

Optimal and Smooth Mobile Robot Path Planning Using GAN, A*, and Cubic Spline Interpolation

Thi-Minh-Tam Le¹, The-Thanh Bui², Van-Luong Dang¹, Duc-Hung Pham^{1*},
Ngoc-Thang Pham¹

¹Hung Yen University of Technology and Education, Vietnam

²Hanoi Industrial and Trade University, Vietnam

*Corresponding author. Email: duchung.pham@utehy.edu.vn

ARTICLE INFO

Received: 24/07/2025
Revised: 29/10/2025
Accepted: 09/02/2026
Published: 28/02/2026

KEYWORDS

Generative adversarial network (GAN);
Cubic spline interpolation;
Neural network;
Robot route planning;
Complex maze.

ABSTRACT

This paper presents a trajectory-planning method for mobile robots that integrates a Generative Adversarial Network (GAN) with grid-based A*. The GAN generator samples obstacle coordinates while enforcing a 1.0-unit clearance and masking forbidden regions around the start (1,1) and goal (14,14). The workspace is discretized into a 30×30, 8-connected lattice; A* with an admissible and consistent Euclidean heuristic returns the globally optimal lattice path, which is subsequently converted into a smooth geometric trajectory via cubic-spline interpolation. On a 15×15 maze with 60 obstacles, the proposed GAN&A* pipeline achieves a path length of 19.26 units, improving over a Sparrow Search Algorithm baseline (21.8 units). To assess scalability, we further evaluate a 20×20 maze with 120 obstacles. Under identical collision models and smoothing, GAN&A* attains 28.81 units, outperforming two sampling-based planners RRT (30.63 units) and PRM (30.51 units). These results indicate that learned environment synthesis coupled with optimal lattice search yields reliable, short, and smooth trajectories, whereas RRT/PRM require substantially larger sampling budgets (RRT*/PRM*) to approach comparable quality.

Hoạch định đường đi tối ưu và làm mượt cho Robot di động sử dụng GAN, A*, và nội suy Spline khối

Lê Thị Minh Tâm¹, Bùi Thế Thành², Đặng Văn Lượng¹, Phạm Đức Hùng^{1*},
Phạm Ngọc Thăng¹

¹Trường Đại học Sư phạm Kỹ thuật Hưng Yên, Việt Nam

²Trường Đại học Công nghiệp và Thương mại Hà Nội, Việt Nam

*Tác giả liên hệ. Email: duchung.pham@utehy.edu.vn

THÔNG TIN BÀI BÁO

Ngày nhận bài: 24/07/2025
Ngày hoàn thiện: 29/10/2025
Ngày chấp nhận đăng: 09/02/2026
Ngày đăng: 28/02/2026

TỪ KHÓA

Mạng sinh đối kháng;
Nội suy spline khối;
Mạng nơ-ron;
Lập kế hoạch lộ trình robot;
Mê cung phức tạp.

TÓM TẮT

Bài báo trình bày một phương pháp lập kế hoạch quỹ đạo cho robot di động bằng cách tích hợp Mạng Đối Sinh (GAN) với A* trên lưới. Bộ sinh của GAN lấy mẫu tọa độ vật cản đồng thời áp đặt khoảng hở an toàn 1,0 đơn vị và che (mask) các vùng cấm quanh điểm xuất phát (1,1) và đích (14,14). Không gian làm việc được rời rạc hóa thành lưới 30×30 liên thông 8; A* với heuristic Euclid có khả năng nhất quán cho đường đi tối ưu toàn cục trên lưới, sau đó được chuyển thành quỹ đạo hình học mượt bằng nội suy spline bậc ba. Trên mê cung 15×15 với 60 vật cản, pipeline GAN&A* đạt độ dài đường đi 19,26 đơn vị, cải thiện so với chuẩn SSA (21,8 đơn vị). Để đánh giá khả năng mở rộng, chúng tôi tiếp tục thử nghiệm trên mê cung 20×20 với 120 vật cản. Với cùng mô hình va chạm và cùng bước làm mượt, GAN&A* đạt 28,81 đơn vị, vượt hai bộ lập kế hoạch dựa trên lấy mẫu RRT (30,63 đơn vị) và PRM (30,51 đơn vị). Kết quả cho thấy tổng hợp môi trường có học kết hợp tìm kiếm tối ưu trên lưới tạo ra quỹ đạo tin cậy, ngắn và mượt; trong khi RRT/PRM cần chi phí mẫu lớn hơn (RRT*/PRM*) để tiệm cận chất lượng tương đương.

Doi: <https://doi.org/10.54644/jte.2026.1970>

Copyright © JTE. This is an open access article distributed under the terms and conditions of the [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/) which permits unrestricted use, distribution, and reproduction in any medium for non-commercial purpose, provided the original work is properly cited.

