiện hơn so với các giải thuật còn lại.

Thông qua hình ảnh và dữ liệu kết quả thu được, thuật toán ACO-BFS-SP-N đã khắc phục được nhược điểm về khả năng chọn điểm của xác suất kích thích. Trong khi ACO-BFS-SP có xu hướng chọn những điểm có ít chướng ngại vật xung quanh dẫn đến không tìm ra được đường đi tới đích thì giải thuật ACO-BFS-SP-N chọn những điểm hướng về điểm đích và có ít chướng ngại vật xung quanh nhất.

Chạy thực nghiệm giải thuật ACO-BFS-SP-N trên 3 bộ dữ liệu, chạy mỗi bộ dữ liệu 30 lần với tỉ lệ chạy thực nghiệm thành công là 100%.

Sau quá trình chạy thực nghiệm chương trình nhận thấy rằng: việc cải tiến thuật toán tối ưu đàn kiến bằng cách loại bỏ những điểm không cần thiết trên đường đi bằng cách kết hợp với thuật toán tìm kiếm theo chiều rộng thu được chu trình đường đi có độ dài ngắn hơn. Cùng lúc cải tiến thêm bằng cách sử dụng xác suất kích thích giúp cho kiến có thể chọn những điểm có ít chướng ngại vật xung quanh nhằm giảm các trường hợp đi vào ngõ cụt và giúp cho những robot có thế di chuyển cách tường một đoạn để tránh va chạm trong thực tế. Đồng thời kết hợp với khả năng điều hướng mục tiêu để kiến không đi lang thang giúp đường đi tìm được ngắn hơn và giảm thời gian thực hiện thuật toán. Điều này đã được thực nghiệm nhiều lần và cho ra kết quả khả quan.

Quan sát từ các hình thu và bảng biểu thu được cho thấy kết quả cải tiến thông qua:

* + Độ dài đường đi: đường đi tìm được của giải thuật ACO-BFS-SP-N có độ dài ngắn hơn so với giải thuật ACO, giải thuật ACO-BFS và giải thuật ACO-BFS-SP ở tất cả các bộ dữ liệu.
  + Thời gian chạy thuật toán: giải thuật ACO-BFS-SP-N có thời gian chạy ít nhất trong toàn bộ các bộ dữ liệu của cả 3 thực nghiệm. Trong hầu hết các bộ dữ liệu thực nghiệm, thuật toán ACO có chu trình di chuyển dài nhất dẫn đến mất nhiều thời gian do những con kiến đi lang thang để tìm tới đích.
  + Tỷ lệ số lần tìm được đường đi: giải thuật ACO-BFS-SP-N, giải thuật ACO và giải thuật ACO-BFS đều có tỷ lệ tìm được đường đi bằng 100%. Trong một số bộ dữ liệu, giải thuật ACO-BFS-SP không tìm thấy đường đi do có xu hướng chọn những đoạn đường ít chướng ngại vật dẫn đến không thể tìm tới đích thì giải thuật ACO-BFS-SP-N đã giải quyết được nhược điểm này.

Kết luận

Đề tài để xuất được giải thuật mới ACO-BFS-SP-N kế thừa nhiều giải thuật khác nhau. Trong đó, nền tảng là giải thuật tối ưu đàn kiến, phương pháp mô hình hóa môi trường dạng phân rã ô.

Đề tài đã trình bày nội dung:

* Cơ sở lý thuyết về robot di động và ứng dụng cùng thách thức của chúng, mô hình hóa môi trường, cùng giải thuật tối ưu đàn kiến.
* Giới thiệu bài toán lập kế hoạch đường đi cho robot di động trong môi trường tĩnh có chướng ngại vật và định hướng giải quyết bài toán, các nghiên cứu liên quan.
* Trình bày về các đề xuất cải tiến giải thuật, đề xuất thuật toán tối ưu đàn kiến với thuật toán tìm kiếm theo chiều rộng (ACO-BFS); tối ưu hóa đàn kiến sử dụng xác suất kích thích (ACO-SP); tối ưu hóa đàn kiến thông qua điều hướng mục tiêu (ACO-N). Giải thuật đề xuất là sự kết hợp của các thuật toán trên với nhau.
* Thực hiện, đánh giá giải thuật cải tiến dựa trên các thực nghiệm khác nhau.
* So sánh kết quả giải thuật ACO-BFS-SP-N với các giải thuật khác.

Kết quả thực nghiệm cho thấy, thuật toán ACO-BFS-SP-N có khả năng tìm được đường đi tối ưu trong khoảng thời gian ngắn với độ hiệu quả cao, trong các môi trường có điểm xuất phát và điểm đích, chướng ngại vật khác nhau. Mặc dù không phải lần thực hiện nào cũng tìm được đường đi khả thi, nhưng tỷ lệ tìm được đường đi khả thi vẫn ở mức độ cao.

