RL HW1 Wet

Omer Cohen 308428127
Dvir Marsh 313592685

April 2022

1 Exercise 3

1.1 3.1.1

The number of possible configurations of the tile is 9!. Including the blank tile we can place 9 tiles in each slot. We don't have to build the whole graph in advance because of the way Dijkstra algorithm works. In each iteration we add to our graph only one of the neighbours of $\mathcal{S}$.We can omit all the vertexes v with $d[v] = \infty$ and it won't hurt the algorithm correctness.

1.2 3.2.1

Let us define Q_i and P_i as the location of the number i on the grids Q, P respectively, where Q is the current state grid and P is the goal grid, Q and P are 3×3 grids. We will suggest the following heuristic:

$$H(Q) = \sum_{i=1}^{8} MD(Q_i, P_i)$$

Because every consistent function is also admissible, we will show consistency: for every adjacent vertexes u,v:

$$c(u, v) + H(u) - H(v) \geq 0$$

$$c(u, v) + H(u) - H(v) = 1 + \sum_{i=1}^{8} MD(u_i, d_i) - \sum_{i=1}^{8} MD(v_i, d_i) = \circledast$$

where d is the destination vertex, and u,v are two adjacent vertexes.
Because u,v are adjacent they are differ in only 1 element, therefore:

$$\circledast = 1 + MD(u_i, d_i) - MD(v_i, d_i) = 1 \pm 1 \geq 0$$

Therefore, this function is consistent and also admissible.

1.3 3.2.3

Fist we will show will show admissibility with consistency.
The current heuristic is:

$$H(Q) = \sum_{i=1}^{8} \mathbb{1}\{Q_i \neq P_i\}$$

where Q is some state, P is the goal state, Q_i is the location of the digit i in some state Q.

$$c(u,v) + H(u) + H(v) = 1 + \sum_{i=1}^{8} \mathbb{1}\{u_i \neq d_i\} - \sum_{i=1}^{8} \mathbb{1}\{v_i \neq v_i\} = \circledast$$

because u,v are adjacent vertexes so one digit (only one) must change its location:

$$\circledast = 1 + \mathbb{1}\{u_i \neq d_i\} - \mathbb{1}\{v_i \neq d_i\} = 1 \pm 1 \geq 0$$

The function is consistent, therefore it is admissible.
The number of states visited with Manhattan distance: 222.
The number of states visited with incorrect tiles distance: 1816
Obviously the Manhattan heuristic is better. It is better because it is more informative. Its tells us how far a wrong tile is from the goal tile.

1.4 3.2.4

Initial position:
0 1 2
3 4 5
6 7 8
Goal position:
8 7 6
5 4 3
2 1 0
Dijkstra running time and state visited: 0:00:47, 176545
A star running time and state visited: 0:00:00.25, 727

1.5 3.2.5

In general, for every α h_α might not be admissible.
for $\alpha = 0$ we are doing Dijkstra.
for $\alpha = 1$ we are doing A star.
for ≥ 1 the function might not be admissible and therefor the solution it provides might not be the shortest path. When $\alpha \to \infty$ the solution will be fastest but very likely won't be the shortest path. Therefor, α set a trade-off between

the solution time and how close is it to the shortest possible path.

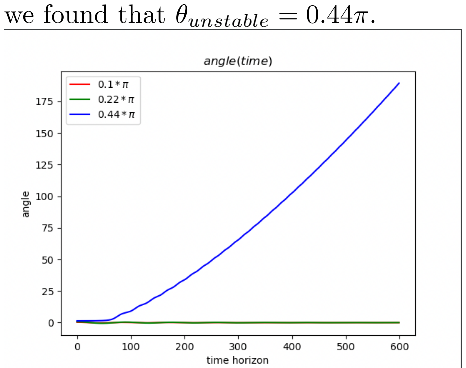
α value	length of paths	planning time	explored nodes
$\alpha = 0$	19	00:04:48	30242
$\alpha = 1$	19	00:00:00.46	222
$\alpha = \infty$	21	~ 0	53

1.6 3.3.2

Our main goal is to keep θ small so we gave relatively high penalty for high theta (0.1), in addition we wanted to give a low penalty to the position of the cart and the use of force therefor (2e-8,5e-7)

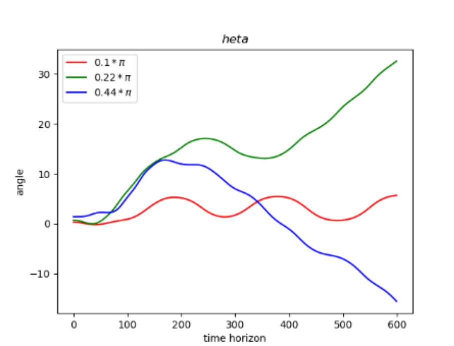
1.7 3.3.3

we found that $\theta_{unstable} = 0.44\pi$.



The graph shows that when $\theta > \theta_{unstable}$ the actual theta the pole isnt stable, otherwise- the pole is stable.

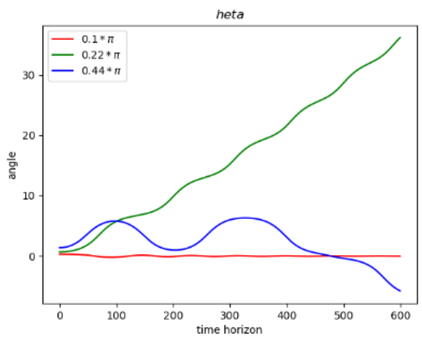
1.8 3.3.5



We can see that this method does not converge for any of the given thetas. The first method did converge for some of the thetas so obviously it is better.

1.9 3.3.5

It worked for us with the original parameters, but in general because the maximal allowed force is smaller we need to give a higher penalty for the use of force.



We can see that now for $\theta = \frac{\theta_{unstable}}{2}$ the method doesn't converge.