An Improved Hybrid A* for Efficient Path Planning in Simple Environments

Zhuoyang Wang*

Department of Computer Science, Konan University, Kobe, Japan

*Corresponding author: s2471017@s.konan-u.ac.jp

Abstract. Path planning plays an essential role in many fields such as autonomous navigation, robot obstacle avoidance, and route optimization. Despite Hybrid A* being able to handle non-holonomic constraints and generating smoother paths than classical A*, it still suffers from computational efficiency issues, especially in relatively simple environments. By modifying parameters, this paper proposes an Improved Hybrid A* algorithm, which, while maintaining the kinematic feasibility of Hybrid A*, reduces the analytic expansion intervals, interpolation distances, and costs of switching direction. These parameters are tuned towards the reduction of reliance on heuristics and faster computation in relatively simple maze environments. The experimental results in maze environments with varying complexities demonstrate that, compared to the traditional Hybrid A* algorithm, the proposed Improved Hybrid A* algorithm can significantly enhance computing efficiency, particularly in less complex environments. This has shown that the morphology approach improves performance and is thus more suitable for autonomous navigation, where computational efficiency is important.

Keywords: Robot; path-planning; Hybrid A*; Improved Hybrid A*; Maze solving.

1. Introduction

In recent years, path planning has received increasing attention, which is a fundamental domain in many applications such as robotics, autonomous vehicles, UAVs, and video games [1-3]. Efficient algorithms for real-time path planning are crucial for applications such as navigation, collision avoidance, and optimal path research. Among all the heuristic methods, the A* algorithm is one of the most popular algorithms, as it applies to structured, discrete environments and can find the shortest paths [4].

To better handle practical constraints encountered in real-world navigation, researchers developed the Hybrid A* algorithm as an advanced extension of the traditional A*. Unlike traditional A*, the Hybrid A* algorithm considers non-holonomic constraints, making it suitable for vehicles with restricted maneuverability, such as autonomous cars or wheeled robots. By incorporating continuous vehicle kinematics into the search process, Hybrid A* ensures paths are both optimal in length and feasible regarding vehicle dynamics [5, 6].

Despite its advantages, the traditional Hybrid A* algorithm still has notable limitations. In relatively simple environments, Hybrid A* often overly depends on comprehensive heuristic evaluations, leading to unnecessary computational complexity and reduced efficiency. Extensive heuristic computations do not significantly enhance path optimality in these simpler cases, suggesting inefficiencies in resource allocation. Additionally, parameters such as interpolation distances, analytic expansion intervals, and direction-switching costs are typically fixed and not tailored to environmental complexity, further contributing to inefficiencies.

To overcome these issues, this paper introduces an improved Hybrid A* algorithm by optimizing planner parameters. The proposed method significantly reduces unnecessary heuristic evaluations by adjusting interpolation distances, analytic expansion intervals, and direction-switching costs, particularly in simpler maze environments. These adjustments improve computational efficiency at the cost of slightly longer paths. Nevertheless, the algorithm still ensures kinematic feasibility and maintains practical applicability.

The primary objective of this study is to improve computational efficiency and adaptability in Hybrid A* planning by reducing excessive heuristic computations. Additional contributions include

empirical analyses demonstrating substantial efficiency gains in simpler maze environments and validation experiments confirming that the adjusted parameters effectively balance computational efficiency with kinematic feasibility. Comparative experiments provided in this paper illustrate the algorithm's effectiveness, emphasizing its suitability for practical autonomous navigation tasks requiring quick and reliable path planning.

2. Related work

2.1. A* And Hybrid A*

A* is one of the most popular path-planning algorithms because of its efficiency and optimal path-finding. However, its grid-based searching approach can lead to unnatural routes with sharp turns and obstacles closely following the path, making it less suitable for real-world situations.

To address the shortcomings of A* in continuous space, researchers proposed Hybrid A*. The hybrid A* algorithm combines discrete search with continuous path optimization and is better able to account for vehicle kinematic constraints and also improves the smoothness of the path. Up to now, current research mainly focuses on how Hybrid A* can be utilized in autonomous driving and robotic navigation scenarios, proving its effectiveness in difficult environments [7-9].

