

# So sánh thuật toán ACO và các thuật toán tìm đường kinh điển trên bản đồ lưới cho robot di động

## Abstract

Bài báo này so sánh hiệu năng của thuật toán Ant Colony Optimization (ACO) với các thuật toán tìm đường kinh điển bao gồm Depth-First Search (DFS), Breadth-First Search (BFS), và A\* trên bản đồ lưới kích thước 50x50 với mật độ chướng ngại vật 20%. Các tiêu chí so sánh bao gồm tỉ lệ thành công, độ dài đường đi trung bình và thời gian tính toán. Kết quả cho thấy ACO và A\* thể hiện hiệu suất vượt trội so với DFS và BFS, đặc biệt là trong độ dài đường đi và thời gian tính toán.

## I. Introduction

Tìm đường đi cho robot di động trong môi trường phức tạp là một bài toán cốt lõi trong lĩnh vực robotics. Thuật toán ACO dựa trên trí tuệ bầy đàn, lấy cảm hứng từ hành vi kiếm ăn của kiến, được sử dụng rộng rãi trong các bài toán tối ưu phức tạp. Báo cáo này nhằm mục đích phân tích hiệu quả của ACO so với các thuật toán kinh điển DFS, BFS, và A\*.

## II. Literature review

Robot di động có thể được định hướng bằng các thuật toán định vị và tìm đường khác nhau như được đề cập bởi Siegwart và Nourbakhsh (2004). Các thuật toán tìm đường truyền thống như DFS, BFS và A\* đều đã được chứng minh tính hiệu quả trong nhiều trường hợp (Hart et al., 1968). Tuy nhiên, gần đây, các phương pháp lấy cảm hứng sinh học như ACO (Bonabeau, 2003; Dorigo và Gambardella, 1997) được chú ý vì khả năng tối ưu cao trong môi trường động và không chắc chắn.

## III. Methodology

### A. Bản đồ dạng lưới

Bản đồ thử nghiệm có kích thước 50x50 với mật độ chướng ngại vật 20%. Điểm xuất phát và điểm đích lần lượt là góc trái trên cùng và góc phải dưới cùng của bản đồ.

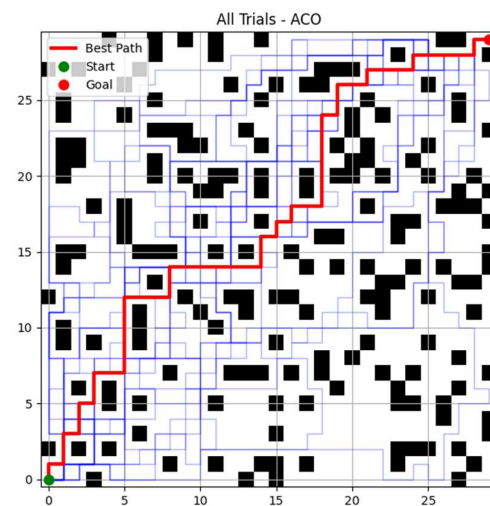
### B. Các thuật toán so sánh

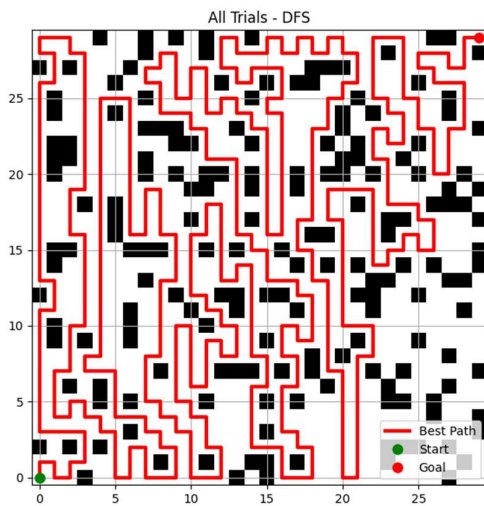
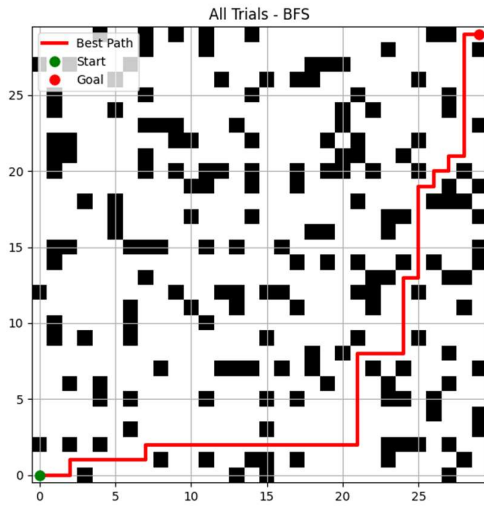
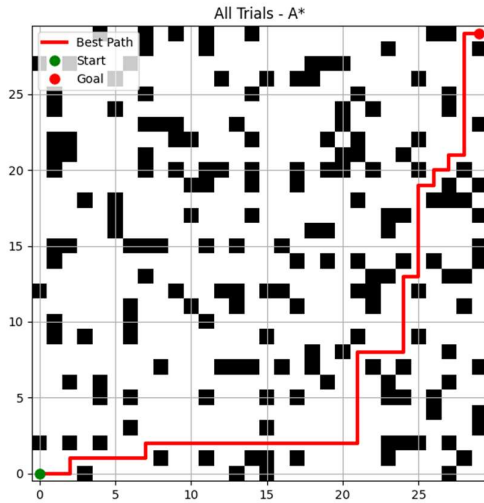
- **ACO**: sử dụng 50 kiến và 50 vòng lặp.
- **A\***: heuristic khoảng cách Euclidean.
- **BFS và DFS**: tìm kiếm dựa trên cơ sở dữ liệu dạng queue và stack tương ứng.

Các thuật toán được thực hiện 50 lần chạy độc lập và được đánh giá dựa trên tỉ lệ thành công, chiều dài đường đi và thời gian thực hiện.

## IV. Experiments

Các thử nghiệm được tiến hành trên một máy tính tiêu chuẩn. Kết quả được ghi nhận và tổng hợp dưới dạng bảng CSV để phân tích chi tiết.





## V. Results and Discussion

Kết quả thống kê như sau:

Algorithm	Success Rate (%)	Avg Path Length	Avg Time (s)
ACO	100	96.34	3.0445
A*	100	61.93	0.0058
BFS	100	58	0.0027
DFS	72	476	0.0052

Các hình ảnh minh họa trực quan hóa đường đi tốt nhất của mỗi thuật toán được đính kèm, thể hiện sự hiệu quả rõ ràng của thuật toán A\* và ACO so với DFS và BFS.

## VI. Conclusion and future work

Kết quả nghiên cứu chỉ ra rằng ACO và A\* thích hợp cho các bài toán cần hiệu quả cao, trong khi BFS có thể là lựa chọn dự phòng đáng tin cậy. DFS ít hữu ích trong thực tế do hiệu suất thấp. Nghiên cứu tương lai có thể hướng đến cải thiện hiệu suất thuật toán ACO thông qua tối ưu tham số và các biến thể thuật toán mới.

## References

- Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press.
- Dorigo, M., & Gambardella, L. M. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1), 53-66.
- Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2), 100-107.

- Siegwart, R., & Nourbakhsh, I. R. (2004). Introduction to Autonomous Mobile Robots. MIT Press.

Đường dẫn mã nguồn và dữ liệu thống kê:  
[https://github.com/Quagn/Project\\_Distributed\\_Intelligent\\_Systems.git](https://github.com/Quagn/Project_Distributed_Intelligent_Systems.git)