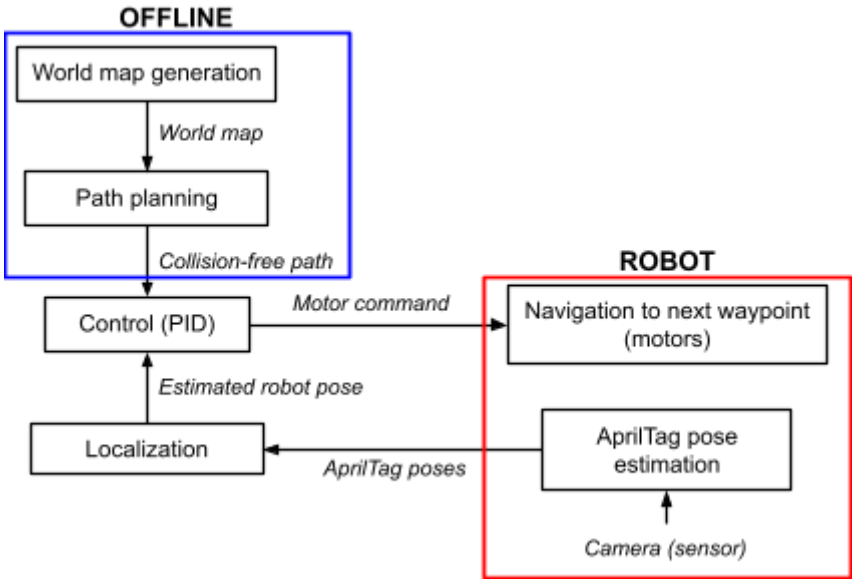
HW 4 Report

**Architecture**




**Representation**

We used a tree-based approach based on our map of the locations of landmarks and the obstacle, represented using (x, y) coordinates. We believed this was a good approach given the complexity of the C-space for the robot (possible configurations must consider landmarks, walls, obstacles). We set our map to size 2.4 m x 2.4 m. The size of the obstacle is 0.36 m x 0.36 m. So we set the boundary region of our box in the code to be 0.40 m x 0.40 m. The obstacle corners were set to (1, 1), (1, 1.4), (1.4, 1), and (1.4, 1.4) in the world map. The wall corners were set to (0.1, 0.1), (0.1, 2.3), (2.3, 0.1), and (2.3, 2.3). Landmarks were placed as shown in the below visualization.