While several previous works have been proposed to enhance Hybrid A*, these works primarily focus on improving path feasibility and smoothness in complex environments. However, much less of the work has focused on making Hybrid A* computation efficient in simple or structured settings where their configurations could lead to over-computing.

2.2. Improvements Of Hybrid A*

To address the limitations of standard Hybrid A*, several studies have focused on improving its path quality and motion feasibility through various techniques. For instance, in autonomous driving applications, Hybrid A* has been enhanced by integrating Dubins curves [10] and Reeds-Shepp curves [11], enabling smoother and more feasible trajectories that better adhere to vehicle motion constraints. Additionally, several studies have employed trajectory optimization techniques [12] to mitigate oscillations and enhance path continuity.

While these enhancements have significantly improved the quality of generated paths, most existing efforts have primarily targeted trajectory smoothness, dynamic feasibility, or obstacle avoidance in complex environments. In contrast, few studies have explored how planner parameters—such as interpolation distance, motion primitive length, and direction-switching cost—affect computational efficiency, particularly in simpler or more structured settings. This gap is the main motivation for the approach proposed in this paper.

To address this limitation, we present an Improved Hybrid A* algorithm that improves computational efficiency by optimizing planner parameters, without altering the structure of the algorithm.

3. Method

3.1. Hybrid A* path-planning

Hybrid A* builds upon A* by using a continuous path search approach (Figure 1), guiding the search by using the same cost function $f(n) = g(n) + h(n)$. However, in this case, $g(n)$ represents the actual cost of a kinematically feasible path from the start node to the current node, calculated based on Dubins trajectories or Reeds-Shepp trajectories, with the path length determined by the minimum turning radius and steering angle. Meanwhile, $h(n)$ estimates the shortest feasible path to the goal using Dubins or Reeds-Shepp trajectories, providing a more precise heuristic for the search. This algorithm is widely used in real-time path planning, ensuring the generation of smooth and feasible paths.

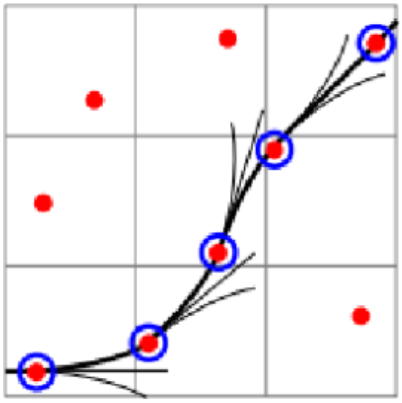


Fig.1 Hybrid A*

The specific calculation is as follows:

$$f(n) = g(n) + h(n) \tag{1}$$

$$g(n) = g(parent) + L \tag{2}$$

$$L = R_{min}\theta \tag{3}$$

Where L is the path length, R_{min} is the minimum turning radius, and θ is the turning angle required to reach the next node.

$$h(n) = 2R_{min}\, sin^{-1}(\frac{d}{2R_{min}}) \tag{4}$$

Where d is the Euclidean distance between n and the goal.

Since Hybrid A* still discretizes the search space, the generated path may not be optimally smooth. To improve trajectory feasibility, it often applies Bézier curves or B-Spline smoothing. A common Bézier curve formulation is:

$$P(t) = (1-t)^2P_0 + 2(1-t)tP_1 + t^2P_2 \qquad t \in [0,1] \tag{5}$$

Where P_0 is the start point, $P_1$is a control point that determines curvature, P_2 is the endpoint.

3.2. Improved Hybrid A* Algorithm

As discussed in previous sections, Hybrid A* can become overly reliant on heuristic evaluations even in relatively simple environments, leading to unnecessary computations and significantly reduced efficiency. To address this issue, the author adjusts key parameters such as interpolation distance and analytic expansion interval that influence computational performance, aiming to greatly improve efficiency while ensuring the kinematic feasibility of the generated path. Below is a step-by-step overview of the proposed algorithm to illustrate the overall planning procedure and parameter configuration.

