RBE 550 – RRT & RRT* Algorithm Implementation

By Chinmayee Prabhakar

Q. ***For RRT, what is the main difference between RRT and RRT*? What change does it make in terms of the efficiency of the algorithms and optimality of the search result?***

Ans: RRT and RRT* are both sampling based algorithms used for motion planning in robotics and automation. The main difference between them is that RRT* has an additional step to optimize the generated tree structure. After adding a new node, RRT* searches for the lowest cost path from the root to the new node, and if a better path is found, the tree is rewired accordingly. This optimization leads to a more efficient search process and more optimal paths.

RRT* is generally more efficient than RRT, with better convergence properties and a higher likelihood of finding a feasible path to the goal. However, the optimization step in RRT* can be computationally expensive and may not always result in the globally optimal path.

Q. ***Compare RRT with the results obtained with PRM in the previous assignment. What are the advantages and disadvantages?***

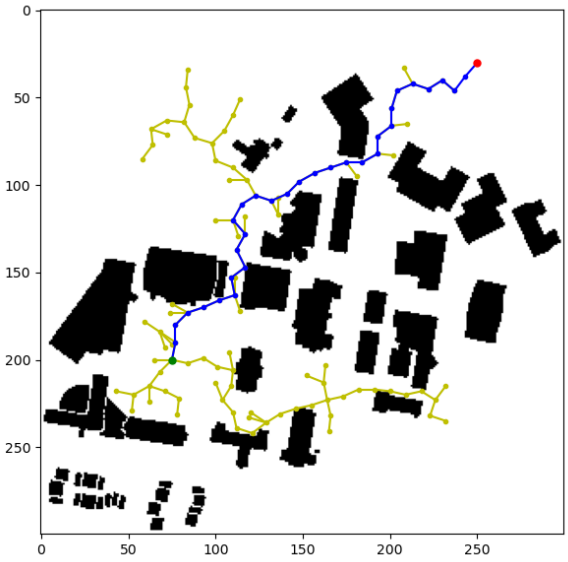
Ans: RRT and PRM differ in their approach to exploring the Configuration - space, connection of nodes, sampling criteria, and completeness. RRT is more focused on finding nodes that can be connected to the tree, while PRM samples the entire C-space and may not always connect all nodes to the start. RRT is probabilistically complete, while PRM may fail to find a path in certain cases.

Algorithm results and explanation

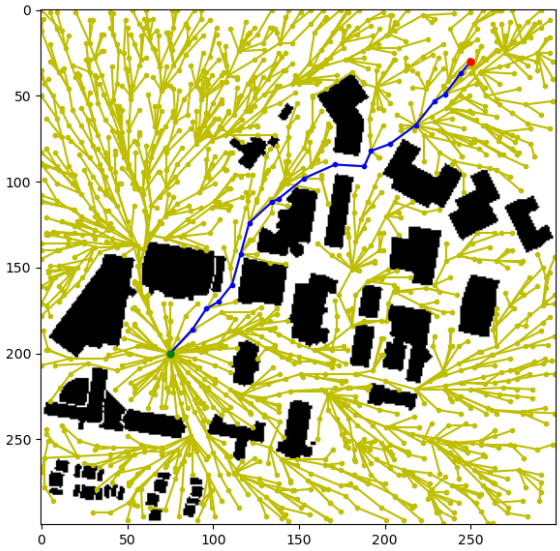
The beginning stages of both algorithms are similar, as they involve identifying a new node in the configuration space that is within a certain distance of the tree, and adding it to the nearest tree node. However, RRT* differs in that it aims to optimize the path from the start node to the new node by rewiring it to the neighboring node with the lowest cost. Furthermore, if the new node provides an alternate low-cost path to neighboring nodes, RRT* will rewire all of them. Although RRT* requires more samples and is computationally more intensive, it provides a more optimal path (i.e., a shorter path from start to goal). In contrast, RRT stops building its tree once it reaches the goal, whereas RRT* continues to optimize its path indefinitely.

Our algorithm consists of separate functions for each of the sub-planning tasks (e.g., calculating the distance between two nodes, checking for collisions, and rewiring), which are combined under

the RRT and RRT* functions to implement the respective algorithms.



RRT



RRT*

```
C:\Users\Admin\anaconda3\python.exe "C:/Users/Admi
It took 107 nodes to find the current path
The path length is 307.81
It took 1661 nodes to find the current path
The path length is 286.16

Process finished with exit code 0
```

RRT

RRT*