1. Giới thiệu

Trong bối cảnh cách mạng công nghiệp 4.0, rô-bốt di động giữ vai trò then chốt trong sản xuất tự động hóa, logistics thông minh và dịch vụ, kéo theo nhu cầu các thuật toán lập kế hoạch quỹ đạo hiệu quả nhằm tối ưu hiệu suất, độ chính xác và tiêu thụ năng lượng. Bài toán lập kế hoạch đường đi trong môi trường phức tạp với nhiều vật cản là thách thức điển hình của tối ưu hóa tổ hợp (NP-hard), đòi hỏi các phương pháp vừa khả thi tính toán vừa thích nghi tốt với kịch bản thực tế. Nhằm xây dựng các kịch bản thử nghiệm phức tạp nhưng có kiểm soát để đánh giá thuật toán, các mô hình sinh dữ liệu đặc biệt là GAN được xem như một hướng tiếp cận phù hợp nhờ khả năng học phân bố và sinh mẫu mới theo dữ liệu và điều kiện ràng buộc [1], đồng thời đã được tổng quan và hệ thống hóa trong nhiều khảo cứu gần đây [2]. Trên nền tảng đó, nhiều nghiên cứu đã khai thác GAN để sinh môi trường phục vụ lập kế hoạch đường đi cho rô-bốt, giúp tạo bản đồ/không gian vật cản đa dạng, tăng độ khó nhưng vẫn đảm bảo tính khả thi của bài toán [3]. Một số hướng tiếp cận tập trung vào việc tạo môi trường phức tạp (nhiều cấu trúc vật cản) để tăng tính thuyết phục của đánh giá thuật toán lập kế hoạch quỹ đạo [4], hoặc cải tiến GAN nhằm mô phỏng tốt hơn môi trường động cho điều hướng tự hành [5]. Đáng chú ý, các khung lai giữa mô hình sinh môi trường và thuật toán tìm đường cổ điển cũng đã được đề xuất, ví dụ kết hợp GAN với A^* để giải bài toán tìm đường trong mê cung phức tạp [6]. Bên cạnh đó, các mô hình sinh sâu còn được mở rộng để hỗ trợ lập kế hoạch trong các không gian có chiều cao và biểu diễn phức tạp hơn [7], và các biến thể GAN tiên tiến tiếp tục được phát triển nhằm nâng cao chất lượng môi trường tổng hợp cho hệ tự hành [8]. Ở hướng tích hợp theo tác vụ, một số nghiên cứu dùng GAN để tăng cường khả năng lập kế hoạch trong bối cảnh có chướng ngại và động học thay đổi, nhấn mạnh tránh vật cản động và tính thích nghi theo thời gian [9], trong khi các nghiên cứu khác tập trung vào sinh bản đồ có phân bố vật cản được điều khiển để phục vụ điều hướng ổn định và đánh giá có kiểm soát [10]. Ngoài GAN chuẩn, Conditional GAN cũng được khai thác để sinh mê cung và môi trường “thực tế hơn” theo điều kiện đặt trước, giúp kiểm soát tốt cấu trúc bài toán khi thử nghiệm thuật toán rô-bốt [11]. Song song với hướng môi trường sinh, về phía thuật toán, các metaheuristic như Sparrow Search Algorithm (SSA) được ứng dụng rộng rãi trong tối ưu hóa và đã được nghiên cứu như một lựa chọn cho lập kế hoạch, nhưng thường đối mặt nguy cơ mắc kẹt cục bộ hoặc suy giảm hiệu năng khi môi trường động/phức tạp [12]. Ở phía cảm nhận, các hệ thống thị giác phục vụ tránh vật cản cho thấy tiềm năng xử lý thời gian thực, là nền tảng cho định vị và lập kế hoạch tiên tiến trong robot/tự hành [13]. Bên cạnh đó, các bộ lập kế hoạch dựa trên lấy mẫu như PRM và RRT cùng các biến thể gần đây vẫn là chuẩn tham chiếu mạnh, tuy chất lượng đường đi phụ thuộc đáng kể vào ngân sách mẫu/kết nối và tinh chỉnh tham số; các đại diện tiêu biểu có thể kể đến PRM-D* [14], các biến thể dựa trên RRT cho môi trường phức tạp [15], và các cải tiến kiểu sampling-improved như FF-RRT* nhằm xử lý các dạng vật cản và địa hình khó [16]. Trên cơ sở đó, bài báo này đề xuất một phương pháp thiết lập quỹ đạo cho rô-bốt di động kết hợp GAN + A^* và nội suy spline khối trên bản đồ mê cung 15×15 với 60 vật cản hình chữ nhật. GAN được dùng để tạo phân bố vật cản ngẫu nhiên nhưng có kiểm soát, bảo đảm khoảng cách tối thiểu 1.0 đơn vị và loại trừ vùng cấm quanh điểm xuất phát (1,1) và đích (14,14). Bộ sinh có kiến trúc MLP ba tầng (Input 100 chiều \rightarrow Dense 128-ReLU \rightarrow Dense 64-ReLU \rightarrow Dense 2) tạo tọa độ trong miền $(x [2, 13.5], y [2, 13.5])$, đồng thời duy trì hành lang $\geq 1.5-2.0$ đơn vị để tăng tính khả thi. Không gian được rời rạc hóa thành lưới 30×30 , liên thông 8; A^* với heuristic phù hợp tính đường đi tối ưu trên lưới, sau đó spline khối chuyên thành quỹ đạo hình học mượt, tránh bám sát biên vật cản. Thử nghiệm Python cho thấy đường đi 19.26 ngắn hơn SSA (21.89), minh chứng lợi ích của việc kết hợp môi trường sinh có kiểm soát (GAN) với tìm kiếm tối ưu trên lưới (A^*) và làm mượt quỹ đạo, đồng thời đặt nền cho so sánh có hệ thống với các bộ lập kế hoạch dựa trên lấy mẫu như PRM và RRT trong các cấu hình mở rộng.

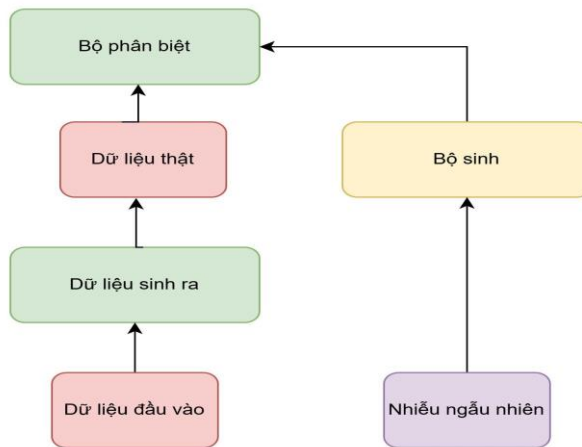
2. Thuật ngữ

2.1. Mạng noron đối kháng (GAN)

Mạng noron đối kháng là một lớp mô hình học sâu được giới thiệu bởi Ian Goodfellow và các cộng sự vào năm 2014, được thiết kế để tạo ra dữ liệu mới tương tự như dữ liệu huấn luyện. GAN bao gồm hai mô hình chính: Generator (Bộ sinh) và Discriminator (Bộ phân biệt). Generator tạo ra dữ liệu giả từ nhiễu ngẫu nhiên, trong khi Discriminator đánh giá xem dữ liệu đầu vào là thật từ tập dữ liệu thực hay giả do Generator tạo ra. Generator và Discriminator được huấn luyện đồng thời theo tính chất đối kháng, trong đó Generator tạo dữ liệu giả để đánh lừa Discriminator, còn Discriminator học cách phân biệt chính xác dữ liệu thật với dữ liệu giả do Generator sinh ra. Quá trình này được mô tả bởi hàm giá trị:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}} [\log(D(x))] - \mathbb{E}_{z \sim p_z} [\log(1 - D(G(z)))] \quad (1)$$

Cấu trúc của mạng GAN được minh họa như Hình 1.



Hình 1. Cấu trúc tổng thể của bộ GAN.

2.2. Thuật toán A*

Thuật toán A* (A-star) là một kỹ thuật tìm kiếm có định hướng được ứng dụng rộng rãi trong lĩnh vực trí tuệ nhân tạo và điều khiển robot, đặc biệt trong các bài toán định tuyến tối ưu trên không gian đồ thị. Phương pháp này khai thác đồng thời hai yếu tố then chốt: chi phí tích lũy từ điểm khởi đầu đến vị trí hiện tại và giá trị ước lượng còn lại từ vị trí hiện tại đến đích. Việc kết hợp giữa dữ liệu thực nghiệm và thông tin dự đoán này cho phép A* đạt hiệu quả cao trong việc xác định đường đi tối ưu với độ chính xác lớn và thời gian xử lý ngắn.

Thuật toán A* dựa trên cấu trúc đánh giá gồm ba thành phần chính: $g(n)$ đại diện cho chi phí đã tiêu tốn từ điểm xuất phát đến nút n hiện tại, $h(n)$ là hàm heuristic dùng để ước lượng chi phí từ nút n đến mục tiêu cuối, $f(n)$: là hàm đánh giá tổng thể được dùng để lựa chọn nút tiếp theo, với công thức:

$$f(n) = g(n) + h(n) \quad (2)$$

2.3. Nội suy Spline

Đây là một kỹ thuật toán học mạnh mẽ để xây dựng các đường cong và bề mặt trơn tru đi qua một tập hợp các điểm dữ liệu cho trước, được gọi là "điểm nút" (knots). Khác với phương pháp nội suy đa thức toàn cục thường tạo ra các dao động không mong muốn, Spline được cấu thành từ nhiều đa thức bậc thấp thường là bậc 3 - Cubic Spline được kết nối một cách liên tục tại các điểm nút. Tính liên tục này không chỉ dừng lại ở giá trị hàm số vị trí C^0 mà còn có thể đảm bảo cả đạo hàm bậc một vận tốc C^1 và đạo hàm bậc hai gia tốc C^2 , tạo nên các đường cong cực kỳ mượt mà. Nhờ đặc tính ưu việt này, nội suy Spline có ứng dụng rộng rãi trong đồ họa máy tính, thiết kế hỗ trợ bởi máy tính CAD, và đặc biệt là trong lập trình quỹ đạo chuyển động cho robot. Về bản chất toán học, nội suy Spline là một hàm từng

khúc (piecewise-defined function), thường được biểu diễn bằng các đa thức bậc ba, tạo thành một đường cong liên tục. Các đặc điểm chính bao gồm:

(1) **Tính từng đoạn:** Trên mỗi khoảng giữa hai điểm nút liên tiếp $[x_i, x_{i+1}]$, đường cong được định nghĩa bởi một đa thức riêng biệt, thường là bậc ba $S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$. Điều này giúp tránh hiện tượng Runge's thường gặp ở nội suy đa thức bậc cao toàn cục.