3.3. Implementation Steps

Algorithm 1 Improved Hybrid A*

Input: Binary maze image, Start and goal poses, Planner parameters: motion Primitive Length, interpolation Distance, analytic Expansion Interval, direction Switching Cost

Output: Planned path (waypoints), Total planning time, Path length

1: Generate a binary maze image using *mazeCreator()*

2: Convert the maze to a binary occupancy map

3: Inflate obstacles to account for robot size

4: Define *state space SE(2)* with bounds and create a state validator

5: Initialize Hybrid A* planner with the following parameters:

MotionPrimitiveLength ←*13*, *InterpolationDistance* ←*0.7*, *AnalyticExpansionInterval* ←*7*, *DirectionSwitchingCost* ←*3*, *MinTurningRadius* ←*10*, *NumMotionPrimitives* ←*3*

6: Define start and goal poses

7: Start timer

8: Execute planner.plan(*startPose, goalPose*)

9: Record elapsed planning time

10: Extract waypoints from the planned path

11: Compute path length from waypoints

12: Visualize the planned path

13: Output planning time and path length

Fig.2 Improve Hybrid A* data

To avoid special cases and enhance the experiment's credibility, multiple mazes are generated by adjusting the branching value while keeping the start and end points fixed, with branching values ranging from 4 to 10. For each value, three different mazes are generated, and the performance of both algorithms in each maze is recorded. The results are average. Due to the large number of mazes, only a selection of the results will be presented in this paper, while the complete data will be summarized in Figure 2.

4. Experiments

4.1. Platform

MATLAB R2024b is a universal data processing platform, used in fields such as data analysis, wireless communication, deep learning, image processing, and computer vision. It is second to none in the numerical calculation of mathematical science and technology applications.

4.2. Evaluation Index

In this research, algorithm performance evaluation will be conducted from two aspects, which are path length and computation efficiency. The formula of path length is as follows:

$$L = \sum_{i=1}^{N-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \quad (6)$$

Where $(x_i,\ y_i)$ represents the ith point on the path, N is the total number of path points.

4.3. Experimental Results

In this experiment, the Hybrid A* and the improved Hybrid A* algorithms are applied to each maze, and the path length planned by each algorithm is calculated using Formula (6). The tic function is used to record the total planning time. Shorter path lengths and lower computation times represent higher computational efficiency.

Figure 2 shows the performance of the Hybrid A* algorithm under different branching values, while Figure 3 shows the Improved Hybrid A*. In these figures, *a, b, c,* and *d* correspond to branching values of 4, 6, 8, and 10, respectively. Figure 4 illustrates the relationship between the branching value and path length, whereas Figure 5 depicts the relationship between the branching value and planning time.

