Assignment 1

Search, Combinatorial Planners and Configuration Space Abstraction

Deadline: October 5, noon, i.e., 12pm
Perfect score: 100 points for CS460 and 120 points for CS560

Assignment Instructions:

Assignments should be completed **individually** by each student.

Submission Process: Submit your reports electronically as a PDF document through Sakai (`sakai.rutgers.edu`). For your reports, do not submit Word documents, raw text, or hardcopies etc. Make sure to generate and submit a PDF instead. On the first page of the PDF indicate your name, whether you are taking CS460 or CS560 and your Net ID. Failure to follow these rules will result in a lower grade in the assignment.

Late Submissions: No late submissions are allowed. You will be awarded 0 points for late assignments.

Extra Credit for LaTeX: You will receive 4% extra credit points if you submit your answers as a typeset PDF (i.e., using LaTeX). Resources on how to use LaTeX are available on the course's website. There will be a 2% bonus for electronically prepared answers (e.g., on MS Word, etc.) that are not typeset. Have in mind that these bonuses are computed as percentage of your original grade, i.e., if you were to receive 50 points and you have typesetted your report using LaTeX then you get 2 points bonus. If you want to submit a handwritten report, scan it and submit a PDF via Sakai. We will not accept hardcopies. If you choose to submit scanned handwritten answers and we are not able to read them, you will not be awarded any points for the part of the solution that is unreadable.

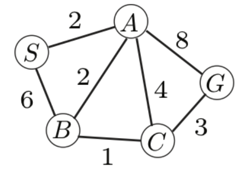
Precision: Try to be precise in your description and thorough in your evaluation. Have in mind that you are trying to convince skeptical evaluators (i.e., computer scientists...) that your answers are correct.

Collusion, Plagiarism, etc.: Each student must prepare their solutions independently from others, i.e., without using common code, notes or worksheets with other students. You can discuss material about the class that relates to the assignment but you should not be provided the answers by other students and you must write down your answers independently. Furthermore, you must indicate any external sources you have used in the preparation of your solution. This includes both online sources or discussions with other students. Unless explicitly allowed by the assignment, do not plagiarize online sources and in general make sure you do not violate any of the academic standards of the course, the department or the university (the standards are available through the course's website). Failure to follow these rules may result in failure in the course.

Course's website:
`https://robotics.cs.rutgers.edu/pracsys/courses/intro-to-computational-robotics/`

Assignment Description

Problem 1 [10 points]. **Search via Uniform-cost and A***. The figure on the right provides a graph with associated edge costs and a table with values of a heuristic function. Using the graph search algorithm covered in class, i.e., a search node may be truncated if it is known to be suboptimal, manually execute the procedures of i) uniform-cost search, and ii) A^* for a path from S to G. For each search process, describe incrementally how the search tree and the fringe (open list) data structure look like at each step of the algorithm (similar to the detailed A^* example in the lectures). Write down the final solution path with the cost. Include a brief discussion of your observations as you performed the search process, e.g., effects of tiebreaking, quality of solution found, impact of the heuristic, etc. You may draw your search tree manually and scan them into your document for submission.



State	$h(x)$
S	7
A	6
B	2
C	1
G	0

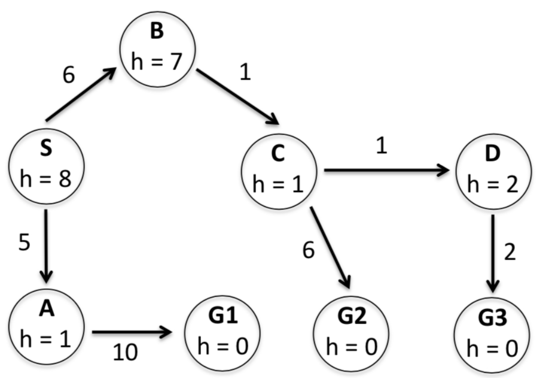
Problem 2 [20 points]. **Faster Informed Search with Bounded Error Guarantees.** We have described the A^* algorithm for informed search, which is guaranteed to return an optimal solution when given an admissible/consistent heuristic and is optimally efficient for the corresponding heuristic. Often in practical applications, however, it is still too expensive to find an optimal solution, so instead we search for good suboptimal solutions. It is still desirable in this case to have algorithms that the solution that they found is not arbitrarily suboptimal. Instead, the difference between the true optimal solution and the one found by the suboptimal algorithm, should be bounded.