(2) **Tính liên tục:** Đây là yếu tố then chốt. Một Spline bậc ba đảm bảo tính liên tục tại các điểm nút trên nhiều cấp độ:

C^0 - **Liên tục vị trí:** Giá trị của hai đa thức liên kế bằng nhau tại điểm nút: $S_i(x_{i+1}) = S_{i+1}(x_{i+1})$.

C^1 - **Liên tục đạo hàm bậc nhất (Vận tốc):** Đạo hàm bậc nhất (góc tiếp tuyến) là liên tục: $S'_i(x_{i+1}) = S'_{i+1}(x_{i+1})$.

C^2 - **Liên tục đạo hàm bậc hai (Gia tốc):** Đạo hàm bậc hai là liên tục: $S''_i(x_{i+1}) = S''_{i+1}(x_{i+1})$.

(3) **Điều kiện biên:** Để hệ phương trình có nghiệm duy nhất, các điều kiện biên phải được áp dụng. Các dạng phổ biến bao gồm quy định giá trị đạo hàm bậc nhất tại hai điểm nút đầu và cuối. Và quy định đạo hàm bậc hai tại hai điểm nút đầu và cuối bằng 0.

(4) **Hệ số xác định:** Các hệ số a_i, b_i, c_i, d_i cho mỗi đoạn được xác định bằng cách giải một hệ phương trình tuyến tính thu được từ các điều kiện nêu trên. Hệ này luôn có nghiệm duy nhất và có thể giải hiệu quả nhờ ma trận thưa và dạng đường chéo trội.

Như vậy, cấu trúc toán học dựa trên tính từng khúc và sự đảm bảo về tính liên tục đến bậc hai chính là nền tảng tạo nên độ trơn mượt và khả năng ứng dụng rộng rãi của Spline trong kỹ thuật.

2.4. Thuật toán SSA

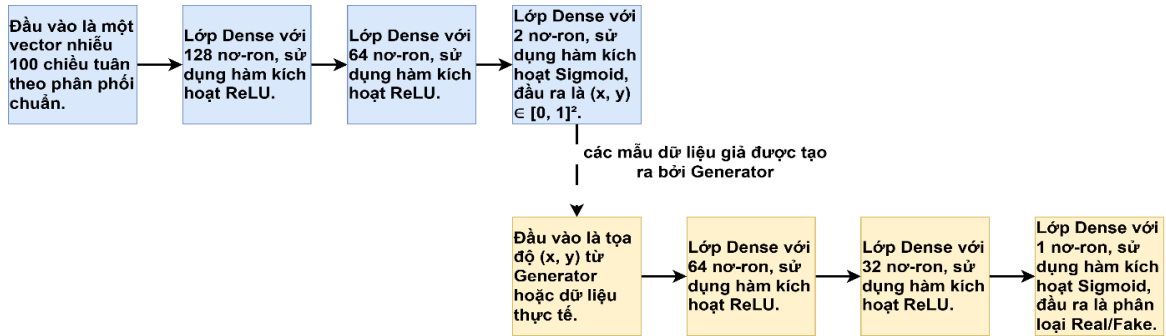
Thuật toán tìm kiếm chim sẻ (Sparrow Search Algorithm - SSA) [12] là một phương pháp tối ưu hóa lấy cảm hứng từ tự nhiên, mô phỏng hành vi kiếm ăn của loài chim sẻ. Thuật toán này được thiết kế để giải quyết các bài toán tối ưu phức tạp bằng cách cân bằng giữa khai phá (exploration) và khai thác (exploitation) không gian tìm kiếm. Để đơn giản hóa, hành vi của chim sẻ được lý tưởng hóa như sau và các quy tắc tương ứng được xây dựng: Các cá thể producers (chim dẫn đầu) thường có mức năng lượng dự trữ cao và chịu trách nhiệm chỉ ra khu vực hoặc hướng đi đến nguồn thức ăn cho các scroungers (chim ăn theo). Họ có vai trò xác định các vùng có nguồn thức ăn dồi dào. Mức năng lượng dự trữ của mỗi cá thể phụ thuộc vào giá trị đánh giá mức độ thích nghi (fitness) của chúng. Khi chim sẻ phát hiện kẻ săn mồi, các cá thể trong đàn bắt đầu phát ra tiếng kêu như một tín hiệu cảnh báo. Nếu giá trị cảnh báo vượt quá ngưỡng an toàn, các producer sẽ phải dẫn tất cả các scrounger đến khu vực an toàn. Mỗi con chim sẻ đều có khả năng trở thành producer nếu nó tìm thấy nguồn thức ăn tốt hơn, tuy nhiên tỷ lệ giữa producer và scrounger trong toàn bộ quần thể vẫn không thay đổi. Những con chim có mức năng lượng cao hơn sẽ đảm nhận vai trò producer. Một số scrounger đang đói có xu hướng bay đến những khu vực khác để tìm kiếm thức ăn nhằm tăng năng lượng. Các scrounger sẽ đi theo producer nào cung cấp được nguồn thức ăn tốt nhất để cùng kiếm ăn. Đồng thời, một số scrounger có thể liên tục theo dõi các producer và cạnh tranh thức ăn nhằm tăng tỷ lệ săn mồi của mình. Những con chim sẻ ở rìa của đàn sẽ nhanh chóng di chuyển về vùng an toàn để có vị trí tốt hơn khi cảm nhận được mối nguy hiểm, trong khi những con chim ở giữa đàn sẽ di chuyển ngẫu nhiên nhằm duy trì khoảng cách gần với các cá thể khác trong nhóm.

3. Phương pháp tối ưu hóa di chuyển robot

3.1. Tạo bản đồ mê cung bằng GAN

Nghiên cứu này đề xuất phương pháp tạo bản đồ mê cung hai chiều sử dụng mạng neuron GAN với kiến trúc Generator được thiết kế đặc biệt nhằm tạo ra các vị trí vật cản hợp lệ trong không gian lưới. Mỗi bản đồ có kích thước 15×15 đơn vị và bao gồm tổng cộng 60 vật cản hình học là hình vuông và hình chữ nhật với các kích thước khác nhau (0.5×0.5 , 1×1 , 1×0.5 , 0.5×1 , 2×1 và 1×2). Generator có cấu trúc ba tầng, đầu vào là vector nhiễu ngẫu nhiên 100 chiều, tiếp theo là hai lớp ẩn Dense với số neuron lần lượt là 128 và 64 sử dụng hàm kích hoạt ReLU, và đầu ra là lớp Dense 2 chiều đại diện cho tọa độ (x, y) của vật cản. Quá trình sinh tọa độ được giới hạn trong không gian hợp lệ với miền giá trị x

$\in [2, 13.5]$ và $y \in [2, 13.5]$, nhằm tránh các vùng cấm bao gồm vùng góc dưới bên trái ($x \in [0, 3], y \in [0, 3]$) và góc trên bên phải ($x \in [13.5, 14.5], y \in [13.5, 14.5]$). Ngoài ra, hệ thống còn đảm bảo các vật cản được phân bố với khoảng cách tối thiểu 1.0 đơn vị giữa các vật thể nhằm duy trì khả năng di chuyển, và giảm mật độ vật cản tại khu vực hành lang $x \in [12, 13.5], y \in [11, 13.5]$ để đảm bảo độ rộng hành lang $\geq 1.5-2.0$ đơn vị. Bản đồ cuối cùng được mô hình hóa trên lưới 30×30 ô vuông (kích thước mỗi ô là 0.5 đơn vị), trong đó giá trị 1 đại diện cho vật cản (hiển thị màu đen) và giá trị 0 đại diện cho đường đi (hiển thị màu trắng). Phương pháp này cho thấy tính khả thi trong việc sử dụng GAN để tự động sinh môi trường phức tạp với các ràng buộc hình học và không gian rõ ràng, phục vụ cho các ứng dụng trong điều hướng robot và mô phỏng tác vụ. Cấu trúc mạng nơron GAN tạo bản đồ mê cung được mô tả ở Hình 2.