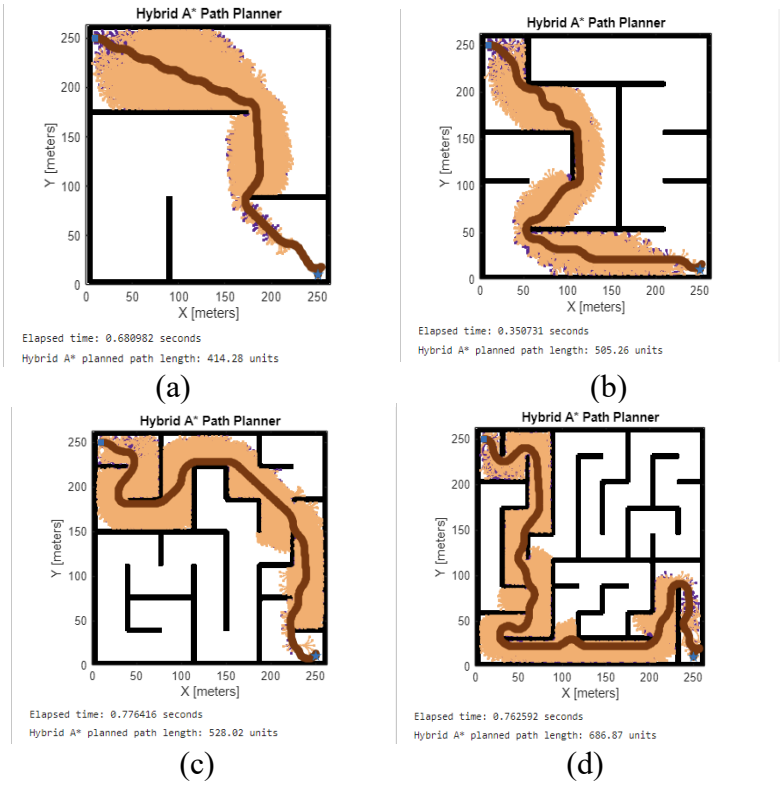
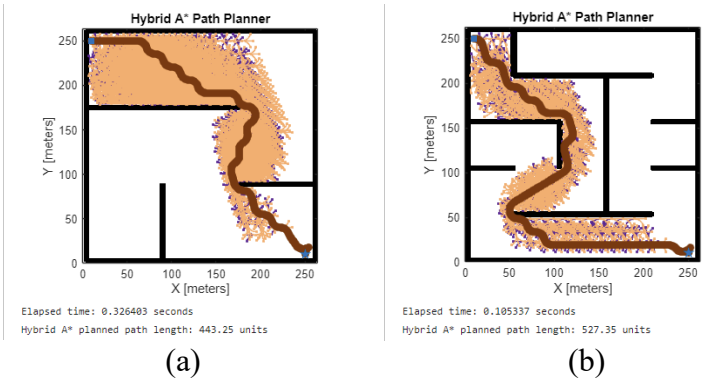
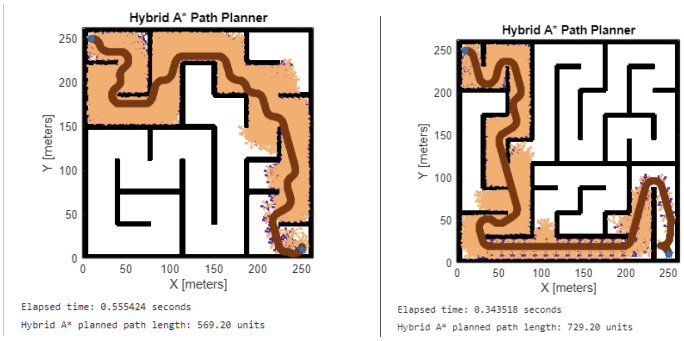


Fig. 2 Performance Of Hybrid A*





(c) (d)

Fig.3 Performance of Improved Hybrid A*

From the visual comparison of both algorithms in the maze environments, it is evident that the original Hybrid A* and the Improved Hybrid A* algorithms are both capable of generating kinematically feasible paths. In terms of path smoothness, the difference between the two approaches is minimal.

Regarding the exploration area and node expansion, the two algorithms cover similar regions during planning. However, in some cases, the Improved Hybrid A* demonstrates a notably lower density of node expansion, indicating a more efficient search process.

The results presented in the two-line charts further support these observations. As shown in Figures 4 and 5. In terms of path length, the two algorithms perform similarly, with the Improved Hybrid A* producing slightly longer paths. However, the difference is marginal and does not compromise the overall path quality.

In contrast, a significant difference is observed in planning time. In most scenarios, the Improved Hybrid A* achieves nearly half—or even greater—reduction in computation time compared to the traditional Hybrid A*. Although there are isolated cases where the performance is comparable or the improved version takes slightly longer, the overall trend consistently shows a clear advantage in efficiency for the improved algorithm.