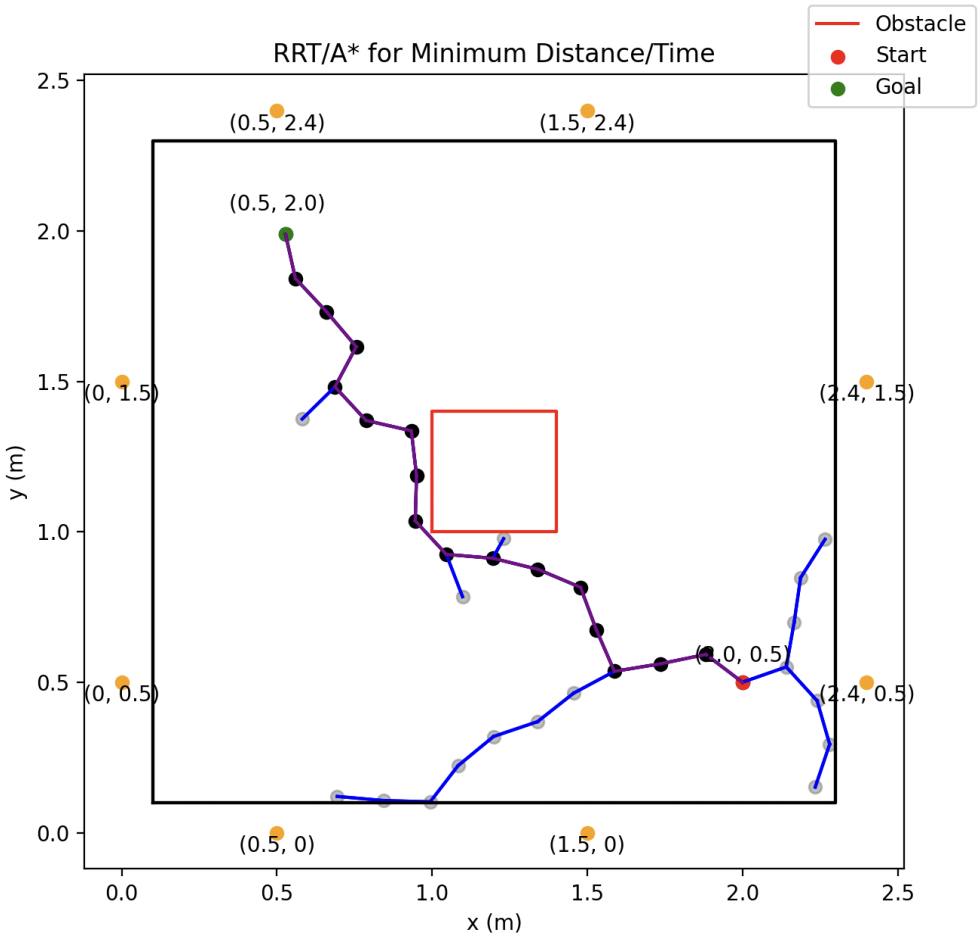
**Algorithm**

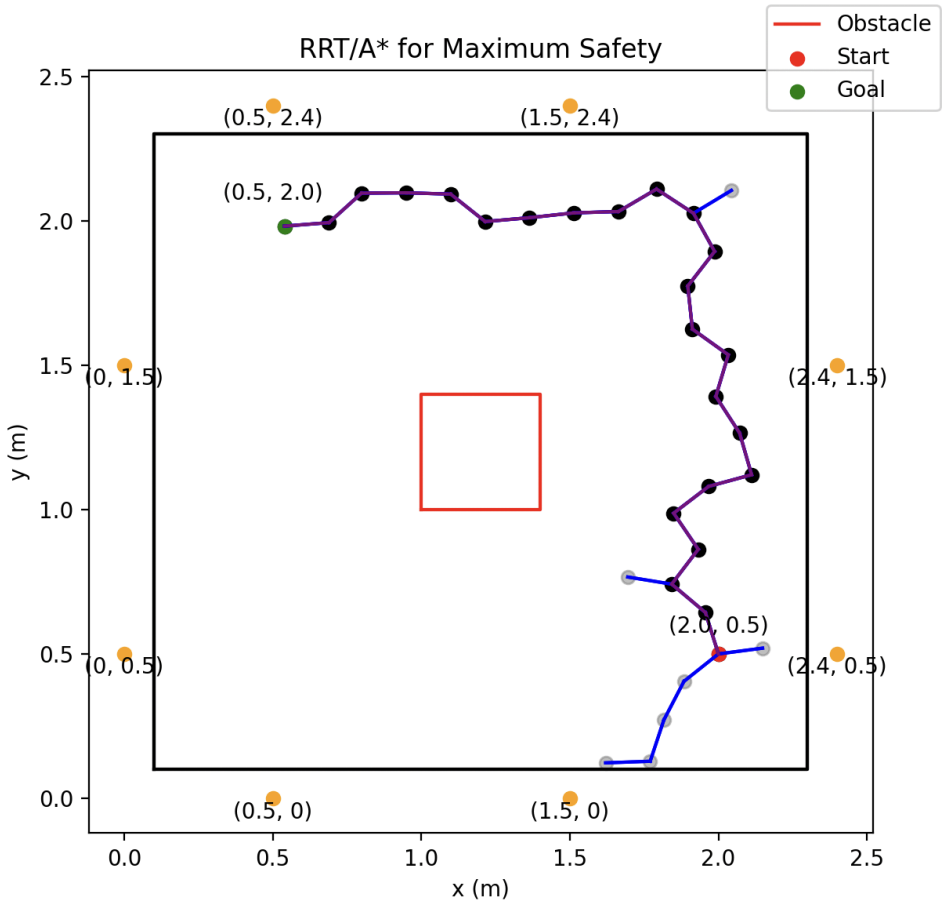
We used a combination of the RRT and A* planners to get better results. We used the properties of the probabilistic RRT planner to quickly explore the large environment and build a graph. We randomly generated robot configurations (x, y) as vertices and edges between them that do not intersect any obstacles or walls provided in our representation. Then we ran A* to select the best path for minimum distance and maximum safety. Then we calculated the required θ to get from one point in the path to another.

We distinguished the two algorithms for minimum distance and maximum safety by modifying the cost function for A* and point generation for the RRT. In the original code, the cost function is based on the Euclidean distance between two points. G cost depends on the cost to reach the current node from the start node and h cost (heuristic function) depends on the cost to reach the goal node from the current node. Therefore, the default A* already attempts to find

the shortest path, making it fulfill the minimum distance requirement. By modifying the cost function to take into account the distance from obstacles (the center obstacle and the wall, specifically), we can fulfill the maximum safety requirement. We can also generate points using RRT that are closer to the middle region between the obstacles and walls (reducing the size of the area that is considered collision-free) to fulfill the requirement. This way, we have two different algorithms, based on the same two planners with modifications, that achieve these different goals.

In the visualization below, the orange dots are landmarks and the wall region is the black box, so the landmarks are considered to be past the walls and therefore they won't be in the collision area. The grey dots with blue edges represent the RRT results, while the black dots with purple edges represent the A* results.






**Results**

Our algorithms successfully created the minimum distance and maximum safety paths. In terms of the robot performance, it successfully managed to achieve the minimum distance and maximum safety requirements without collisions and sticking pretty close to the generated paths.

Video for Minimum Distance/Time: https://youtu.be/zshtPr-Oiwk

Video for Maximum Safety: https://youtu.be/1Xi58OufvjY

*For this assignment, we launched hw2_solution.py and did path planning offline. The modified files are in rb5_control.py.