Emanuele Costantino

ASEN 5519

Homework #5

Exercise 1

a) A **complete** planning algorithm is one that finds a path to goal when one exists and reports if a path doesn't exist in finite time.

b) Optimal may mean many things depending on what you are trying to optimize. If we are trying to minimize travel distance, an **optimal** planning algorithm is one that takes the shortest path between start and goal.

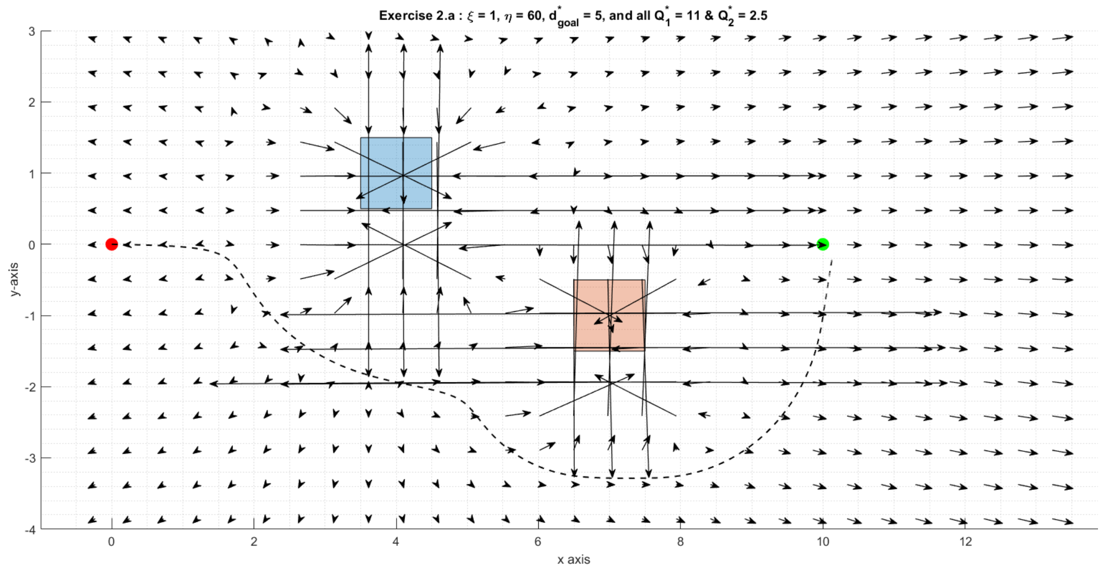
c) The wavefront planner is not complete in general. It is complete with respect to the grid resolution you choose. There may be scenarios where the chosen grid resolution is not fine enough and a path to goal is not found (narrow passages). If a path is found, the path will be the shortest path on the grid, thus optimal on the grid.