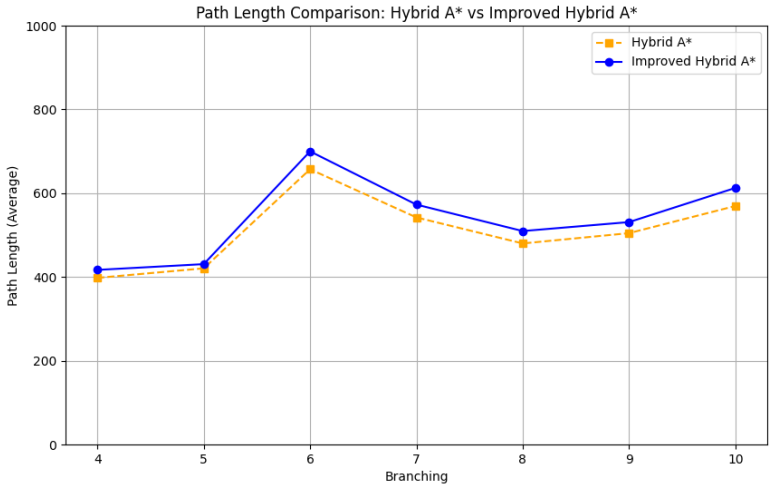


Fig.4 Line Chart of Path Length and Branching

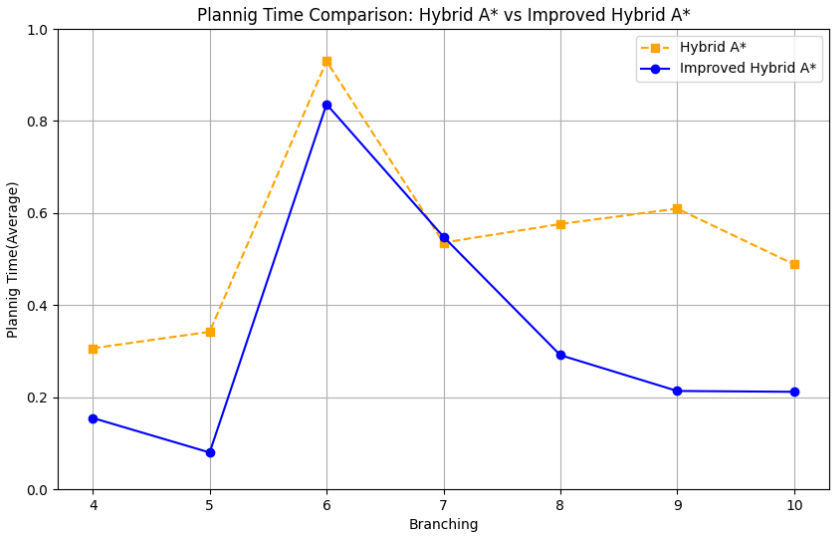


Fig.5 Line Chart of Planning Time and Branching

5. Discussion

The experimental results demonstrate that the Improved Hybrid A* algorithm achieves comparable path quality to the traditional Hybrid A* while significantly reducing computation time in most scenarios. The paths generated by both algorithms remain kinematically feasible, and differences in path length are minimal, suggesting that the improved algorithm does not affect overall path optimality.

The most notable advantage of the Improved Hybrid A* is its enhanced computational efficiency. By adjusting key planner parameters—such as interpolation distance and analytic expansion interval—the algorithm is able to reduce unnecessary node expansion, especially in relatively simple or structured environments. This leads to faster planning without altering the core search mechanism.

Moreover, the visualization of node expansion density supports the claim that the improved algorithm conducts a more targeted and efficient search, reducing redundancy. While isolated cases show comparable or slightly longer planning times, these are exceptions rather than the norm and may be attributed to specific maze configurations or local minima in heuristic evaluation.

These research findings confirm that moderate parameter adjustments can lead to meaningful performance improvements, particularly when the planning task does not require high-resolution path evaluation. The proposed method thus provides a practical and lightweight improvement to Hybrid A* for applications where computational efficiency is critical.

However, although the improved Hybrid A* algorithm has significantly improved computational efficiency, its performance may decline in more complex or dynamic environments. This is because such environments often require extensive computation, and the improved algorithm, which relies on reducing heuristic dependence, may fail to generate an optimal path or result in a substantial increase in path length. Additionally, using fixed parameter values in this improved algorithm may further contribute to reduced performance in complex scenarios.

Future research could explore adaptive parameter tuning based on real-time environmental analysis, allowing the planner to adjust its behavior dynamically according to local complexity. Another viable direction is to integrate the proposed parameter optimization into more advanced planning frameworks, such as those combining Hybrid A* with sampling-based or learning-based methods. These approaches may further enhance performance in diverse and unpredictable environments while preserving the lightweight nature of the current solution.

