Solutions for HW4

Yunhai Han

January 31, 2020

1 E3.1 Shortest paths and distances on the circle

From the hint, if the two points are on an unit circle, we don't have to take its radius into consideration, because the length of arc is the product of the radius and the radian.

1.1 i

It's easy to compute the both distance, so I put the answers below:

$$dist_{cc}(\theta_1, \theta_2) = \frac{11\pi}{12} \text{ and } dist_{c}(\theta_1, \theta_2) = \frac{13\pi}{12}$$

1.2 ii

First we could use the operator *modulo* to shift any arbitrary θ into the region $[0, 2\pi)$ by taking the remainder. Since any θ on the circle is represented in the interval $[-\pi, \pi)$, we need to subtract π from the remainder.

We preprocess any input data θ_1 and θ_2 with the above method, and then compute the counter-clockwise distance from the new θ_1 and θ_2. There are the two possible cases:

- θ_1 is larger than θ_2
- θ_1 is less than θ_2

For the first case, we could compute the counter-clockwise distance by the following formula:

$$dist_{cc}(\theta_1, \theta_2) = \theta_2 - \theta_1 + 2\pi$$

For the second case, we could compute the counter-clockwise distance by the following formula:

$$dist_{cc}(\theta_1, \theta_2) = \theta_2 - \theta_1$$

1.3 iii

Similarly, the first step is to shift any arbitrary θ into the region $[-\pi, \pi)$ and this can be done with the same method. Then we could take advantage of the following property to help us compute the clockwise distance:

$$dist_{cc}(\theta_1, \theta_2) = dist_c(\theta_2, \theta_1)$$

Hence, it is equivalent for us to compute the counter-clockwise distance from θ_2 to θ_1. Since we have already provided the formula for this, this would be easy to swap the two variables and obtain the results.

1.4 iv

Since we know how to compute the counter-clockwise and clockwise distance between any two θ, we compare the two results and then select the smallest one as the distance between the two angles. Also, we could implement this algorithm in another way:

$$dist_{circle}(\theta_1, \theta_2) = \textbf{min}(dict_{cc}(\theta_1, \theta_2), dict_c(\theta_1, \theta_2))$$

$$dist_{circle}(\theta_1, \theta_2) = \textbf{min}(dict_{cc}(\theta_1, \theta_2), dict_{cc}(\theta_2, \theta_1))$$

$$\textbf{if } \theta_1 > \theta 2 : dist_{cc}(\theta_1, \theta_2) = -(\theta_1 - \theta_2) + 2\pi \quad \textbf{and} \quad dict_{cc}(\theta_2, \theta_1) = \theta_1 - \theta_2$$

$$\textbf{if } \theta_1 \leq \theta 2 : \text{just swap the positions of } \theta_1 \text{ and } \theta_2$$

From the above equations, we could quickly compute the distance between any two angles.

2 The configuration space of the new robot

There are two different kinds of flat disk robots model: the first one is the translating planar robot and the other is roto-translation planar robots. For the first one, the configuration space is $Q = R^2$ and for the second one, the configuration space is $Q = R^2 \times S^1$. For the 2-link robot, the manipulators with its configuration space are referred to as S̈elective Compliance Assembly Robot Armör SCARA in brief. In the figure 315, the robot arm has two revolute joints and a vertical prismatic joint, so the configuration space could contain three variables. Two of such variables describe the revolution and the other one describes the prismatic motion.

Hence, in total, this new robot's configuration space could contain five or six variables, which depends on the model for the flat disk robot. The position of the end effector is represented by R^3 in the workspace and once we determine the configuration space and build the forward and inverse kinematic maps, we could translate the position into the configuration space for the motion planning task.

3 Torus

3.1 a

From the lecture, we know that the vertical on the left is identified with the vertical segment on the right and the horizontal segment on the top is identified with horizontal segment on the bottom. So, it is easy to tell which nodes represent the same point on the torus. I draw a picture for the illustration.

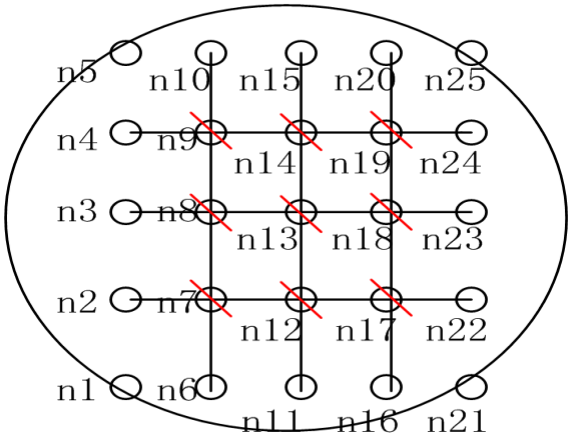


Figure 1: Nodes and edges

Each line connects the two nodes which represent the same point on the torus. For example, the node $n_{11} = (0, -\pi)$ and the node $n_{15} = (0, \pi)$ represent the same point. Besides, you can see that the four nodes on the four corners indeed represent the same point.