A variant of A^* that is popular in practical applications computes the f-value (based on which the algorithm orders nodes n in the fringe data structure to be expanded) according to the following equation:

$$f(n) = g(n) + \epsilon \cdot h(n)$$

where $\epsilon \geq 1$ is a parameter given to the algorithm. In general, the larger the value of ϵ, the faster the search is and the higher the cost of the path to a goal node discovered. The heuristic does not acquire negative values.

1. Examine first how this variant works on the following graph, where S is the start node node and G_1, G_2, G_3 are alternative goal nodes:



   Execute the variant of A^* on the above graph with $\epsilon = 2$, and complete a table indicating the state of the fringe at every step of the algorithm (i.e., the ordered list of nodes in the fringe as well as their g, h, f values) and which node is expanded next.

2. After this variant is executed for $\epsilon \geq 1$, a goal node G is found of cost $g(G)$. Let C^* denote the optimal solution cost and assume that the heuristic is admissible/consistent. Among the following guarantees for the quality of the solution that this variant discovers, indicate the strongest bound for which you can provide a proof. Please include your proof in your submission.

   - $g(G) \leq \epsilon \cdot C^*$
   - $g(G) \leq C^* + \epsilon$
   - $g(G) \leq C^* + 2 \cdot \epsilon$

- $g(G) \leq 2^{\epsilon} \cdot C^*$
- $g(G) \leq \epsilon^2 \cdot C^*$

A proof sketch is sufficient for CS460 students. A formal, correct proof is needed by CS560 students.

3. This variant is equivalent to other algorithms as special cases. For each of the following, describe the behavior of the approach and mention equivalances to algorithms discussed in the class if they exist:

   i $\epsilon = 1$

   ii $\epsilon = 0$

   iii $\epsilon \to \infty$ (i.e., as ϵ becomes arbitrarily large)

Problem 3 [20 points]. **Combinatorial Planners** Draw a polygonal environment that represents your Rutgers Net ID. For instance, if your Rutgers Net ID is ab123, then draw a rectangular boundary and inside this rectangle draw 5 polygons, where each one of them corresponds to a polygonal version of a character in your Rutgers NetID, e.g., a polygon looking like an A, like a B, like 1, 2 and 3. Keep the number of vertices low per polygon. Treat the interior of the polygons representing your Rutgers Net ID as obstacles and the space between the boundary and these polygons as the free space for a point robot to move. Given this polygonal representation:

1. Draw the visibility graph and the reduced visibility graph for this environment. On your graph indicate the solution path in order to go from the top left corner of the boundary to the bottom right corner.

2. Draw a trapezoidal decomposition of the free space and the corresponding adjacency graph that arises from it. On your graph indicate the solution path in order to go from the top left corner of the boundary to the bottom right corner.

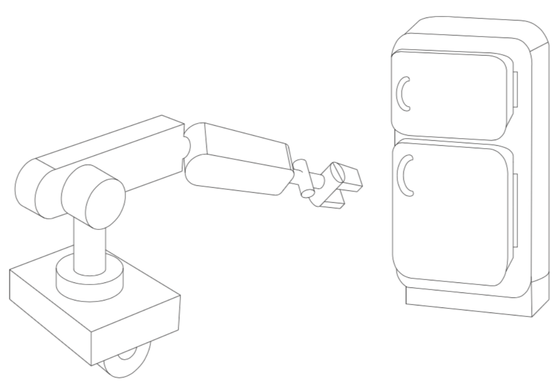
3. If you were to compute the Generalized Voronoi Diagram of the free part of this polygonal environment, estimate the number of vertices that the corresponding diagram would have.

You can manually draw the environment and the output of the above algorithms. In that case, scan them or take a photo of them to include them into your document for submission.

Problem 4 [20 points]. **Configuration space**.

i. We mentioned that a rigid body that can rotate and translate over a plane (e.g., a car) has a three dimensional configuration space abbreviated as $SE(2) = \mathbb{R}^2 \times S^1$. What if the corresponding car/body is operating on the surface of a three dimensional sphere? What is the C-space for the car in this case? Provide a brief justification of your answer.

ii. Assume each of your arms has n degrees of freedom. You are driving a car, your torso is stationary relative to the car (owing to a tight seatbelt!), and both hands are firmly grasping the wheel, so that your hands do not move relative to the wheel. How many degrees of freedom does your arms-plus-steering wheel system have? Explain your answer.



iii. The mobile manipulator of the Figure above consists of an arm with 6 rotational DoFs and a multi-fingered hand mounted on a mobile base with a single wheel. You can think of the wheeled base as the same as the rolling coin in our lecture slides. The wheel (and base) can spin together about an axis perpendicular to the ground, and the wheel rolls without slipping. The base always remains horizontal and grounded on the plane.