Tuy nhiên giải thuật ACO-BFS-SP-N vẫn phụ thuộc nhiều vào các yếu tố như số lượng chướng ngại vật hay hình dáng chướng ngại vật. Việc chạy thực nghiệm với kết quả so sánh cũng cho thấy giải thuật ACO-BFS-SP-N tốt hơn giải thuật ACO, giải thuật ACO-BFS và giải thuật ACO-BFS-SP về thời gian chạy, độ dài và độ tối ưu của chu trình đường đi tìm được.

Thông qua nghiên cứu của đề tài, tác giả đưa ra đề xuất giải phải cải tiến kết hợp tối ưu hóa bầy kiến với đột biến bằng thuật toán tiến hóa vi sai để tìm được đường đi ngắn nhất trong môi trường.

Trong tương lai, tác giả sẽ tìm hiểu và nghiên cứu các phương pháp mô hình hóa môi trường, thuật toán khác phù hợp hơn. Cũng mong muốn có thể tối ưu đường đi với nhiều mục tiêu khác, ví dụ như độ mượt, độ an toàn, giúp bài toán có thể đáp ứng được yêu cầu với các thực nghiệm khó hay chính là các yêu cầu trong thực tế. Ngoài ra sẽ tìm kiếm các nghiên cứu, thuật toán giải bài toán tương tự để so sánh với giải thuật ACO-BFS-SP-N.

Tài liệu tham khảo

|  |  |
| --- | --- |
| [1] | Alatortsev, S., Stellmacher, S., & Ortmeier, F. (2015). Robotic task sequencing problem: A survey. *Journal of intelligent & robotic systems*, *80*(2), 279-298. |
| [2] | Han-ye Zhang, Wei-ming Lin, and Ai-xia Chen. Path planning for the mobile robot: A review. Symmetry, 10(10):450, 2018. |
| [3] | Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. *IEEE computational intelligence magazine*, *1*(4), 28-39. |
| [4] | Canny, J. (1988). *The complexity of robot motion planning*. MIT press. |
| [5] | Ajeil, F. H., Ibraheem, I. K., Azar, A. T., & Humaidi, A. J. (2020). Grid-based mobile robot path planning using aging - based ant colony optimization algorithm in static and dynamic environments. *Sensors*, *20*(7), 1880. |
| [6] | Ning, J., Zhang, C., Sun, P., & Feng, Y. (2019). Comparative study of ant colony algorithms for multi-objective optimization. *Information*, *10*(1), 11. |
| [7] | Sariff, N., & Buniyamin, N. (2006). An Overview of Autonomous Mobile Robot Path Planning Algorithms. *2006 4th Student Conference on Research and Development*, 183-188. |
| [8] | Wikipedia contributors. Motion planning Wikipedia, the free encyclopedia, 2021. [Online; accessed 11-June-2021] |
| [9] | Cai, K., Wang, C., Cheng, J., De Silva, C. W., & Meng, M. Q. H. (2020). Mobile Robot Path Planning in Dynamic Environments: A Survey. *arXiv preprint arXiv:2006.14195*. |
| [10] | BK Patle, Anish Pandey, DRK Parhi, A Jagadeesh, et al. A review: On path planning strategies for navigation of mobile robot. Defence Technology, 15(4):582606, 2019. |
| [11] | Trần Thị Cẩm Giang (2020). Trí tuệ nhân tạo (Curriculum). |
| [12] | Khaled Akka and Farid Khaber (2018). Mobile robot path planning using an improved ant colony optimization. *International Journal of Advanced Robotic Systems*. 1-7. |
| [13] | D R Parhi and J K Pothal (2011). Intelligent navigation of multiple mobile robots using an ant colony optimization technique in a highly cluttered environment. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science.* |
| [14] | Q Vu Duc. (2017). *Ap dung thuat toan toi uu hoa dan kien đe giai quyet bai toan vi tri co so* (Doctoral dissertation). |
| [15] | Guanyu Wang (2018). A Comparative Study of Cuckoo Algorithm and Ant Colony Algorithm in Optimal Path Problems. *MATEC Web of Conferences 232, 03003 (2018).* |
| [16] | D Do Duc (2013). *Phương pháp tối ưu đàn kiến và ứng dụng* (Doctoral dissertation). |
| [17] | Shekhawat, A., Poddar, P., & Boswal, D. (2009). Ant colony optimization algorithms: introduction and beyond. In *Artificial Intelligence seminar, Indian Institute of Technology, Bombay–India*. |
| [18] | Shi, Y. (Ed.). (2012). *Innovations and Developments of Swarm Intelligence Applications*. IGI Global. |
| [19] | Alpkiray, N., Torun, Y., & Kaynar, O. (2018). Probabilistic Roadmap and Artificial Bee Colony Algorithm Cooperation For Path Planning. *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, 1-6. |
| [20] | Rainer Storn and Kenneth Price. Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization, 11(4):341359, 1997. |