

# Problem Set 1

Teera Tesharojanasup

Northeastern University, Boston

July 10th, 2024

## Overview

Problem set 1 for CS 4100 Summer II. Taught by assistant teaching professor, [Rajagopal Venkat](#). [1]

## 1 Categorizing AI Environments

- Q1 Under what (implementation-related) assumptions is maze-solving a fully observable environment? What are the agent's percepts?
- Q2 Under what (implementation-related) assumptions is maze-solving a partially observable environment? What are the agent's percepts?
- Q3 Is a known environment always fully observable? Explain with an example.
- Q4 Is a partially observable environment always an unknown environment? Explain with an example.
- Q5 What is the difference between an unknown environment and a non-deterministic environment? Explain with an example.

## 2 Search

In this problem set, we will be using Rook Jumping Mazes (RJMs) - a game based on Rook moves in Chess - to understand various search techniques. In a RJM, you are given a square  $n \times n$  grid, with each cell containing a number between 1 to  $n - 1$ . From any cell, the player may move either horizontally or vertically but must take exactly the number of steps as shown on the player's current cell. Regardless of the length of the jump (i.e. number of cells passed), this action is considered a single jump (i.e., having a cost of 1) for purposes of our search. The objective is to get from an assigned start state to a pre-specified goal state. Consider this example:

	1	2	3	4	5
A	4	G	3	3	2
B	3	3	4	4	4
C	1	1	3	4	4
D	3	3	3	1	2
E	3	1	1	3	3

The cell D1 is the start state and the cell A2 is the goal state. The shortest solution to this RJM is: [D1, A1, A5, C5, C1, C2, D2, A2]. Such a game lends itself well to the various search algorithms we covered in class.

Q6 Can both Breadth-First Search (BFS) and Depth-First Search (DFS) be used to solve RJMs? Will both yield the shortest path?

Q7 Assuming that each jump has a cost of 1, irrespective of the number of cells the rook passes while making that jump, we can use A\* search to find the shortest path. Recall from class that A\* search requires a consistent heuristic to guarantee an optimal solution. Show that for this problem, the Manhattan distance from any cell to the goal cell divided by (n - 1) is a consistent heuristic, where n is the number of cells in a row/column. In other words, show that

$$H(\text{node}, \text{goal}, n) = \frac{|\text{node}_x - \text{goal}_x| + |\text{node}_y - \text{goal}_y|}{n - 1}$$

is a consistent heuristic. Remember that a proof may not rely on a single example, but instead must hold true in general.

Q8 Consider the following RJM:

	1	2	3	4
A	2	3	2	3
B	1	2	1	1
C	3	2	2	3
D	1	1	3	G

On this maze, using the heuristic from Q7, complete the A\* search process shown below, starting with Step 2. Terminate your search when the Goal node (cell D4) is popped from the priority queue. Double check your calculations!

### 3 Local Search

Your professor is a coffee snob, and since he knows a thing or two about AI, he tries to build an espresso machine that will learn a user's preferences over time to pull the perfect shot. He lets the machine control the following values:

- Weight of coffee beans (15-21 grams)
- Grind size (250 - 400 microns)
- Water Pressure (7 - 11 bars)
- Brew time (27 - 33 seconds)

Q9 Based on our discussions in class, is this problem well-suited to local search algorithms? Explain your reasoning.

Q10 How you would implement such a program? What would be a reasonable objective function? (This question is somewhat open-ended, so don't worry too much about the 'right' answer. I am really looking for whether your thought process reflects a solid understanding of how local search works.)

Q11 Which of the optimizations and local search variants discussed in class (random restarts, varying step size, simulated annealing and genetic algorithms) are applicable to this problem, and why?

## 4 Adversarial Search

- Q12 Construct a game tree example with a depth of 4 and a branching factor of 2, that illustrates the best-case for alpha-beta pruning, except for the trivial solution. Hand drawn figures accepted for this question, but please ensure clarity. (Here's some inspiration: [CS151, 2005](#). [Jim Marshall, SLC](#))
- Q13 Why is it desirable to combine iterative deepening with alpha-beta pruning from a practical standpoint?

## 5 Academic Integrity

- Q14 Review, and copy/paste the academic integrity acknowledgement in your final submission as the answer to Q14.

I have read and understood the academic integrity policy as outlined in the course syllabus for CS4100. By pasting this acknowledgement in my submission, I declare that all work presented here is my own, and any conceptual discussions I may have had with classmates have been fully disclosed. I declare that generative AI was not used to answer any questions in this assignment. Any use of generative AI to improve writing clarity alone is accompanied by an appendix with my original, unedited answers.

## References

- [1] R. Venkatesaramani, "Personal website." <https://rajagopalvenkat.com/>. Accessed: 2024-07-10.