Assignment 1

Geometric Motion Planning

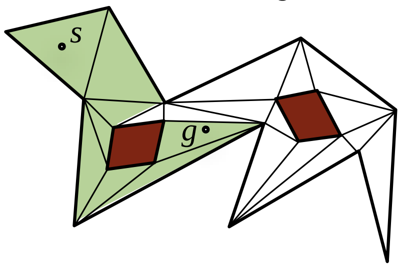
Name:	Adarsh Srivastava
Matriculation Number:	A0254358X

Instructions

- Due date: 14 Feb 2023 in class.
- Type up your solutions or write neatly.
- Submit your work in hard copy. Make sure to include the cover page.
- The starred questions provide extra-credits.

Problem	Max Points	Points
1	10	
2	10	
3	10	
4	15	
5	15	
6	15*	
7	25	
8	10	
9	15	
10	10	
11	10 + 5* (proof)	
Total	130+20*	

1. For a point robot in a 2-D polygonal environment with obstacles, its free space is decomposed into and represented as a set of triangles. A *channel* is a sequence of adjacent triangles that contains a collision-free path connecting a start position s and a goal position g.



(a) Define a search graph G over the triangles so that we can use G to find a channel σ that connects s and g. Describe what the nodes and edges of G represent.

(b) Given a channel σ that connects s and g, describe a method to generate a collision-free path from s to g. Draw an example to illustrate your solution.

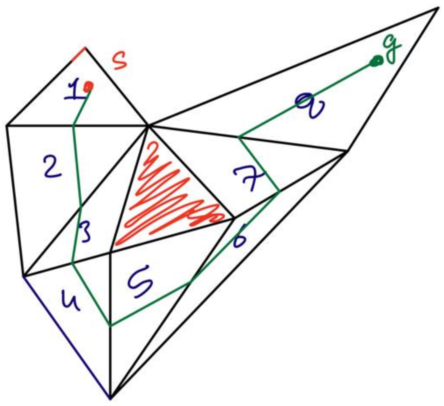
<u>Solution:</u>

(a) Graph G = (V, E) where,

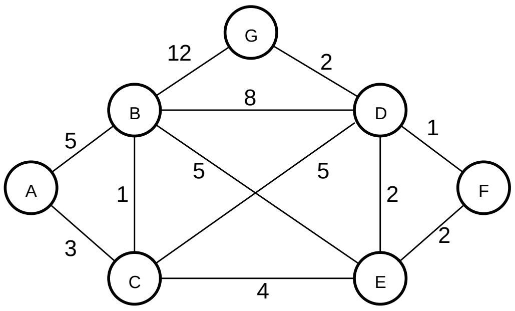
V = Set of nodes where every node corresponds to a free (non-obstacle) triangle.

E = Set of edges between all such nodes $v_1, v_2 \in V$ that correspond to adjacent triangles (sharing a side) in the triangular environment.

(b) One collision free path generation strategy is: Travel in a straight line to the midpoint of the side of the current triangular cell that is shared between current cell and next cell in the channel σ. For example:



2. Find the shortest path to node A from every other node in the weighted graph below.



(a) Apply the backward dynamic programming algorithm. Let $V^*(s)$ be the shortest-path length from node s to A and $V_i(s)$ be the estimated shortest-path length in the i'th iteration of the dynamic programming algorithm. Show the values for $V_0(s)$, $V_1(s)$, and $V_2(s)$ as well as $V^*(s)$ for all nodes in the graph.

(b) Apply the Dijkstra's algorithm. The shortest paths form a tree. Draw the shortest-path tree.

Solution:

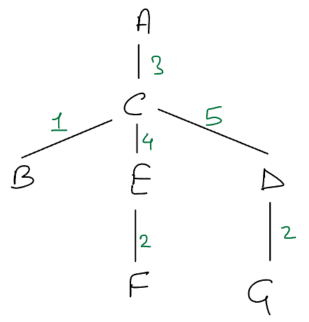
(a)

	A	B	C	D	E	F	G
V_0	0	+INF	+INF	+INF	+INF	+INF	+INF
V_1	0	5	3	+INF	+INF	+INF	+INF
V_2	0	4	3	8	7	+INF	17
V_*	0	4	3	8	7	9	10

(b)

