University of Alberta
Computing Science Department
Search & Planning in AI - CMPUT 366

Midterm
15 Marks
February 17, 2023

Name: Student ID:

Instructions: The exam starts at 11.00 am and it finishes at 11.50 am. Use the space below each question to write your answer to the questions. You can write your answer using a pen or a pencil, whichever you prefer. Please leave your ID on your desk so we can check it as you sign the attendance sheet for the exam.

1. (3 Marks) In the Missionary & Cannibals problem one needs to define how to transport 3 missionaries and 3 cannibals across a river on a boat that carries at most 2 people. Naturally, there must be at least one person on the boat so it can cross the river and, once the boat crosses the river from bank A to bank B, people on bank A cannot use the boat until it is brought back to A. In this problem there can't be more cannibals than missionaries in one of the banks at a given time.

   Define the Missionary & Cannibals problem as a search problem. You should assume that states of the problem where there are more cannibals than missionaries in one of the banks have no children (i.e., no actions can be applied from those states). You should also assume that the action of crossing the river costs 1.0, independently on the number of people on the boat.

   This question can be answered in many different ways. Here is one way of answering it:

   - States are defined by the tuple (M_l, C_l, M_r, C_r, s), where M_l and C_l are the number of missionaries and cannibals on the left bank of the river. M_r and C_r are defined similarly for the other bank; s is the bank where the boat is currently located.
   - Initial state: $(0, 0, 3, 3, r)$.
   - Goal state: $(3, 3, 0, 0, l)$. Naturally, we could swap initial and goal and the answer still be correct.
   - Actions: Subtract $1 \leq n \leq 2$ from M_s and/or C_s, where s is the side of the boat, and add n to M and/or C of the other bank; change s to the other bank-value.
   - Cost function: 1.0 for any action.

2. (3 Marks) Consider the search problem shown in the 5×7 grid below where an agent is allowed to move in all eight directions (up, down, left, right, and diagonally) and walls are shown in black. The actions for moving up, down, left, and right cost 1.0, while the diagonal moves cost 1.5. The initial state s is in cell $(5, 1)$ and the goal s_g is in cell $(5, 7)$. The numbers in the cells show the heuristic value for each state; the heuristic is consistent. The agent cannot "cut corners", i.e., it cannot move diagonally if there is a wall in between the cells. For example, the agent cannot move from cell $(2, 5)$ to cell $(1, 6)$ with a diagonal move. Answer the following questions regarding this search problem and heuristic function.

	1	2	3	4	5	6	7
1	8.0	7.0	6.0	5.5	5.0	4.5	4.0
2	7.5	6.5	5.5		4.0		3.0
3	7.0	6.0	5.0		3.0		2.0
4	6.5	5.5	4.5		2.5		1.0
5	s	5.0	4.0	3.0	2.0		s_g

a) (1.5 Marks) Draw a line in the grid above denoting the solution path A* encounters for this problem (write "A*" next to the line). The line should go through the optimal path: (5, 1) (4, 2) (3, 3) (2, 3) (1, 3) (1, 4) (1, 5) (1, 6) (1, 7) (2, 7) (3, 7) (4, 7), (5, 7)

b) (1.5 Marks) Draw a line in the grid above denoting the solution path WA* encounters for this problem; you should assume that the w-value is so large that the f-values used to guide the search are dominated by the h-value (write "WA*" next to the line). (5, 1) (5, 2) (5, 3) (5, 4) (5, 5) (4, 5) (3, 5) (2, 5) (1, 5) (1, 6) (1, 7) (2, 7) (3, 7) (4, 7), (5, 7)

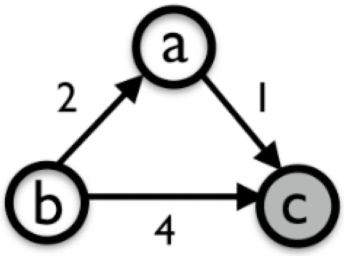
Hint: There is no need to actually do the computation of the steps of the algorithms. You should be able to answer this question just by observing the h-values of the problem.

3. (3 Marks) One of your friends is trying to solve (with no success!) a pathfinding problem with A*. While chatting with them you discover that they have implemented a good heuristic function and that the search space has a reasonable number of transpositions and that is why they chose to use A*. You also discover that the problem has no real-time constraints such as video games and GPS and it is fine if the search took weeks of computation to solve the problem. Finally, you friend mentioned that A* runs out of memory after a few hours of computation. Based on what your friend reported to you, suggest an alternative search algorithm they might want to attempt implementing for solving the problem. Explain to your friend why this algorithm is a good alternative to A* for this particular problem.

