

Autonomous Navigation and Perception (086762)

Homework #4

Submission guidelines:

- Submission in pairs by: [27 April 2022](#).
- Email submission as a *single zip or pdf file* to anpsubmission@gmail.com.
- Make sure to name the submitted file ID1-ID2.zip (or .pdf) with ID1,ID2 - the students' ids.
- For hands-on parts, your source code should be submitted as well (include it in the above zip file).

Theoretical questions:

1. (Graph search) Consider the graph $G = (V, E)$ shown in Figure 1. Find the shortest path between the denoted start and goal nodes using the following algorithms. In each case, provide a detailed explanation of the algorithm run, and state the order in which states are expanded.
 - (a) Breadth-first search (BFS) algorithm, considering unweighted edges, as in Figure 1(left).
 - (b) Depth-first search (DFS) algorithm, considering unweighted edges, as in Figure 1(left).
 - (c) Best-first search (Dijkstra) algorithm, considering weighted edges, as in Figure 1(right).
 - (d) A* algorithm, considering weighted edges and the heuristic function h , as in Figure 1(right).
 - (e) Is A* more efficient than Dijkstra in the considered setting? Explain your answer.

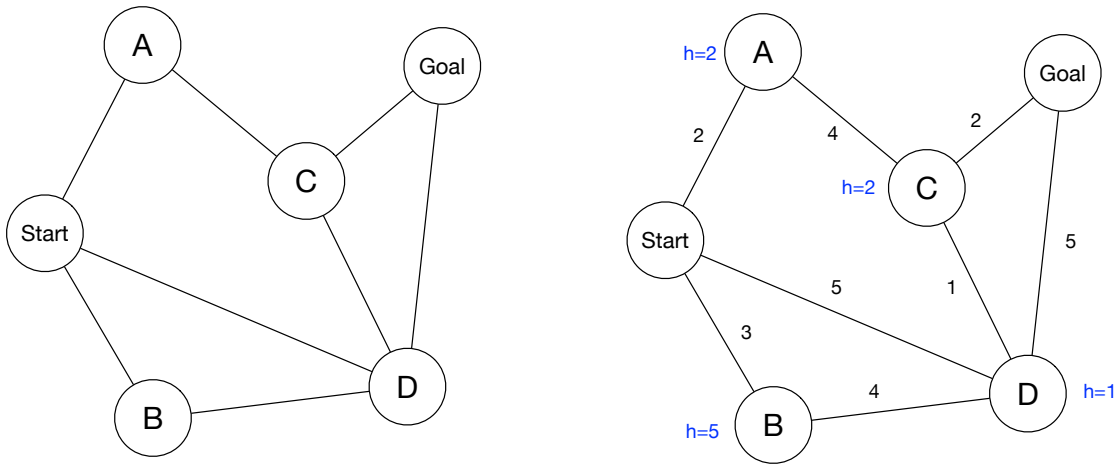


Figure 1

2. (A*) In this question we focus on the properties of heuristic functions. Reminder:
 - A heuristic function is **admissible** if $h(x_i) \leq \text{dist}(x_i, x_\tau)$ for every node x_i with coordinates x_i in graph G , where $\text{dist}(x_i, x_\tau)$ is the shortest distance from x_i to x_τ .
 - A heuristic function is **consistent** if
 - $h(x_\tau) = 0$ for the goal node τ with coordinates x_τ .

- $h(x_i) \leq c(x_i, x_j) + h(x_j)$ for every node i with coordinates x_i and its children j with coordinates x_j .
- A heuristic function is ϵ –**consistent**, with $\epsilon \geq 1$, if
 - $h(x_\tau) = 0$ for the goal node τ with coordinates x_τ .
 - $h(x_i) \leq \epsilon c(x_i, x_j) + h(x_j)$ for every node i with coordinates x_i and its children j with coordinates x_j .
- (a) Prove that if $h^{(1)}$ and $h^{(2)}$ are consistent heuristics, then $h \doteq \max(h^{(1)}, h^{(2)})$ is also consistent.
- (b) Prove that if $h^{(1)}$ and $h^{(2)}$ are consistent heuristics, then $h \doteq h^{(1)} + h^{(2)}$ is ϵ –consistent.
- (c) Provide an example of an admissible heuristic h that is not consistent. Show explicitly why h does not satisfy the consistency property.

Hands-on tasks:

1. (Probabilistic Roadmap) Consider a 2D environment of size $N \times N$ and a known set C_{obs} of scattered obstacles. For simplicity, consider all obstacles are represented by rectangular polygons of size $n_x^{obs} \times n_y^{obs}$ aligned with x and y axes (i.e. not rotated).
 - (a) Implement *on your own* a function `GeneratePRM` which creates a simplified 2D probabilistic roadmap (PRM) $G = (V, E)$. In this simplified version, consider any two (sampled) points v_1 and v_2 in the 2D space can be connected by a straight line if the Euclidean distance $\mathbf{dist}(v_1, v_2)$ between them is less than a distance threshold th_d , and the path from v_1 to v_2 does not intersect with an obstacle (i.e. in C_{free}). Assign the weight of the corresponding edge (v_1, v_2) to be $\mathbf{dist}(v_1, v_2)$. Assume sampling of new candidate positions is done from a uniform distribution
 - Input: distance threshold th_d , number of nodes N_{nodes} , set of obstacles C_{obs} .
 - Output: a PRM $G = (V, E)$.
 - (b) Let $N = 100$. Randomly scatter $n_{obs} = 15$ identical obstacles in the 2D environment, with $n_x^{obs} = 15$ and $n_y^{obs} = 10$. Given this environment (in terms of obstacles), draw on separate plots the obstacles and the corresponding probabilistic maps considering all permutations $N_{nodes} \in \{100, 500\}$ and $th_d \in \{20, 50\}$. For each configuration, indicate in the plot the number of edges and an average node degree in the constructed PRM G .
2. (Graph search) Consider the generated environment and the constructed PRM with $N_{nodes} = 100$ and $th_d = 50$ from clause 1b. Consider start and goal positions to be, respectively, at the bottom left and top right corners of the 2D $N \times N$ considered environment. For simplicity, denote by x_{start} and x_{goal} the closest nodes in G to these positions. Implement *on your own* the Dijkstra algorithm, or/and an A* with Euclidean distance heuristic, and use it to find the shortest path from x_{start} to x_{goal} . Plot the corresponding shortest path on top of the PRM, while indicating the obstacles.