3.2 b

If we are allowed to deviate from the nodes in the graph, the shortest path between the two nodes are the direct segment connecting the two nodes. If we are not allowed to do that, which means it needs to connect different nodes, the shortest path between the points n_7 and n_{19} is shown as below:

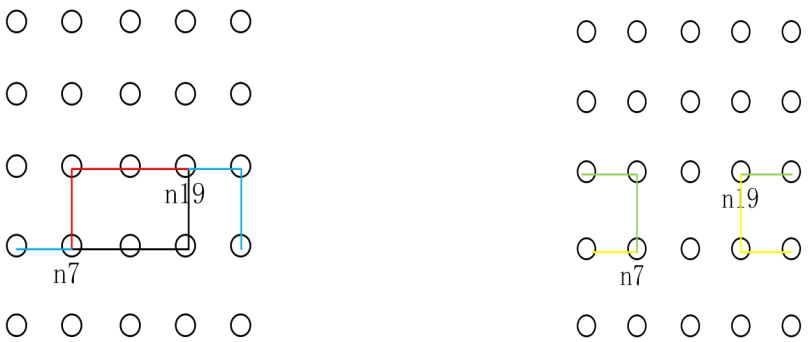
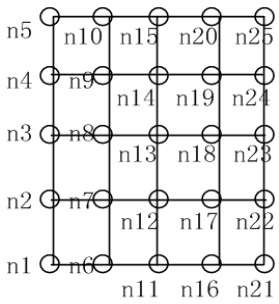


Figure 2: The shortest path

There are in total five different shortest paths. The shortest distance is three steps.

3.3 c

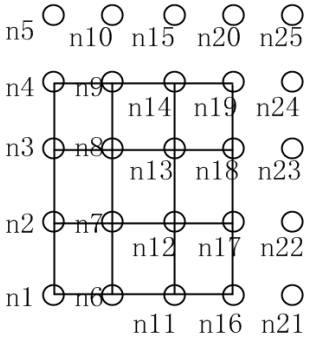
If the edges are defined as the problem says, only the neighboring nodes are connected. Besides, since the square is defined as a torus, the vertical segment on the left is identical to the one on the right and it's the same for the horizontal segment. To combine these two constraints, we could list the set of all the edges:



(n1, n2) (n1, n6) (n11, n12) (n11, n16) (n21, n22)
(n2, n3) (n2, n7) (n12, n13) (n12, n17) (n22, n23)
(n3, n4) (n3, n8) (n13, n14) (n13, n18) (n23, n24)
(n4, n5) (n4, n9) (n14, n15) (n14, n19) (n24, n25)
(n5, n10) (n15, n20)
(n6, n7) (n6, n11) (n16, n17) (n16, n21)
(n7, n8) (n7, n12) (n17, n18) (n17, n22)
(n8, n9) (n8, n13) (n18, n19) (n18, n23)
(n9, n10) (n9, n14) (n19, n20) (n19, n24)
(n10, n15) (n20, n25)

Figure 3: Graph edges and nodes

From the above figure, only the neighboring nodes could be connected via an edge. There are 40 edges totally. However, for example, we know that the node n_1, n_5, n_{21} and n_{25} represent the same point, so we could modify this graph in order to take this constraint into consideration.

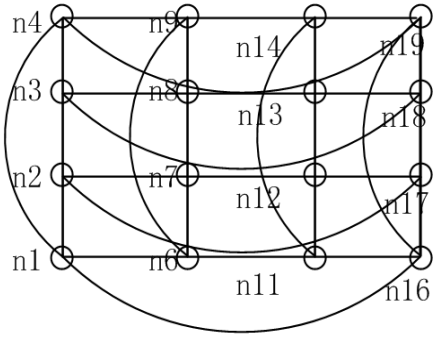


(n1, n2) (n1, n6) (n11, n12) (n11, n16) (n1, n2)
(n2, n3) (n2, n7) (n12, n13) (n12, n17) (n2, n3)
(n3, n4) (n3, n8) (n13, n14) (n13, n18) (n3, n4)
(n4, n1) (n4, n9) (n14, n11) (n14, n19) (n4, n1)
(n1, n6) (n11, n16)

(n6, n7) (n6, n11) (n16, n17) (n16, n1)
(n7, n8) (n7, n12) (n17, n18) (n17, n2)
(n8, n9) (n8, n13) (n18, n19) (n18, n3)
(n9, n6) (n9, n14) (n19, n16) (n19, n4)
(n6, n11) (n16, n1)

Figure 4: Graph edges and nodes

I remove the nodes $n_5, n_{10}, n_{15}, n_{20}, n_{25}, n_{21}, n_{22}, n_{23}, n_{24}$ because they represent the same nodes as n_1, n_2, n_3 and n_4, respectively. You can see after this operation, some edges are identical, for example, there are two (n_1, n_6) in the set. I remove these edges which appear for more than one time and draw the modified graph edges and nodes.