Current (popped) node	**Priority Queue Deduped (Node, Cost)**	**Path to A**	**Cost to A for current node**
A	(C, 3) , (B, 5)	A	0
C	(B, 4), (E, 7), (D, 8)	A <- C	3
B	(E, 7), (D, 8), (G, 16)	A <- C <- B	4
E	(D, 8), (F, 9), (G, 16)	A <- C <- E	7
D	(F, 9), (G, 10)	A <- C <- D	8
F	(G, 10)	A <- C <- E <- F	9
G	*<empty>*	A <- C <- D <- G	10

Corresponding shortest paths tree is:



3. Let M be a 3×3 orthonormal matrices with determinant +1. We have discussed in the class that every such matrix corresponds to a rotation in 3-D space, and vice versa. This implies that M has only 3 *independent* degrees of freedom (DOFs); however, M contains 9 parameters.
   (a) What are the constraints on the 9 parameters that reduce the number of DOFs of M to 3?
   (b) M is required to have determinant +1. Does this constraint reduce the number of DOFs of M? Why or why not?

Solution:

(a)

Let M = $\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$

Since this is an orthonormal matrix,

All columns are orthogonal to each other, i.e.,

$$a_{11}a_{12} + a_{21}a_{22} + a_{31}a_{32} = 0$$
$$a_{11}a_{13} + a_{21}a_{23} + a_{31}a_{33} = 0$$
$$a_{12}a_{13} + a_{22}a_{33} + a_{32}a_{33} = 0$$

And the vectors are normalized unit vectors, so:

$$a_{11}^2 + a_{21}^2 + a_{31}^2 = 1$$
$$a_{12}^2 + a_{22}^2 + a_{32}^2 = 1$$
$$a_{13}^2 + a_{23}^2 + a_{33}^2 = 1$$

These are 6 independent constraints which reduce DoF from 9 to 3.

(b) The determinant comes as a by-product of the above equations. We can use matrix multiplication and the above 6 equations to verify that:

$$M^{\top}M = I$$
$$\Rightarrow \det(M^{\top}M) = \det(I)$$
$$\Rightarrow \det(M^{\top})\det(M) = 1$$
$$\Rightarrow (\det(M))^2 = 1$$
$$\Rightarrow \det(M) = \pm 1$$

Thus, this is not an independent new constraint, and so does not reduce the DoF of the system.

4. Give the dimension of the configuration space for the following systems. Briefly justify your answer.
   (*a*) An articulated robot in a 2D plane with a fixed base and two revolute joints.
   (b) Two mobile robots freely translate and rotate in the plane.
   (c) An aerial manipulator consisting of two manipulators attached to an unmanned aerial vehicle (UAV). Each manipulator has 6 revolute joints.

Solution:

(a) 2. Base is fixed, so no translation. At any point of time, specifying the angles of the two revolute joints is enough to completely specify the configuration. Further, since they take values unconstrained from each other, it is not possible to reduce the number of parameters further.

(b) 6. Assuming non-point robots, translation is 2 dof for each robot and rotation is another dof each.

(c) 18. 3 dof for UAV translation. 3 dof for 3D space rotation of UAV (yaw, pitch, roll). Additionally, one angle per revolute joint for each arm = 6x2 + 3 + 3

5. This problem explores the configuration space of lines and that of line segments.
   a. Consider an infinite line ℓ that translates and rotates freely in 3-D space. Give two different parameterizations of the configuration space C for ℓ: one that makes use of angles and one that makes no use of angles.
   b. What is the dimension of C?
   c. Consider a straight-line segment s that translates and rotates freely in 3-D space. What is the dimension of the configuration space for s? Can you use the two parameterizations in part (*a*) for s? What modifications would be needed if any?

Solution:

(a) With angles:
Consider a sphere of radius r centred at the origin. Consider a point on the sphere specified by two angles (latitude and longitude) theta and phi. This point uniquely specified the plane tangent to the sphere and touching the sphere at this point. Now to specify any line on this plane, we can use one more parameter omega – the angle wrt some axis on this plane.

These four params - *r, theta, phi, omega* uniquely specify any line in a 3D plane.

Without angles:
Choose any point on the line in the 3D plane – (x_0, y_0, z_0). Choose a unit vector pointing in the direction of the line (v_x, v_y, v_z) (three params with one constraint of being a unit vector). This can uniquely specify a line.

(b) 4. Any configuration of the line can be specified in no less than 4 params, like the first example above.

