ASEN 5519 - Algorithmic Motion Planning
Fall 2020

Homework 4

Assigned October 23; Due November 6

Exercise 1. Implement an A^* search algorithm that takes weighted directed graph $G = (V, E, W)$, where

- $V = \{v_1, \ldots, v_n\}$ is a set of nodes,
- $E \subseteq V \times V$ is a set of edges (ordered pairs of vertices),
- $W : E \to \mathbb{R}$ is a weight function,

and an initial node $v_{\text{start}} \in V$, a final node $v_{\text{goal}} \in V$, and a heuristic cost to goal function $h : V \to \mathbb{R}$ as input and returns a path $q = q_0 q_1 \ldots q_m$ such that $q_0 = v_{\text{start}}$, $q_m = v_{\text{goal}}$, and minimizes the path length

$$W(q) = \sum_{i=0}^{m-1} W((q_i, q_{i+1})).$$

(a) Solve the example on slide 22 of Lecture 10 using your A^* implementation. Report the path, path length, and number of iterations it took to find this path.

(b) How can you turn your A^* search implementation to Dijkstra's?

(c) Solve the example in part (a) using Dijkstra's algorithm. Report the path, path length, and number of iterations it took to find this path.

(d) Between A^* and Dijkstra's algorithm, which one performs better? Which one would you choose to search a large graph for the shortest path between (i) v_{start} and v_{goal} and (ii) v_{start} and every $v \in V$?

(e) How would you modify your implementation for a graph that is undirected?

Exercise 2. Implement a simple probabilistic roadmap (PRM) planner that samples n configurations (before validity check) and connects each valid sample to every valid configuration that is within radius r from it through a straight line. Assume that the C-space is a rectangle in $\mathbb{R}^2$ with $x \in [x_{\min}, x_{\max}]$ and $y = [y_{\min}, y_{\max}]$. The program should take n, r, obstacles, C-space boundaries, and q_{start} and q_{goal} as input and return the roadmap, a path from q_{start} to q_{goal}, the path length, and the computation time.

(a) Solve the planning problem in **Exercise 2.(a)** of **Homework 3** with boundaries $x \in [-1, 11]$ and $y = [-3, 3]$.

i. Plot the roadmap and the solution path for $n = 200$ and $r = 1$. Indicate the path length in the title of the plot.

ii. Vary n and r and benchmark your solutions in three categories of number of valid solutions, path length, and computation time. For benchmarking, use 100 runs for each

$$(n, r) \in \{(200, 0.5), (200, 1), (200, 1.5), (200, 2), (500, 0.5), (500, 1), (500, 1.5), (500, 2)\}.$$

Show your results using boxplots.

iii. Based on your empirical evaluations, what are the optimal values for n and r? Justify your answer.

iv. Augment your PRM planner with a path smoothing technique and re-evaluate your benchmarks. What are the optimal values for n and r with path smoothing? Justify your answer.

(b) Solve the planning problems in **Exercise 7** of **Homework 1** using your PRM (no path smoothing). For W_1, the boundaries are $x, y \in [-1, 13]$. For W_2, the boundaries are $x \in [-6, 36]$ and $y \in [-6, 6]$. For each W_1 and W_2, perform the following steps:

i. Plot the roadmap and the solution path for $n = 200$ and $r = 2$. Indicate the path length in the title of the plots.

ii. Vary n and r and benchmark your solutions in three categories of number of valid solutions, path length, and computation time. For benchmarking, use 100 runs for each

$$(n, r) \in \{(200, 1), (200, 2), (500, 1), (500, 2), (1000, 1), (1000, 2)\}.$$

Show your results using boxplots.

iii. Based on your empirical evaluations, what are the optimal values for n and r in W_1 and W_2? Justify your answer.

iv. Enable path smoothing option in your PRM and re-evaluate your benchmarks. What are the optimal values for n and r with path smoothing? Justify your answer.

(c) Does your PRM implementation need to change in order for it to solve the planning problem in **Exercise 4** of **Homework 3**? Justify your answer.

Exercise 3. Implement the basic GoalBiasRRT planner with step size r and goal bias probability of p_{goal}. Assume that the C-space is a rectangle in $\mathbb{R}^2$ with $x \in [x_{\min}, x_{\max}]$ and $y = [y_{\min}, y_{\max}]$. The program should take r, p_{goal}, maximum number of iterations n, obstacles, C-space boundaries, q_{start}, q_{goal}, and radius ϵ (centered at q_{goal}) for the termination condition at goal as input and return a path from q_{start} to q_{goal}, the path length, and the computation time.

Solve the planning problems in **Exercise 2.(a)-(b)** using your basic RRT implementation with $n = 5000$, $r = 0.5$, $p_{\text{goal}} = 0.05$, and $\epsilon = 0.25$.

(a) For each environment, plot one solution path and its corresponding tree. Indicate the path length in the title of the plot.

(b) For each environment, use 100 runs to benchmark your implementation in three categories of number of valid solutions, path length, and computation time. Show your results using boxplots.

(c) Does your RRT implementation need to change in order for it to solve the planning problem in **Exercise 4** of **Homework 3**? Justify your answer.

Exercise 4. You have implemented four planners (gradient descent with a potential function, wavefront, PRM, and RRT) to solve planning problems in **Exercise 2.(a)-(c)**. Reflect on the performance of these planners and provide a discussion on their advantages and disadvantages.