- Ignoring the multi-fingered hand, describe the configuration space of the mobile manipulator.
- Now suppose that the robot hand rigidly grasps a refrigerator door handle and, with its wheel and base completely stationary, opens the door using only its arm. With the door open, how many degrees of freedom does the mechanism formed by the arm and the open door have?
- A second identical mobile manipulator comes along, and after parking its mobile base, also rigidly grasps the refrigerator door handle. How many degrees of freedom does the mechanism formed by the two arms and the open refrigerator door have?

Problem 5 [15 points]. **C-space obstacles**. Let W be the bounded 2D square region with the bottom left corner being $(0, 0)$ and the top right corner being $(10, 10)$. There are 3 polygonal obstacles whose boundaries are defined by the following list of points in the clockwise direction:

1. $(1, 4), (4, 3), (4, 1)$,
2. $(6, 6), (7, 7), (8, 6), (7, 5)$,
3. $(2, 8), (4, 8), (4, 7), (2, 7)$.

Now consider a rectangular (rigid body) robot with width 0.6 and length 0.8 that only translates (i.e., the robot does not rotate) inside W. Assume that the point of reference for the robot is the center of this rectangular shape. Answer the following questions.

i Compute the free configuration space, $\mathcal{C}_{free}$, for the robot when it moves inside W. For instance, you can draw digitally or on paper $\mathcal{C}_{free}$.

ii Draw W, the obstacles, and the free configuration space $\mathcal{C}_{free}$ (you may use python to do this or draw it manually) in a figure. Is $\mathcal{C}_{free}$ connected? If not, how many components (i.e., continuous pieces) does $\mathcal{C}_{free}$ have?

iii Provide a collision free path, in the form of a list of points where the path turns, that moves the center of the rectangular robot from $x_I = (6.5, 2)$ to $x_G = (7, 9)$. That is, the path should be a sequence of straight line segments connecting the x_I and x_G; the points that you provide are the points where these lines segments meet. Draw your path on the same figure from the previous question.

Problem 6 [15 points]. **2D transformation and collision detection**. Let a rigid body be defined by the polygon formed with points $A = (8, 4), B = (10, 6), C = (12, 5), D = (11, 3), E = (9, 2)$ in the clockwise direction. Apply the transformation that rotates the rigid body by 40 degrees followed by a translation of (3, 4). Provide solution to the following questions.

i What is the transformation matrix T?

ii Where are the points A to E after the transformation?

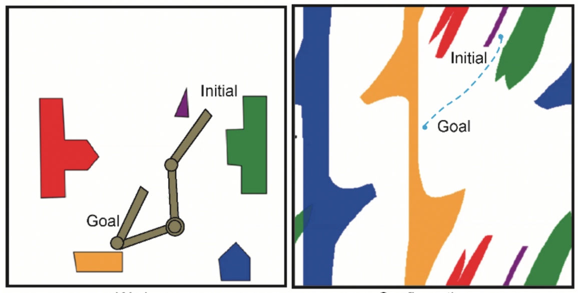
iii Does the transformed rigid body collide with the polygon formed by connecting the points $F = (3, 11), G = (9, 12), H = (6, 6), I = (2, 7)$ in the clockwise direction? If so, which edges of the two bodies intersect?

iv Suppose another transformation moves the original A to $(13, 8)$ and the original E to $(15, 9)$. What is the transformation matrix T' in this case?

Problem 7 [25 points]. **C-free of a two link arm**. [Optional for CS460 - Attempt this problem after you have tried the ones above] Consider the visualization we have covered during the lectures, which provides the configuration space of a 2-link planar arm in the presence of polygonal obstacles. Write a script (in python or in your preferred environment of choice) to generate similar visualizations of the C-space of a 2-link planar arm. Place the arm at the center of the workspace and a small number (3-5) of polygonal obstacles around it which allow some robot configurations to be collision free. Provide in your report at least a couple of examples of environments that you were able to compute a solution and a description of what did you have to implement in order to complete the visualization tool.

You can introduce some simplifications in this process - if you want to, they are not necessary:

i You can discretize the workspace into a grid, where some of the cells are occupied given the obstacles, while others are not.



Workspace Configuration space

1. Define a grid over the configuration space as well by discretizing each rotational DoF of the arm.

ii You can assume that each link of the robot is a one-dimensional, line segment.

iii For each cell in the configuration space, identify whether it is in the free part C_{free} for this robot or whether it leads into a collision with an obstacle (you can also color obstacles in the C-space differently similar to the figure above).