(c) For a line segment, aside from the unit vector (2 dimensions), we need a point (3 dimensions) (either start or end) and the length of the line (1 dimension). Thus we need at least 6 params – so dimension is 6.

The parameterizations cannot be directly used. We need to either add length and one point or two points constrained by distance between them.

6. This problem examines the relationship between distance in the configuration space and distance in the workspace. Specifically, if a robot moves by a certain amount in the configuration space, how much can a point on the robot move in the workspace? Consider a planar robot arm with n sequential links. Each link is a straight-line segment of length L. One endpoint of the link is called the *origin*, and the other is called the *extremity*. The origin of the first link is fixed. The origin of the ith link ($2 \le i \le n$) coincides with the extremity of the $(i-1)$th link at a point called a *joint*. A link can rotate freely about the joint.

   (a) A configuration q of this robot can be represented by the joint angles $(\theta_1,\theta_2,\ldots,\theta_n)$. The metric d_c in the robot's configuration space is defined as

   $$d_c(q,q') = \max_{1\le i\le n} |\theta_i - \theta_i'|$$

   for two configurations q and q'. Suppose that the robot moves from a configuration $q = (\theta_1,\theta_2,\ldots,\theta_n)$ to a configuration $q' = (\theta_1',\theta_2',\ldots,\theta_n')$ along the straight-line segment joining q and q' in the Cartesian space R^n. In other words, the robot moves along the path $(1-\lambda)q + \lambda q'$ for $0 \le \lambda \le 1$. Show that no point on the robot traces a path longer than $\alpha d_c(q,q')$ for some positive constant α. Give a bound of α in terms of L, the link length, and n, the number of links.

   (b) Let $d_w(q,B)$ denote the minimum distance between the robot placed at a configuration q and a (workspace) obstacle B, *i.e.*, the distance between the closest pair of points on the robot placed at q and B. Using the result from part (*a*), calculate the radius ρ of the neighborhood

   $$N(q) = \{q' \mid d_c(q,q') \le \rho\}$$

   in which the robot is guaranteed to move freely without colliding with B. Express your answer in terms of α and $d_w(q,B)$.

<u>Solution:</u>

(a) $d_c(q,q')$ is the maximum change in theta of any of the links in the arm.

Straight line between configurations q and q' means the robot directly moves from q to q' without first contorting into a weird intermediate configuration.

Let's consider the case where, for a given change in config and $d_c(q,q')$, some point on the arm traces the longest possible path of any point for the given change.

Given the links configuration, this will happen when $d_c(q,q')$ = change in theta for the first link, and the rest of the arm staying as extended as possible (i.e., all further links line up to create a linear straight line – if it were not so, due to the constraint, any angle in any further link would reduce the reach of the arm – but we are considering max reach case.)

In this case, the point at the tip of the last link traces a path of length:

$$(n-1)L\, d_c(q,q')$$

Since the first link is not movable. By basic geometry, in this config, any other point will trace a path proportional to it's distance from moving (first) joint, and so would be less than the above number.

For rotation along any other joint, the resulting circle's radius would be smaller, and therefore would be again smaller than this number.

7. Describe how to sample points uniformly from each set below. For part (*a*) and (*b*), use polar-coordinates parametrization.
   (a) A circle with center c and radius r: $S = \{p \in R^2 \mid \|p - c\| = r\}$.
   (b) A disc B_2 with center c and radius r: $B_2 = \{p \in R^2 \mid \|p - c\| \leq r\}$.
   (c) Alternatively, sample the disc B_2 using the rejection method. Sample uniformly from the bounding box of B containing B_2 and reject the samples outside of the disc. Which method would you choose, polar-coordinates parametrization or rejection sampling? Why?
   (d) An N-dimensional sphere with center c and radius r: $B_N = \{p \in R^N \mid \|p - c\| \leq r\}$. If using rejection sampling, discuss the rejection rate and how it scales with the dimension of the space, N.
   (e) Can you think of a way to sample B_N without rejection for arbitrary large N?

Solution:

(a) By sampling uniformly between 0 and 2*pi (using scaling)– and using this number as the angle from some axis of the line joining the sampled point and center.

(b) Uniformly sampling two params – theta (the angle wrt some axis, between 0 and 2*pi) and d (the distance from center, between 0 and r)

(c) Both are 2 dimensional and equally easy to sample and generate equally uniform samples. Rejection sampling wastes samples, so I will choose polar coordinates.

