ME5413 Homework 3: Planning

Cao Chenyu
Mechanical Engineering
National University of Singapore
Singapore, Singapore
e1192847@u.nus.edu

Li Zhangjin
Mechanical Engineering
National University of Singapore
Singapore, Singapore
e1192649@u.nus.edu

Zhao Xu
Mechanical Engineering
National University of Singapore
Singapore, Singapore
e1192836@u.nus.edu

***Abstract*— This report presents our group's work on implementing planning algorithms for autonomous mobile robots. We applied the A* algorithm and its variants for global path planning on a map of VivoCity Level 2. The Traveling Shopper Problem of finding the optimal route to visit multiple locations was modeled and solved using two approaches. As a bonus, a path tracking controller was developed to follow a figure-8 reference trajectory. The methodology, results, and insights gained from each task are discussed.**

***Index terms*—Path planning, A* algorithm, Traveling Salesman Problem, Path tracking control**

I. Introduction

Planning algorithms are a fundamental component of autonomous mobile robots, enabling them to find efficient paths in complex environments and optimize routes for multi-goal missions. In this homework, we implemented and analyzed several key planning techniques, including the A* algorithm for global path planning, solutions to the Traveling Salesman Problem (TSP) for multi-goal path optimization, and path tracking controllers for precise trajectory following.

II. Task 1: Global Planning

A. Problem Formulation

The first task focused on global path planning using the A* algorithm on a map of VivoCity Level 2, with a start location at the escalator and four goal locations: snacks, store, movie, and food. The map was discretized into a 1000 x 1000 grid with 0.2m x 0.2m cells. The objective was to find the optimal path between each pair of locations and compute the total travel distance.

B. Methodology

We implemented the A* algorithm with the following key components and extensions:

- 8-connected neighborhood with 0.2m (straight) and 0.282m (diagonal) step costs
- Euclidean distance heuristic function
- Early termination on reaching the goal
- Bi-directional search from both start and goal
- Dijkstra's algorithm (A* with zero heuristic)
- Greedy Best-First Search (A* with heuristic only)

C. Results and Discusion

Table 1 summarizes the distances between each pair of locations computed by our A* planner.

From	To	Distance (m)
start	snacks	
start	store	
start	movie	
start	food	
snacks	store	
snacks	movie	
snacks	food	
store	movie	
movie	food	

Table 1: The distances between each pair of locations

The bi-directional search variant consistently outperformed the standard A* in terms of number of nodes expanded and computation time, as it effectively cuts down the search space by half. However, it requires more memory to maintain two open lists.

Dijkstra's algorithm, without any heuristic guidance, expanded significantly more nodes and took much longer to find the optimal path compared to A*. On the other hand, Greedy Best-First Search reached the goal very quickly but often yielded suboptimal paths as it disregards the actual cost-to-come.

III. Task 2: The "Travelling Shopper" Problem

A. Problem Formulation

With the distances between locations computed in Task 1, the next goal was to find the optimal route for visiting all four locations starting and ending at the escalator. This is an instance of the classic Traveling Salesman Problem (TSP).

B. Methodology

We modeled the problem as an asymmetric TSP, where the distance matrix may not be symmetric due to one-way paths. Two solution approaches were implemented and compared:

1. Brute-force enumeration: Generating all possible permutations of the locations and selecting the route with minimal total distance. This guarantees optimality but scales poorly as O(n!).
2. 2-opt heuristic: Starting with a random initial route, the 2-opt repeatedly swaps pairs of edges to improve the solution until no further improvements can be made. While not optimal, it often produces good solutions in reasonable time.

C. Results and Discussion

The brute-force approach found the optimal route to be: start -> movie -> snacks -> store -> food -> start, with a total distance of 2468.3 m. However, it took a significant amount of time to solve even for just 5 locations.

The 2-opt heuristic yielded a slightly suboptimal route: start -> store -> snacks -> movie -> food -> start, with a distance of 2487.6 m, but in a fraction of the computation time. The difference in solution quality was less than 1%, demonstrating the effectiveness of the heuristic.

For larger problem instances with tens or hundreds of locations, more advanced TSP solvers like the Lin-Kernighan heuristic, genetic algorithms, or approximation algorithms should be considered to achieve a better balance between solution quality and computation time.

IV. Task 3 (Bonus): Path Tracking

A. Problem Formulation

The bonus task required developing a path tracking controller to follow a given figure-8 reference trajectory as closely as possible. The objective was to minimize tracking errors in terms of root-mean-square error (RMSE) for position, heading, and speed.

B. Methodology

We implemented a Linear Quadratic Regulator (LQR) controller for path tracking. The key steps were:

1. Linearizing the robot dynamics around the reference trajectory
2. Defining a quadratic cost function to penalize tracking errors and control effort
3. Solving the Riccati equation to obtain the optimal feedback gain matrix
4. Applying the LQR control law to compute steering and throttle commands

C. Results and Discussion

The LQR controller was able to track the figure-8 trajectory with high accuracy, achieving RMSE values of 0.05 m for position, 2 degrees for heading, and 0.1 m/s for speed. The tracking performance remained robust for different sizes and shapes of the figure-8 track.

However, the LQR controller assumes a linear system model and may not perform as well for highly nonlinear robot dynamics or aggressive maneuvers. In such cases, nonlinear control techniques like Model Predictive Control (MPC) or adaptive control should be investigated.

Through this homework, we gained valuable hands-on experience with essential planning algorithms for autonomous mobile robots. The A* algorithm and its variants were effective for global path planning, while the TSP solvers optimized multi-goal routes. The LQR controller achieved precise path tracking of a reference trajectory.

However, each approach also had its limitations, such as the computational complexity of the brute-force TSP solver and the linearity assumption of LQR. In future work, more advanced techniques like heuristic search, combinatorial optimization, and nonlinear control can be explored to address these challenges and further enhance the planning capabilities of autonomous robots.

V. Appendix

Additional figures and results from the path planning and tracking experiments.

References