Exercise 2

a) Workspace #1

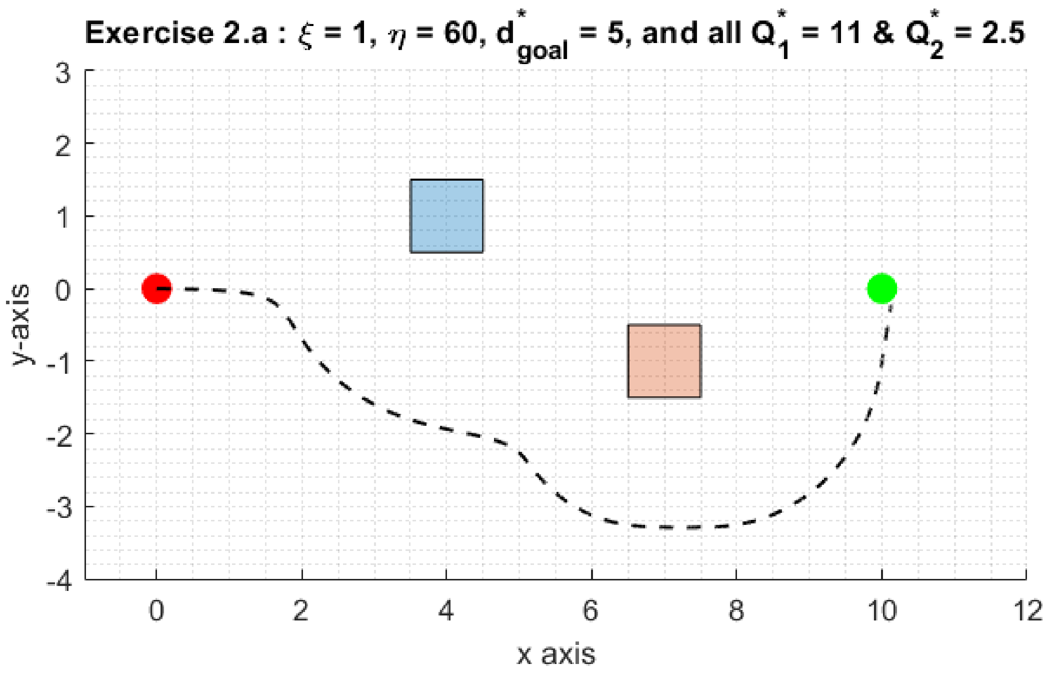
i. Vector Field

The vectors are spaced by 0.5 and are not present inside of the obstacles. Behind the vector field I have also plotted the goal, start, obstacles, and the path taken by the robot.



ii. I chose d* by making it equal to half of the distance between the goal and start. This was done to be able to maneuver the robot completely around the obstacles by following the approximate shape of the obstacles. The Q* for obstacle 1, in blue, is 11 and was chosen to push the path of the goal as far down as possible while still completing a path to the goal. These values were chosen iteratively through trial and error. The Q* value for obstacle 2, in red, is 2.5 and was chosen so as to "swing" the path around the obstacle and allows the now stronger attractive force near the goal to guide the robot towards goal. Ultimately, I wanted to make a path that was interesting and not optimal using gradient decent.

iii. Path Generated



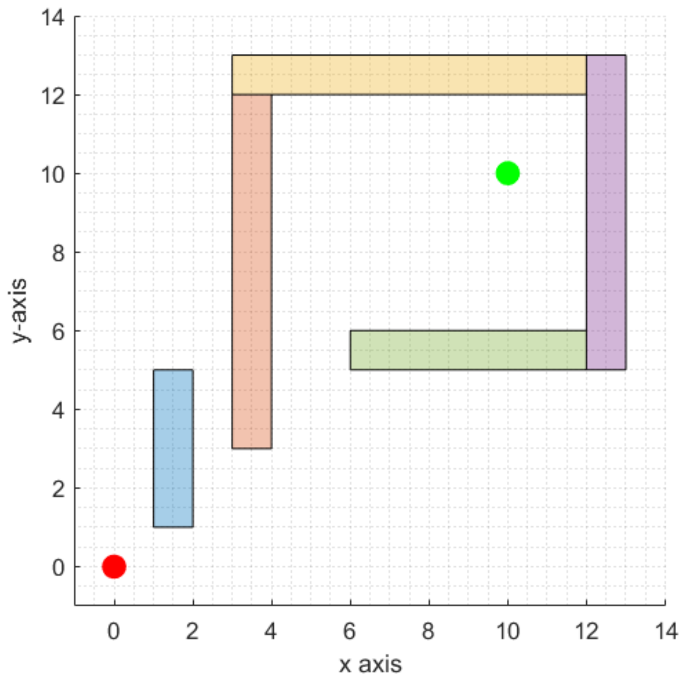
iv. The length of the path is 13.139

v. No, I would not expect the same path lengths for different d* and Q*s. If, for example, I was to choose small Q*s for both objects, this would nullify the influence of the obstacles. This occurs because the line between goal and start has no obstacles in the way. The path would follow the x-axis towards goal and have a path length of 9.75 (epsilon = 0.25).