Hình 2. Cấu trúc mạng nơron GAN tạo bản đồ mê cung.

Quá trình huấn luyện GAN sử dụng hàm mất mát Binary Cross Entropy (BCE) để tối ưu hóa cả Generator và Discriminator. Hàm mất mát của GAN được định nghĩa như sau:

Hàm mất mát của Discriminator:

$$L_D = -\mathbb{E}_{x \sim p_{\text{data}}} [\log(D(x))] - \mathbb{E}_{z \sim p_z} [\log(1 - D(G(z)))] \quad (3)$$

trong đó $x \sim p_{\text{data}}$ là tọa độ vật cản thật từ tập dữ liệu, $z \sim p_z$ là vector nhiễu ngẫu nhiên đầu vào của Generator, $D(x)$ là xác suất Discriminator dự đoán rằng (x) là dữ liệu thật, $G(z)$ là tọa độ vật cản giả do Generator tạo ra từ nhiễu (z), \mathbb{E} là kỳ vọng toán học.

Hàm mất mát của Generator:

$$L_G = -\mathbb{E}_{z \sim p_z} [\log(D(G(z)))] \quad (4)$$

Generator cố gắng tối thiểu hóa xác suất mà Discriminator phát hiện dữ liệu giả. Trong quá trình huấn luyện, Generator và Discriminator được cập nhật xen kẽ thông qua thuật toán tối ưu hóa Adam với tốc độ học được thiết lập là 0.0002 và tham số $\beta_1 = 0.5$. Quá trình này đảm bảo rằng Generator tạo ra các bản đồ mê cung với phân bố vật cản ngẫu nhiên nhưng tuân thủ các ràng buộc hình học, như khoảng cách tối thiểu 1.0 đơn vị giữa các vật cản và tránh các vùng cấm gần điểm bắt đầu $S(1,1)$ và điểm mục tiêu $G(14,14)$.

3.2. Tối ưu đường di chuyển ngắn nhất của robot

Thuật toán A* được sử dụng để tìm đường đi ngắn nhất từ điểm bắt đầu $S(1,1)$ đến điểm mục tiêu $G(14,14)$ trên lưới 30×30 . Hàm mục tiêu của thuật toán A* được định nghĩa ở công thức (2) với $g(n)$ là tổng khoảng cách Euclidean của các đoạn đường đi từ điểm bắt đầu $S(1,1)$ đến nút hiện tại $n(x_n, y_n)$. $h(n)$ là hàm Heuristic. Với mỗi bước di chuyển, robot có thể di chuyển theo 8 hướng ngang, dọc, hoặc chéo. Chi phí cho mỗi bước di chuyển được xác định như sau: Di chuyển ngang hoặc dọc là 1.0 tương ứng với $\text{grid_size} = 0.5$ nhân đôi để chuyển sang tọa độ thực. Di chuyển chéo: Chi phí là $\sqrt{2}$ (khoảng

cách Euclidean giữa hai ô lưới chéo nhau. $g(n)$ cho đường đi từ (S) qua các nút $n_1, n_2, \dots, n_k = n$ được tính theo công thức sau:

$$g(n) = \sum_{i=1}^{k-1} d(n_i, n_{i+1}) \quad (5)$$

trong đó $d(n_i, n_{i+1})$ là khoảng cách Euclidean giữa hai nút liên tiếp được tính theo công thức (6):

$$d(n_i, n_{i+1}) = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \quad (6)$$

với (x_i, y_i) và (x_{i+1}, y_{i+1}) là tọa độ của hai nút liên tiếp trên lưới. Hàm heuristic $h(n)$ ước lượng chi phí từ nút hiện tại $n(x_n, y_n)$ đến điểm mục tiêu G (14, 14)

$$h(n) = \sqrt{(x_n - 14)^2 + (y_n - 14)^2} \quad (7)$$

Hàm heuristic có thể chấp nhận, không đánh giá cao chi phí thực tế và đảm bảo tính đơn điệu giúp thuật toán A* tìm được đường đi ngắn nhất một cách hiệu quả. Từ công thức (2), (5), (6) và (7) có thể tính $f(n)$ như sau:

$$f(n) = g(n) + h(n) = \sum_{i=1}^{k-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} + \sqrt{(x_n - 14)^2 + (y_n - 14)^2} \quad (8)$$

Thuật toán A* sẽ ưu tiên mở rộng nút có giá trị $f(n)$ nhỏ nhất trong tập mở để đảm bảo tìm được đường đi tối ưu. Để tạo ra một đường cong mượt mà đi qua các điểm trên đường đi $P = \{p_1, p_2, \dots, p_m\}$, trong đó mỗi điểm $p_i = (x_i, y_i)$ sử dụng nội suy spline khối. Đối với mỗi đoạn giữa hai điểm p_i và p_{i+1} hàm spline khối được xác định như sau:

$$S_i(t) = a_i + b_i t + c_i t^2 + d_i t^3, \quad t \in [0, 1] \quad (9)$$

Trong đó a_i, b_i, c_i, d_i là các hệ số của đa thức bậc ba, được tính dựa trên tọa độ của các điểm p_i, p_{i+1} và điều kiện đảm bảo tính liên tục của đạo hàm bậc nhất và bậc hai tại các điểm nối, t là tham số hóa đoạn đường đi, với $t=0$ tại p_i và $t=1$ tại p_{i+1} . Để đảm bảo tính mượt mà đường spline thỏa mãn điều kiện liên tục bậc 0 (đi qua các điểm p_i). Điều kiện liên tục bậc 1 (đạo hàm bậc nhất liên tục, đảm bảo không có góc gấp khúc). Điều kiện liên tục bậc 2 (đạo hàm bậc hai liên tục, đảm bảo độ cong mượt mà). Trong bài báo hàm smooth_path còn sử dụng cơ chế line-of-sight để loại bỏ các điểm trung gian không cần thiết. Kiểm tra đoạn thẳng giữa hai điểm p_i và p_j (với $j > i + 1$). Nếu đoạn thẳng này không va chạm với vật cản dựa trên lưới 30x30, các điểm trung gian p_{i+1}, \dots, p_{j-1} sẽ bị loại bỏ. Tiêu chí kiểm tra va chạm sử dụng phương trình đường thẳng tham số giữa hai điểm $p_i(x_i, y_i)$ và $p_j(x_j, y_j)$:

$$\begin{cases} x(t) = x_i + t(x_j - x_i) \\ y(t) = y_i + t(y_j - y_i) \end{cases} \quad t \in [0, 1] \quad (10)$$