6. Conclusion

In this work, the author proposed an Improved Hybrid A* algorithm by optimizing key planner parameters, including interpolation distance, analytic expansion interval, and direction-switching cost. These modifications aim to enhance computational efficiency while maintaining the kinematic feasibility of the generated paths. Through a series of experiments in maze environments of varying complexity, we compared the performance of the original and improved algorithms in terms of path length, node expansion, and planning time. The experimental results demonstrate that the improved Hybrid A* achieves similar path quality compared to the original version, while significantly reducing computation time in most cases. This confirms that even lightweight parameter-level adjustments can lead to meaningful performance improvements, particularly in structured or low-complexity environments where excessive heuristic evaluation is unnecessary. This study highlights the importance of planner parameter tuning in practical path planning and offers a simple yet effective way to improve Hybrid A* performance without altering its core framework. Future work may explore adaptive parameter strategies or combine this approach with more advanced or learning-based planners to further improve robustness and efficiency in dynamic and complex environments.

References

[1] Van den Berg, Jur, et al. "Anytime Nonparametric A*." Proceedings of the AAAI Conference on Artificial Intelligence, 4 Aug. 2011, vol. 25, no. 1, pp. 105–111.

[2] Sturtevant, Nathan, and Robert Geisberger. "A Comparison of High-Level Approaches for Speeding up Pathfinding." Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, 10 Oct. 2010, vol. 6, no. 1, pp. 76–82, .

[3] Kolivand, Hoshang, and Mohd Shahrizal Sunar. "Survey of Shadow Volume Algorithms in Computer Graphics." IETE Technical Review, 2013, vol. 30, no. 1, p. 38.

[4] Foead, Daniel, et al. "A Systematic Literature Review of A* Pathfinding." Procedia Computer Science, 2021, vol. 179, pp. 507–514.

[5] H. Sang, Y. You, X. Sun, Y. Zhou, and F. Liu, "The hybrid path planning algorithm based on improved A* and artificial potential field for unmanned surface vehicle formations," Ocean Engineering, Mar. 2021, vol. 223, p. 108709.

[6] B. B. K. Ayawli, R. Chellali, A. Y. Appiah, and F. Kyeremeh, "An Overview of Nature-Inspired, Conventional, and Hybrid Methods of Autonomous Vehicle Path Planning," Journal of Advanced Transportation, Jul. 2018, vol. 2018, pp. 1–27.

[7] N. Wang and H. Xu, "Dynamics-Constrained Global-Local Hybrid Path Planning of an Autonomous Surface Vehicle," IEEE Transactions on Vehicular Technology, Apr. 2020, vol. 69, no. 7, pp. 6928–6942.

[8] U. Orozco-Rosas, K. Picos, and O. Montiel, "Hybrid Path Planning Algorithm Based on Membrane Pseudo-Bacterial Potential Field for Autonomous Mobile Robots," IEEE Access, 2019, vol. 7, pp. 156787–156803.

[9] J. Li, G. Deng, C. Luo, Q. Lin, Q. Yan, and Z. Ming, "A Hybrid Path Planning Method in Unmanned Air/Ground Vehicle (UAV/UGV) Cooperative Systems," IEEE Transactions on Vehicular Technology, Dec. 2016, vol. 65, no. 12, pp. 9585–9596.

[10] Y. Lin and Srikanth Saripalli, "Path planning using 3D Dubins Curve for Unmanned Aerial Vehicles," May 201.

[11] T. Fraichard and A. Scheuer, "From Reeds and Shepp's to Continuous-Curvature Paths," IEEE Transactions on Robotics, Dec. 2004, vol. 20, no. 6, pp. 1025–1035.

[12] J.-J. Kim and J.-J. Lee, “Trajectory Optimization With Particle Swarm Optimization for Manipulator Motion Planning,” IEEE Transactions on Industrial Informatics, Jun. 2015, vol. 11, no. 3, pp. 620–631.