(d) Here, sampling directly becomes difficult with N dimension, whereas sampling for an N dimensional hypercube is easy (n times sampling between 0 and side length). Hence will choose rejection sampling.

Since distance between points blows up as dimensions increase, so with high dimensions a lot more points will be sampled outside the sphere and will need to be rejected.

(e)

8. Suppose that the configuration space C is the unit square $[0,1] \times [0,1]$. The multi-query PRM algorithm, first samples n collision-free configurations and then tries to connect these milestones by calling LINK.
   (a) If the algorithm calls LINK for every pair of roadmap nodes, give an asymptotic upper bound on the number of calls to LINK.
   (b) Suppose that the algorithms calls LINK only if the Euclidean distance between two milestones is smaller than a threshold t. Give an asymptotic bound on the number of calls to LINK when $t = O(1/\sqrt{n})$. You may assume that the milestones are distributed roughly uniformly in C.

   Solution:

   (a) n(n-1)

   (b) $t = O(1/\sqrt{n})$

   Let's draw a circle around each point of threshold t.

   Area of circle = πt^2

   $= \pi/n$

   We sampled a total of n points uniformly. Assuming interior point and large n, number of points inside the circle should be proportional to the area of circle wrt the area of square (1).

   So, expected number of points inside circle = $(\pi/n) * n = \pi$

   Let this point call each of these points in Link.

   We now have two link calls between each point – pairwise we need one.

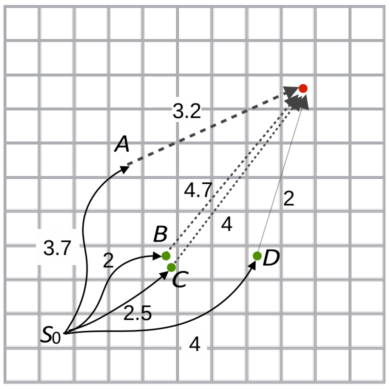
   So, number of link calls for n points = $\pi n/2$

9. Among the motion planning algorithms discussed in the class (dynamic programming, A*, PRM, EST, and RRT), choose the best algorithm for the robot motion planning tasks below. Justify your choice by considering the configuration space dimensions, environment characteristics, computational efficiency, ...
   (*a*) A robot car trying to park itself in an open parking slot.
   (b) Because of the COVID-19 pandemic, a cleaning robot must move around and disinfect the busy areas of the hospital every 2 hours.
   (c) A self-reconfigurable modular robot consists of many identical modules and can reconfigure its shape to fit a task. See the video for an example.

   Solution:

   (a) Hybrid A*, due to low dimensionality (3) and static environment
   (b) Low dimensionality but very dynamic environment, so EST
   (c) High dimension and likely dynamic environment – RRT.

10. Consider the hybrid A* algorithm described in the class and apply it to plan the motion of an autonomous robot car. Suppose that the A* search starts at the node S, shown in the figure below. By applying four candidate actions, it reaches new nodes: A,B,C, and D. The cost-to-come and the heuristic estimate of the cost-to-go for all the new nodes are shown in the figure.



(a) What is the dimension of the grid used in the hybrid A* search?

(b) What does the priority queue contain at the stage of the hybrid A* search illustrated in the figure? For each item, in the priority queue, specify the node and its associated f-value.

Solution:

(a) 3. X, y, theta

(b)

P-Queue (Node, F value)
D, 6.0
C, 6.5
B, 6.7
A, 6.9

11. A key step in applying the A* algorithm in practice is to design a good heuristic function. Suppose that we want to apply A* to a shortest-path problem and have two heuristic functions $h_1(x)$ and $h_2(x)$, both of which are admissible.
    (a) Show that $h(x) = \max\{h_1(x), h_2(x)\}$ is also admissible.
    (b) Of the three heuristic functions, $h_1(x), h_2(x)$, and $h(x)$, which one would you use? Why? Give a proof if you can.

Solution:

(a) Admissible means NEVER overestimates.
Let C(x) be the true cost function from x.
So $h_1(x)$ *<= C(x) and* $h_2(x)$ *<= C(x) for all x.*
It follows that for any x, the maximum of the two will also be *<= C(x)*
Therefore, $h_3(x)$ *<= C(x)*

(b) Admissibility guarantees no overestimation. After this, we want the heuristic to be as close to C(x) as possible for good results. So, choosing the higher value of several admissible heuristics makes sense. Hence, I will choose h3(x).