Bất kỳ điểm nào trên đoạn thẳng $(x(t), y(t))$ nằm trong ô lưới có giá trị 1 (vật cản), đoạn thẳng được coi là không khả thi. Sau khi loại bỏ các điểm trung gian, tọa độ của các điểm còn lại được làm tròn để bám đúng cạnh ô lưới tức là bội số của grid_size = 0.5. Độ dài đường đi tối ưu (P) được tính bằng tổng khoảng cách Euclidean giữa các điểm liên tiếp trên quỹ đạo đã làm mượt:

$$\text{Length}(P) = \sum_{i=1}^{m-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \quad (11)$$

Đầu ra là tọa độ của các vật cản (x, y) được tạo từ quá trình xây dựng mạng nơ-ron GAN. Hàm ánh xạ của bộ sinh như sau:

$$(x, y) = G(z; \theta_G) \quad (12)$$

Trong đó θ_G là tham số của bộ sinh được tối ưu hóa trong quá trình huấn luyện GAN. Để đảm bảo khoảng cách tối thiểu 1.0 đơn vị giữa các vật cản, bài báo sử dụng một ràng buộc:

$$\min_{i \neq j} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \geq 1.0 \quad (13)$$

Với (x_i, y_i) và (x_j, y_j) là tọa độ của hai vật cản bất kỳ. Các vật cản không được nằm trong vùng gần điểm bắt đầu $(S(1, 1))$ ($x: 0-3, y: 0-3$) và điểm mục tiêu $G(14, 14)$ ($x: 13.5-14.5, y: 13.5-14.5$). Điều này được biểu diễn bằng các điều kiện: $(x_i, y_i) \notin [0, 3] \times [0, 3]$ và $(x_i, y_i) \notin [13.5, 14.5] \times [13.5, 14.5]$. Như vậy, để làm mượt đường đi, kỹ thuật nội suy spline khối cải tiến được áp dụng thông qua hàm `smooth_path`. Hàm này sử dụng cơ chế line-of-sight để kiểm tra các đoạn thẳng giữa các nút trên đường đi, loại bỏ các điểm trung gian nếu không va chạm vật cản, và làm tròn tọa độ để bám đúng cạnh ô lưới. Điều này đảm bảo quỹ đạo mượt mà, không bám viền vật cản, và khả thi trong thực tế.

3.3. Quy trình thực hiện thuật toán

Quy trình lập kế hoạch đường đi dựa trên GAN và A* bao gồm các bước sau:

Bước 1: Tối ưu hóa không gian môi trường và tìm đường với A*

Quá trình bắt đầu bằng việc khởi tạo môi trường điều hướng với dữ liệu bản đồ được sinh bởi mạng GAN. Generator của GAN được huấn luyện để sinh ra danh sách tọa độ các vật cản phân bố trong không gian 2D kích thước 15×15 đơn vị. Để đảm bảo tính khả thi của quá trình điều hướng, các tọa độ gần điểm bắt đầu $S=(1,1)$ và điểm đích $G=(14,14)$ cụ thể là các vùng $x \in [0,3], y \in [0,3]$ và $x \in [13.5, 14.5], y \in [13.5, 14.5]$ được loại bỏ khỏi danh sách vật cản. Danh sách vật cản cuối cùng, ký hiệu là *obstacles_extra*, bao gồm 60 vật thể, được bố trí đồng đều trong phạm vi $x, y \in [2, 13.5]$ bảo đảm khoảng cách tối thiểu 1.0 đơn vị giữa các vật cản. Ngoài ra, tại chỉ số thứ 16, một vật cản cố định có tọa độ $[6.3, 7.5, 6.8, 8.8]$ được giữ nguyên nhằm kiểm soát cấu trúc bản đồ. Dữ liệu GAN sau đó được chuyển đổi thành lưới nhị phân 30×30 (với kích thước ô lưới `grid_size=0.5`). Các biên của không gian (tường bao quanh) được thêm vào tại $x=0, x=15, y=0, y=15$. Trong lưới này, giá trị 1 đại diện cho ô chứa vật cản, trong khi giá trị 0 biểu diễn ô trống có thể đi qua. Trước khi thực hiện lập kế hoạch đường đi, thuật toán kiểm tra xem điểm bắt đầu S và điểm kết thúc G có trùng với vị trí vật cản hay không. Nếu có, hệ thống sẽ đưa ra cảnh báo và đề xuất các điểm thay thế thích hợp, ví dụ: $S=(3,3), G=(27,27)$. Tiếp theo, thuật toán tìm đường A* được triển khai với tập mở khởi tạo tại điểm S, sử dụng hàm heuristic Euclidean để ước lượng chi phí đến G. Các nút lân cận trong 8 hướng được duyệt, và nút có tổng chi phí thấp nhất sẽ được chọn để mở rộng đường đi. Quá trình tiếp tục cho đến khi điểm G được đạt đến.

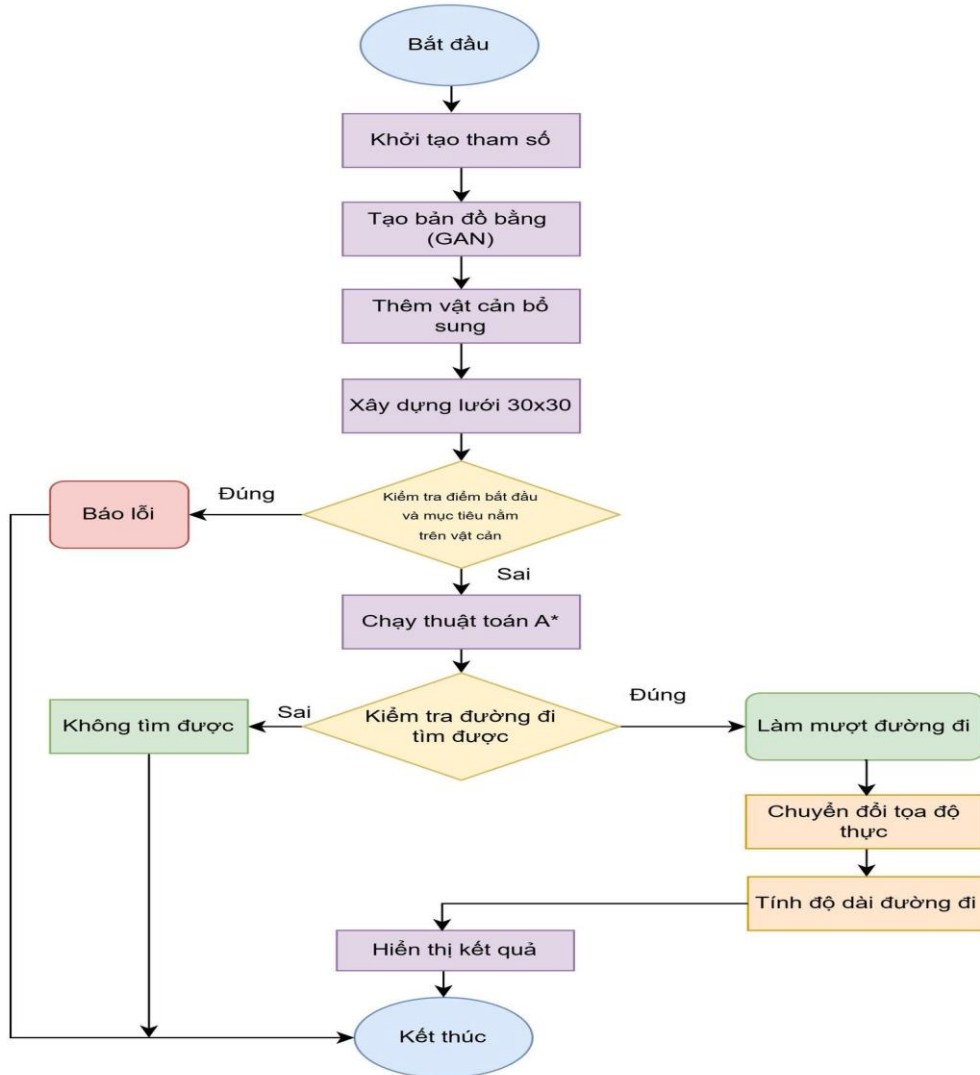
Bước 2: Làm mượt đường đi

Sau khi có được đường đi thô từ thuật toán A*, một hàm làm mượt đường đi (*smooth_path*) được áp dụng nhằm tối ưu hóa quỹ đạo di chuyển. Cụ thể, thuật toán kiểm tra từng đoạn thẳng nối các cặp nút không liên tiếp trong đường đi và loại bỏ các nút trung gian nếu đoạn thẳng đó không giao cắt với vật cản. Bước này giúp giảm độ gấp khúc và nâng cao tính khả thi trong môi trường thực tế. Tọa độ các điểm cũng được làm tròn để căn chỉnh với ranh giới ô lưới.

