EECS465 final project: Search-based Planning

by

Yilin Jia

1 Introduction

Autonomous navigation stands as a critical component in the development of intelligent systems ranging from domestic robots to autonomous vehicles. These systems require reliable methods to navigate efficiently in complex environments, making the optimization of search-based planning algorithms an essential area of research. The problem addressed in this project centers on enhancing the performance and efficiency of search algorithms within the domain of robotics, where the ability to quickly find an optimal path in a dynamic and often unpredictable environment is paramount.

The PR2 robot, which serves as the platform for our study, is emblematic of the challenges faced in robotics navigation. It operates in a world that demands real-time decision-making and adaptation to changing conditions. Consequently, the implementation and improvement of search algorithms like A* and its variants, such as ANA*, are not just academic exercises but necessities for advancing the field of robotics. These algorithms are the cornerstone of pathfinding and motion planning applications, including but not limited to, service robots in healthcare, autonomous delivery systems, and search-and-rescue operations where time and efficiency can mean the difference between success and failure.

In this project, we build upon the A* algorithm implemented in Homework 3, adapting it to the ANA* algorithm for an 8-connected space, which allows for more natural and realistic movements by the PR2. The motivation behind using the ANA* algorithm is its ability to improve search efficiency by dynamically adjusting its heuristic based on the current state of the search, potentially offering faster convergence to the optimal path compared to A*. We aim to investigate this hypothesis by comparing the performance of ANA* with our implementation of A* across a variety of navigation problems that are both interesting and challenging.

2 Implementation

2.1 Heuristic

Euclidean heuristic:

$$h_s = \sqrt{(n_x - g_x)^2 + (n_y - g_y)^2 + \min|n_\theta - g_\theta|^2, (2\pi - |n_\theta - g_\theta|)^2}$$

My heuristic:

$$h_s = \sqrt{(n_x - g_x)^2 + 0.6(n_y - g_y)^2 + 4\sin{(n_\theta - g_\theta)}^2}$$

The reason I use sin is because it can make the theta part more smooth. Since there is a wall in the setting, it will make the robot stuck on the y direction. So, I make the factor of y direction smaller to mitigate its impact.

2.2 algorithm

In this experiment, we will use two algorithm A star and ANA star.

A* Algorithm

The A* (pronounced ”A-star”) algorithm is a widely-used pathfinding and graph traversal algorithm in computer science and artificial intelligence. It is an informed search algorithm that finds the shortest path from a start node to a goal node, considering various obstacles and costs. The algorithm's efficiency comes from its use of a heuristic to estimate the cost of reaching the goal from each node. The total cost function for a node n in A* is represented as:

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

where:

- $f(n)$ is the total estimated cost of the cheapest solution through node n.
- $g(n)$ is the actual cost from the start node to node n.
- $h(n)$ is the heuristic estimate of the cost from node n to the goal.

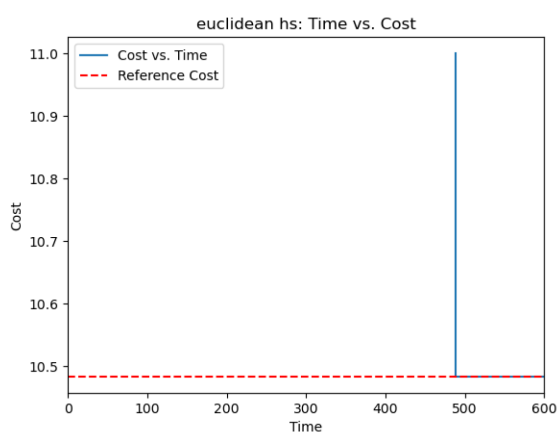
If the heuristic $h(n)$ is admissible, meaning that it never overestimates the true cost, A* is guaranteed to find an optimal solution.

ANA* Algorithm

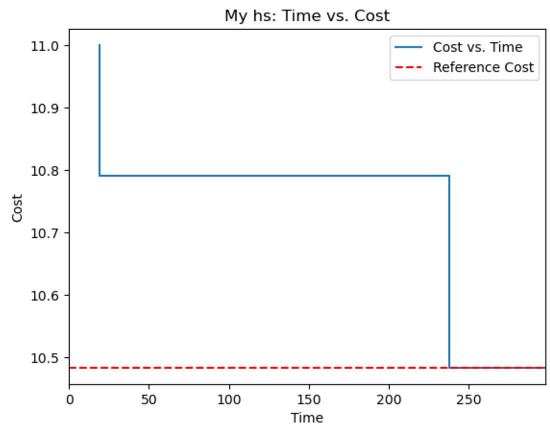
ANA* (Anytime Non-parametric A*) is a variant of A* designed for situations with uncertain planning time. As an "anytime" algorithm, ANA* can provide a valid solution even when interrupted before completion, with the solution quality improving over time. ANA* works by initially finding a suboptimal solution and iteratively refining it. The algorithm dynamically adjusts its heuristic to focus on parts of the search space that are most likely to lead to better solutions. This property is particularly beneficial in real-time systems where quick responses are essential.

2.3 Performance

With Euclidean heuristic, A star takes 637 seconds to get the optimal solution. ANA star takes 488 seconds to get the optimal solution.



With my heuristic, A star takes 492 seconds to get the optimal solution. ANA star takes 238 seconds to get the optimal solution.



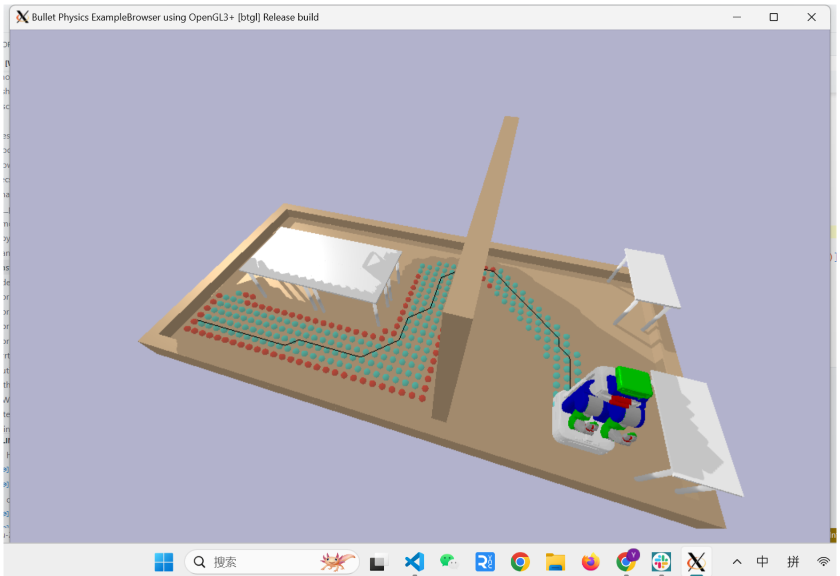


Figure 1: path

3 Conclusions

According to figure 2.3 and figure 2.3, we can find that the ANA star can achieve the optimal solution faster than A star. Besides, my heuristic achieves the optimal solution faster than euclidean heuristic.