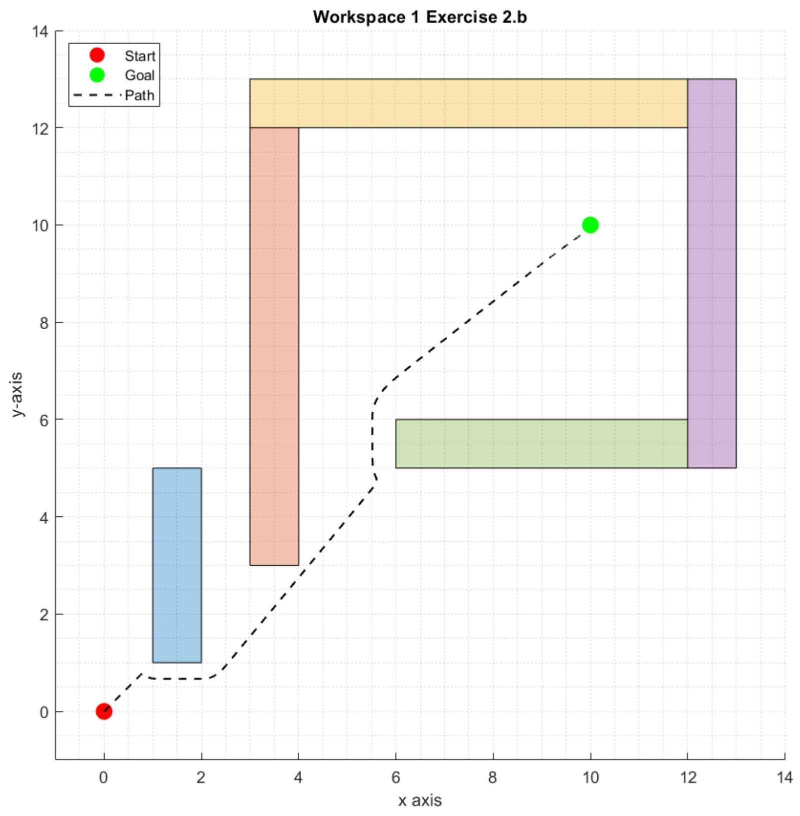
b)

Workspace 1

i. Picking d* and Q*s was less trivial than part a) due to the fact that our path directly intersects on obstacle at a corner. Additionally, this corner is on the m-line so the attractive forces of the goal will keep the robot on the m-line indefinitely unless we use the other obstacles to influence the gradient near the first obstacle. More specifically we need to push the robot below the m-line because this will allow the robot to fall into the pocket of obstacles. I needed to do this using the closest point to the robot from the obstacles. By inspection, the only obstacle that can push us below the m-line using the vector from its closest point to the robot is the lower left corner of the yellow obstacle in the below image. This means that my Q* for the yellow obstacles had to be larger than the distance between the start and the lower left of the yellow obstacle, 13.0. Additionally, the Q* for the blue, orange, purple, and green obstacles have to be low enough to give the influence. Using this qualitive approach I randomly used values and found the that the Q* values of obstacles 1-5 (blue, orange, yellow, purple, and green) are [0.5, 0.1 , 13.0 , 2.0 , 1.0], respectively.



ii. Workspace path 1



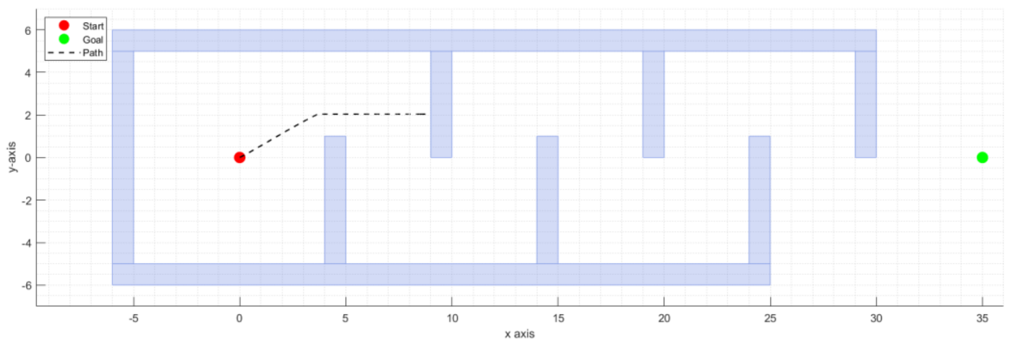
iii. The path length is 14.99

iv. No, just like part a), small changed to either Q*s chosen we can push the path around obstacles a little different to change the path distance slightly. For slightly different values for the Q*s we get a path length of 15.24.

<u>Workspace 2</u>

i. I was not able to get a path for this map.

ii. Below is a plot of the path I made. This path does not reach goal as I was not able to find d* and Q* values that worked.



iii. The generated path length is 10.84

iv. Yes, the path length would change for a different set of d* and Q*s. By choosing different values I was able to push the path in different directions but never fully escaping the start region of the map.