Bước 3: Tạo đường đi tối ưu

Cuối cùng, đường đi sau khi làm mượt được chuyển đổi từ tọa độ lưới về tọa độ thực trong không gian 15×15 bằng cách nhân với hệ số $grid_size=0.5$. Độ dài tổng thể của đường đi được tính toán dựa trên tổng khoảng cách Euclidean giữa các cặp điểm liên tiếp. Kết quả là một đường đi ngắn, mượt và khả thi cao để triển khai cho robot trong môi trường có chướng ngại vật.

Quá trình mô phỏng được đưa trên lưu đồ thuật toán được thể hiện ở Hình 3 như sau:

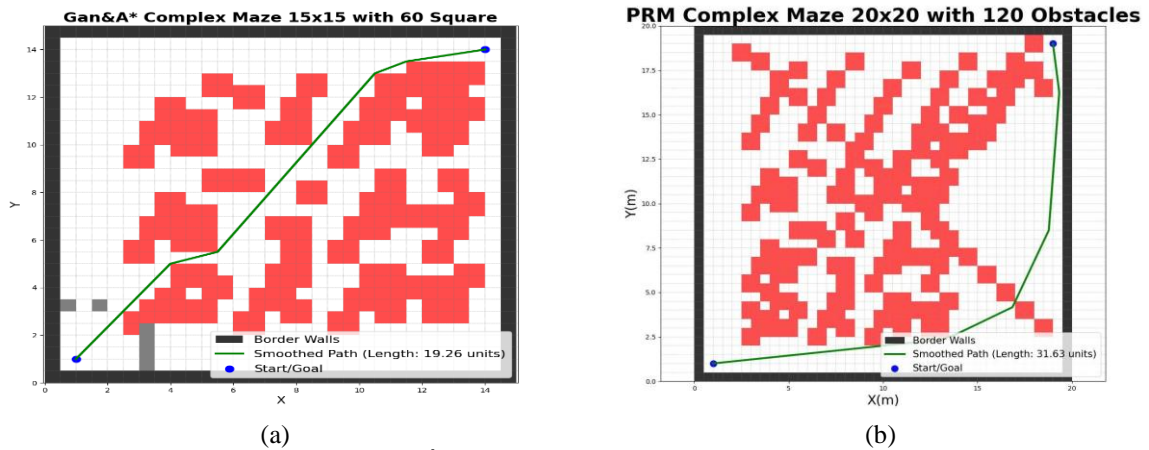


Hình 3. Lưu đồ thuật toán.

4. Kết quả mô phỏng và thảo luận

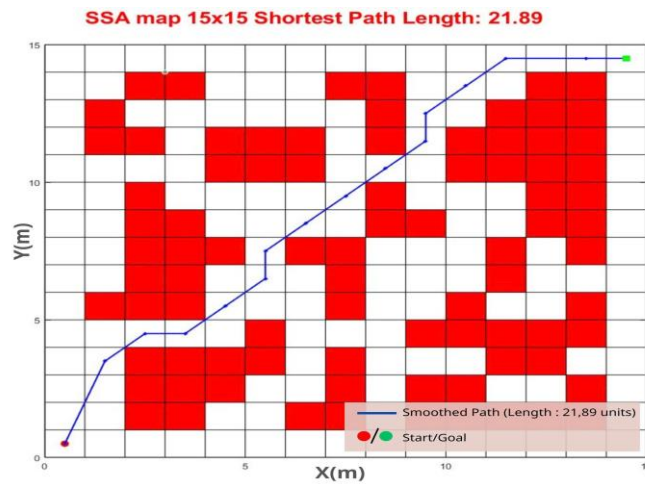
4.1. Kết quả mô phỏng

Mô phỏng này được phát triển bằng ngôn ngữ Python, sử dụng các thư viện chuyên dụng để tạo ra môi trường lập kế hoạch đường đi phức tạp. TensorFlow và NumPy đảm nhiệm việc tạo ra và xử lý hàng ngàn chướng ngại vật ngẫu nhiên thông qua mô hình GAN, bên cạnh đó thư viện Matplotlib phụ trách trực quan hóa toàn bộ không gian mô phỏng 15×15 và 20×20 đơn vị. Thuật toán A* được triển khai với sự hỗ trợ của Heapq cho hàng đợi ưu tiên và thư viện Math để tính toán heuristic, từ đó tìm ra và tối ưu hóa quỹ đạo di chuyển tối ưu cho robot trong môi trường có vật cản. Thực hiện mô phỏng chia ra làm 5 kịch bản. Kịch bản thứ nhất mô phỏng sử dụng thuật toán GAN + A* với không gian mô phỏng 15×15 , 60 vật cản ở Hình 4a, kịch bản thứ 2 mô phỏng sử dụng thuật toán GAN + A* không gian mô phỏng 20×20 đơn vị, 120 vật cản Hình 4b.



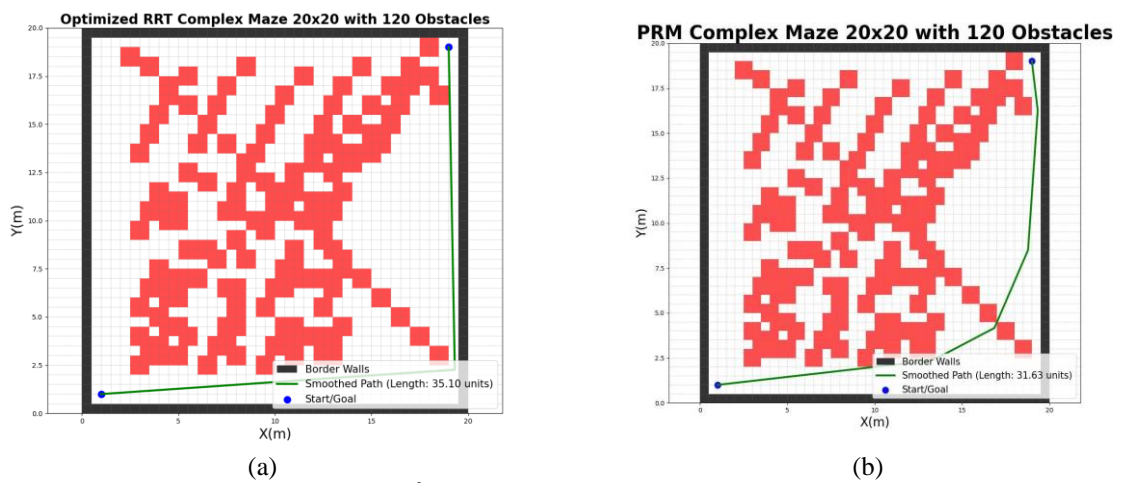
Hình 4. Kết quả mô phỏng thuật toán GAN + A*.