(n1, n2) (n1, n6) (n11, n12) (n11, n16)
(n2, n3) (n2, n7) (n12, n13) (n12, n17)
(n3, n4) (n3, n8) (n13, n14) (n13, n18)
(n4, n1) (n4, n9) (n14, n11) (n14, n19)
(n6, n7) (n6, n11) (n16, n17) (n16, n1)
(n7, n8) (n7, n12) (n17, n18) (n17, n2)
(n8, n9) (n8, n13) (n18, n19) (n18, n3)
(n9, n6) (n9, n14) (n19, n16) (n19, n4)

Figure 5: Graph edges and nodes

From the above figure, there are in total 32 edges and 16 nodes.

3.4 d

The neighbors of node n_1 are nodes n_2, n_6, n_4 and node n_{16}. The node n_{22} is identical to the node n_2 and the neighbors of node n_2 are nodes n_{17}, n_1, n_3 and node n_7.

3.5 e

Each figure below corresponds to each layer and I mark the edges by red segments.

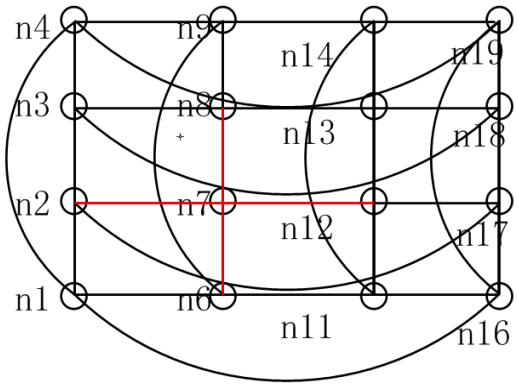


Figure 6: Layer 0

For the first layer, we start from the node n_7, and there are four edges connecting n_7 to other nodes. These four nodes would be set as the start node for the next exploration.

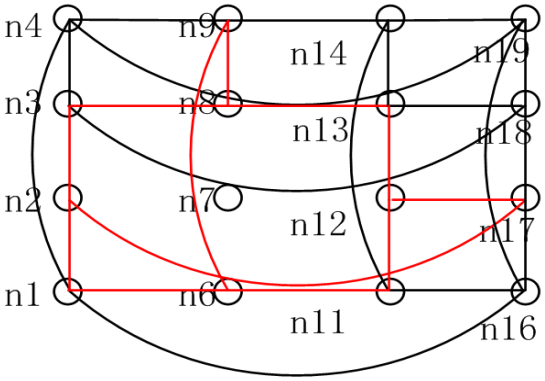


Figure 7: Layer 1

For the second layer, we start from the four nodes detected in the previous step. In the figure, you could see that the node n_3 is connected to both n_2 and n_8, but in fact, when we run BFS algorithm, the first time n_3 visited, its parent node would be set. Then, for the next time, it would not be detected since its parent node is not NONE any

more. It's the same for the node n_1, n_9, n_{13}, n_{17} and n_{11}. Because it is unnecessary for me to decide the operation order, I just connect all of them.

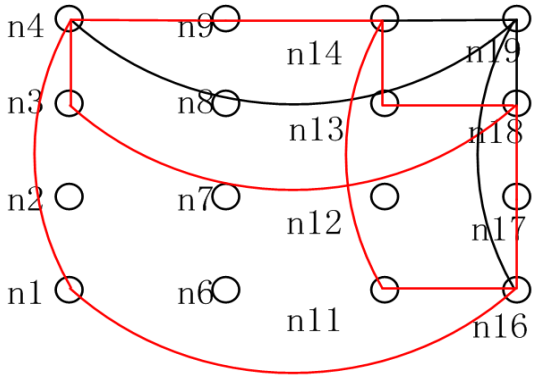


Figure 8: Layer 2

For the same reason, the node n_4, n_{14}, n_{18} and n_{16} would be only visited once.

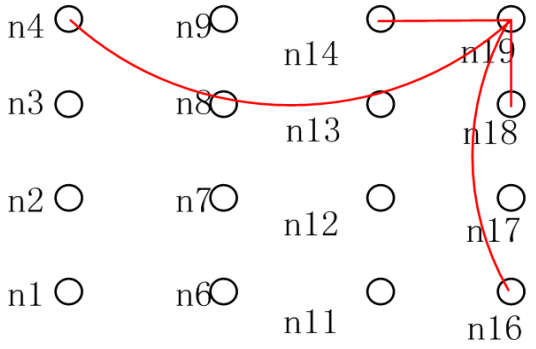


Figure 9: Layer 2

For the same reason, the node n_{19} would be only visited once. After this iteration, Q is empty and the algorithm terminates.

Finally, we successfully search through all the graph and since all the nodes in the graph are reachable, this graph is connected.