Kịch bản thứ 3 mô phỏng sử dụng thuật toán SSA với không gian mô phỏng là 15x15, 60 vật cản ở Hình 5.



Hình 5. Kết quả mô phỏng thuật toán SSA.

Kịch bản thứ 4 mô phỏng sử dụng thuật toán RRT với không gian mô phỏng là 20x20, 120 vật cản ở Hình 6a và Kịch bản thứ 4 mô phỏng sử dụng thuật toán PRM với không gian mô phỏng là 20x20, 120 vật cản ở Hình 6b.



Hình 6. Kết quả mô phỏng thuật toán RRT, PRM.

4.2. Thảo luận

Kết quả mô phỏng Hình 4a với môi trường mê cung 15x15, lưới 30x30 ($grid_size = 0.5$), 60 vật cản, điểm bắt đầu (1, 1), và điểm mục tiêu (14, 14). Kết quả mô phỏng cho thấy: Độ dài đường đi: Đường đi tối ưu có độ dài 19.26. Độ mượt của quỹ đạo khi dùng hàm `smooth_path` đảm bảo đường đi không bám viền vật cản, với các điểm tọa độ được làm tròn để bám đúng cạnh ô lưới. Hiệu quả tránh chướng ngại vật tốt do bản đồ được tạo bởi GAN và `obstacles_extra` đảm bảo không có vật cản trong vùng cấm, hỗ trợ thuật toán A* tìm đường đi hiệu quả. Từ hai kết quả mô phỏng của Hình 4a và Hình 5 ta có thể tổng hợp đánh giá kết quả của hai kịch bản này do cùng không gian 15x15 đơn vị, 60 vật cản bằng một bảng so sánh giữa thuật toán GAN + A* với thuật toán SSA qua các tiêu chí được thể hiện ở Bảng 1 như sau:

Bảng 1. Kết quả so sánh GAN + A* và thuật toán SSA.

Tiêu chí đánh giá	GAN + A*	SSA (Sparrow Search Algorithm)	Nhận xét khoa học
Độ dài đường đi	19.26 đơn vị	21.89 đơn vị	GAN + A* rút ngắn quãng đường đi ~12%, giúp giảm thời gian và chi phí vận hành robot.
Độ mượt quỹ đạo	Cao sử dụng nội suy spline khối, tránh va chạm vật cản	Thấp quỹ đạo gấp khúc, bám sát vật cản	GAN + A* cho lộ trình mượt, phù hợp di chuyển thực tế, giảm rung chấn khi di chuyển.
Hiệu quả tránh vật cản	Rất tốt GAN sinh bản đồ không có vật cản gần điểm S và G	Trung bình không kiểm soát được vùng cấm	GAN hỗ trợ A* bằng cách loại bỏ các chướng ngại vùng cấm, tăng tỷ lệ thành công.
Cấu trúc bản đồ đầu vào	Sinh bởi GAN, có kiểm soát phân bố vật cản	Sinh ngẫu nhiên hoặc cố định	Bản đồ của GAN tăng tính thực nghiệm và đa dạng kiểm thử cho các mô phỏng.
Khả năng hội tụ giải pháp tối ưu	Cao nhờ heuristic Euclidean và lưới 8 hướng	Trung bình dễ mắc kẹt tại cực trị cục bộ	A* hội tụ nhanh và chính xác trong môi trường sinh bởi GAN.
Khả năng triển khai thực tế	Cao quỹ đạo khả thi, mềm mại, tránh chướng ngại tốt	Hạn chế do thiếu mượt và thiếu linh hoạt	GAN + A* thích hợp cho robot thực địa, xe tự hành hoặc robot logistics.

Phân tích từ dữ liệu định lượng ở Bảng 1 cho thấy phương pháp GAN + A* vượt trội hơn rõ rệt so với SSA ở mọi tiêu chí đánh giá chính. Đặc biệt, độ dài đường đi ngắn hơn, quỹ đạo mượt hơn và hiệu quả tránh vật cản cao hơn giúp nâng cao độ tin cậy và tính ứng dụng trong thực tiễn. Việc sử dụng GAN để sinh bản đồ có kiểm soát không chỉ tăng độ khó bài toán mà còn giúp thuật toán A* hoạt động hiệu quả hơn nhờ loại trừ các yếu tố bất lợi như vùng cấm và vật cản dày đặc gần điểm S và G.

Bên cạnh đó chúng tôi đã thực hiện đa dạng các kịch bản khác nhau như thay đổi kích thước không gian và số vật cản, phát triển thêm với các thuật toán RRT và PRM để làm rõ luận điểm và mục tiêu khi thực hiện bài báo. Trong bối cảnh bài toán lập kế hoạch đường đi trong không gian có số chiều cao, thuật toán RRT (Rapidly-exploring Random Tree) ưu việt hơn các phương pháp dựa trên lưới truyền thống. Kết quả nghiên cứu và mô phỏng cho thấy RRT có khả năng tìm ra lời giải khả thi một cách nhanh chóng nhờ cơ chế lấy mẫu ngẫu nhiên và mở rộng cây tham lam, mà không yêu cầu rời rạc hóa toàn bộ không gian tìm kiếm. Tuy nhiên, đường đi do RRT tạo ra thường thiếu tính tối ưu và trơn tru, điều này củng cố cho nhận định rằng các phiên bản cải tiến như RRT* với cơ chế viết lại để hội tụ tiệm cận về đường đi tối ưu là hướng phát triển cần thiết và hiệu quả cho các ứng dụng đòi hỏi độ chính xác cao [15]-[16]. Thuật toán PRM (Probabilistic Roadmap) được coi là một chiến lược hiệu quả cho các

bài toán lập kế hoạch đa truy vấn trong không gian cấu hình phức tạp. Phương pháp này hoạt động dựa trên việc xây dựng một đồ thị từ các điểm mẫu ngẫu nhiên, tạo ra một mạng lưới kết nối khả thi trong không gian tự do. Ưu điểm nổi bật của PRM là sau khi hoàn tất giai đoạn xây dựng bản đồ tốn kém ban đầu, nó cho phép truy vấn đường đi giữa các cặp điểm bất kỳ một cách cực kỳ nhanh chóng. Tuy nhiên, hiệu suất của PRM phụ thuộc mạnh mẽ vào mật độ lấy mẫu và có thể gặp khó khăn trong việc phát hiện các "điểm thắt cổ chai" hẹp, từ đó hạn chế tính đầy đủ của lời giải trong một số môi trường đặc thù [14].

Bảng 2. Kết quả so sánh GAN + A* và thuật toán RRT, PRM.

Chỉ tiêu	GAN + A*	RRT	PRM
Tối ưu đường đi	Có bảo đảm (Euclid + chi phí $\sqrt{2}$, admissible/consistent)	Không bảo đảm (trừ RRT*)	Không bảo đảm (trừ PRM*)
Độ dài quan sát	28.81u (ngắn nhất)	30.63u	30.51u
Tính quyết định	Quyết định (cùng input \Rightarrow cùng kết quả)	Ngẫu nhiên theo seed/mẫu	Ngẫu nhiên theo mẫu/kết nối
Nhu cầu tinh chỉnh	Thấp (heuristic cố định)	Cao (số mẫu, bước mở rộng, goal-bias)	Cao (n, bán kính/k-NN)
Thời gian (2D lưới vừa)	Rất hiệu quả; chạy trên đồ thị 40x40, làm mượt hậu kỳ	Tăng theo số mẫu	Tăng theo n & cạnh
Tối ưu đường đi	Hạn chế (cần rời rạc hóa)	Mạnh	Mạnh

Từ Bảng 2, đối với bài toán lập kế hoạch đường đi trên lưới 2D, A* vượt trội so với RRT, PRM về tối ưu toàn cục, độ ngắn của đường đi và tính ổn định. RRT, PRM chỉ nên ưu tiên khi không gian liên tục, bậc tự do cao hoặc cần xử lý ràng buộc động học. Trong các trường hợp này, nên dùng các biến thể RRT* hoặc PRM* và tăng chi phí mẫu để tiệm cận chất lượng đường đi của A*.

5. Kết luận và hướng phát triển

Phương pháp sử dụng GAN để tạo bản đồ mê cung, thuật toán A* để tìm đường đi ngắn nhất, và nội suy spline khối để làm mượt quỹ đạo đã chứng minh tính khả thi và hiệu quả trong bài toán lập kế hoạch lộ trình robot. Nhờ khả năng tạo bản đồ phức tạp với phân bố vật cản có kiểm soát, hiệu quả tìm kiếm của A*, và quỹ đạo mượt mà từ nội suy spline khối, phương pháp này đạt được độ dài đường đi tối ưu và khả năng tránh chướng ngại vật hiệu quả. So với thuật toán truyền thống như SSA hay RRT và PRM. Phương pháp đề xuất cho thấy hiệu suất vượt trội về độ dài đường đi, độ mượt của quỹ đạo, và khả năng hội tụ. Nghiên cứu này mở ra triển vọng ứng dụng rộng rãi trong các lĩnh vực như robot thực địa, xe tự hành, và robot logistics. Trong tương lai, nhóm tác giả phát triển tiếp ứng dụng GAN và kết hợp thuật toán A* với thuật toán tối ưu khác tạo phương pháp lai áp dụng đối với trường hợp thiết lập quỹ đạo dịch chuyển của robot trong môi trường 3D, điều kiện ràng buộc thời gian thực và các vật cản động.

Xung đột lợi ích

Các tác giả tuyên bố không có xung đột lợi ích.

TÀI LIỆU THAM KHẢO

- [1] I. Goodfellow, J. Pouget-Abadie, and M. Mirza, "Generative adversarial networks," vol. 63, no. 11, pp. 139–144, Nov. 2020.
- [2] A. Creswell, T. White, and V. Dumoulin, "Generative adversarial networks: An overview," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65, Jan. 2021.
- [3] Y. Chen, Y. Shi, and B. Zhang, "GAN-based environment generation for robot path planning," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2356–2372, Aug. 2022.
- [4] J. Wang, L. Zhang, and H. Liu, "Path planning with GAN-generated complex environments for mobile robots," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 3, pp. 3421–3435, Mar. 2023.
- [5] X. Li, Y. Zhao, and Z. Chen, "Improved GAN for dynamic environment generation in autonomous navigation," *Applied Soft Computing*, vol. 115, p. 108234, Jan. 2023.
- [6] Q. Zhang, T. Liu, and F. Yang, "Hybrid GAN-A* framework for mobile robot path planning in complex mazes," *Robotics and Autonomous Systems*, vol. 149, p. 104123, Jan. 2024.
- [7] T. Nguyen, V. Tran, and H. Pham, "Deep generative models for path planning in high-dimensional environments," *Neurocomputing*, vol. 514, pp. 128–142, Dec. 2024.

- [8] Y. Liu, X. Chen, and Z. Wang, "Advanced GAN for synthetic environment generation in autonomous systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 2, pp. 892–905, Feb. 2025.
- [9] J. Park, H. Kim, and S. Lee, "GAN-enhanced path planning with dynamic obstacle avoidance for mobile robots," vol. 49, no. 1, pp. 123–138, Jan. 2025.
- [10] S. Kim, J. Park, and H. Lee, "GAN-based map generation with controlled obstacle distribution for robot navigation," vol. 12, pp. 45678–45689, 2024.
- [11] D. Tran, H. Nguyen, and T. Vu, "Conditional GAN for generating realistic maze environments in robotic navigation," *Journal of Intelligent & Robotic Systems*, vol. 110, no. 3, pp. 456–470, 2025.
- [12] J. Xue and B. Shen, "A novel swarm intelligence optimization approach: Sparrow search algorithm," *Systems Science & Control Engineering*, vol. 8, no. 1, pp. 22–34, 2020, doi: 10.1080/21642583.2019.1708830.
- [13] V. H. A. Phan, C. T. Nguyen, D. T. Au, T. D. Phan, M. T. Duong, and M. H. Le, "Vision-based perception for autonomous vehicles in obstacle avoidance scenarios," in *Proc. 17th Int. Conf. Human System Interaction (HSI)*, Ulsan, Republic of Korea, 2025, pp. 1–7.
- [14] C. Liu *et al.*, "PRM-D*: Method for mobile robot path planning," *Sensors*, vol. 23, no. 7, p. 3512, 2023, doi: 10.3390/s23073512.
- [15] "CERRT: A mobile robot path planning algorithm based on RRT in complex environments," *Applied Sciences*, vol. 13, no. 17, p. 9666, 2023, doi: 10.3390/app13179666.
- [16] J. Cong *et al.*, "FF-RRT*: A sampling-improved path planning algorithm for mobile robots against concave cavity obstacles," *Complex & Intelligent Systems*, vol. 9, pp. 7249–7267, 2023, doi: 10.1007/s40747-023-01111-6.

Thi-Minh-Tam Le is with Faculty Electrical and Electronic Engineering, Hung Yen University of Technology and Education.
Email: leminhutamutehy@gmail.com. ORCID: <https://orcid.org/0009-0000-5147-5370>. Tel: 0989658725.

The-Thanh Bui is with Faculty of Electromechanics, Hanoi industrial textile garment university. Currently a lecturer at the Faculty of Electromechanics, Hanoi industrial textile garment university, with research fields in automation control and robotics.
Email: thanhtb@hict.edu.vn. ORCID: <https://orcid.org/0009-0005-3581-122X>.

Van-Luong Dang was born in 1996 in Hung Yen, Vietnam. He graduated in Industrial Electronics from Hung Yen University of Technology and Education in 2019. He is currently pursuing a Master's degree at the same university, majoring in program H03241 (2024–2026). His research interests include pathfinding algorithms and distance measurement.
Email: luongkchy116@gmail.com. ORCID: <https://orcid.org/0009-0002-7920-4390>.

Duc-Hung Pham was born in Hung Yen Province, Vietnam, in 1983. He received the B.S. degree in Automatic Control from Hanoi University of Science and Technology, Vietnam, in 2006, the M.S. degree in Automation from Hanoi University of Science and Technology, Vietnam, in 2011, and he received Ph.D. degree in the Department of Electrical Engineering, Yuan Ze University, Chung-Li, Taiwan, in 2022. He is also a Lecturer with Faculty Electrical and Electronic, Hung Yen University of technical and education, Vietnam. His research interests include fuzzy logic control, neural network, cerebellar model articulation controller, brain emotional learning-based intelligent controller, fault tolerant control, secure communication and robot control.
Email: duchung.pham@utehy.edu.vn. ORCID: <https://orcid.org/0000-0003-3344-1593>.

Ngoc-Thang Pham is with Faculty Electrical and Electronic Engineering, Hung Yen University of Technology and Education.
Email: phamngocthangutehy@gmail.com. ORCID: <https://orcid.org/0009-0002-1107-8965>. Tel: 0912287247.