I would suggest IDA* with a transposition table. IDA* has linear memory requirement and could run for much longer than A* (essentially forever if the solution depth isn't too large). The transposition table would avoid some of the transpositions in the search tree and IDA* would be able to use the heuristic function that my friend already designed to the problem.

4. (3 Marks) A classmate is attempting to solve a search problem with A* and their implementation is occasionally failing to return optimal solutions. Your classmate explains that they are using two admissible and consistent heuristic functions to guide the search. They decided to use two heuristic functions because they have two great functions and couldn't decide which one to use. Once a node n is

generated, they randomly choose one of the two heuristics to evaluate n; the f-value of n is computed with the randomly chosen heuristic. Noting that you looked confused, your classmate drew a small example on a piece of paper to illustrate their approach. The example they drew is shown below.



In the example, b is the initial state and c the goal state. The first heuristic function, h_1, assigns the following values to the states: $h_1(a) = 0, h_1(b) = 1, h_1(c) = 0$; the second heuristic, h_2, assigns the following: $h_2(a) = 1, h_2(b) = 3, h_2(c) = 0$. During search, when b is generated, h_2 could be randomly selected to evaluate b; once b is expanded, h_1 could be randomly chosen to evaluate a. Answer the following question while considering your classmate's description of the issue they are facing and the small example they gave you.

a) (1 Mark) What is the property that the heuristic function used to guide the search (the one that randomly chooses h_1 or h_2 to evaluate a node) does not have that both h_1 and h_2 have?

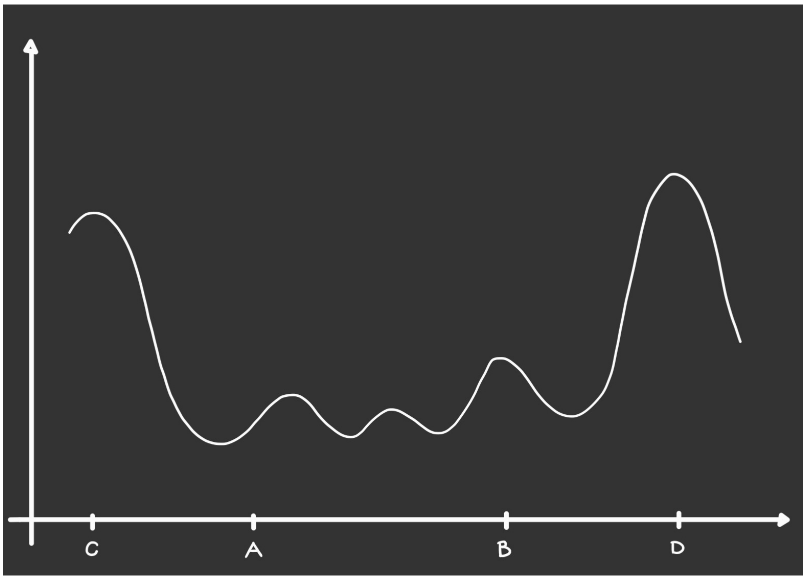
   Consistency

b) (2 Marks) Given that your classmate doesn't want to change the heuristic function they to guide the search, what is possibly wrong with their implementation? Why do you suspect this is the problem?

Their implementation is possibly missing the re-opening of nodes. I suspect that this is the problem because the resulting heuristic function is inconsistent and not re-opening nodes could lead to sub-optimal solutions even with an admissible heuristic function.

5. (3 Marks) A colleague at work is using Hill Climbing (HC) for solving a combinatorial search problem that requires one to maximize the value of an objective function. Your colleague has an idea of how good the solution should be in terms of its objective value. Unfortunately, HC is returning candidates whose values are far from the values that are considered good. Your colleague hypothesized that the topology of the search space looks like the plot below, where the y-axis shows the objective value and the x-axis different candidates.

   Your colleague further hypothesized that their HC implementation is likely generating initial candidates that always fall in the region between A and B, while the desired solutions are around points C and D. They concluded that it is hard to sample directly around C and D and that the initial candidates of search will inevitably be from the region between A and B.



Suggest a different search algorithm that could be used to obtain candidate solutions with better objective values. Justify your answer.

Possible correct answers: Stochastic hill climbing, Beam search, Random Walks, and Simulated Annealing. All these algorithms are able to escape local optima.

Incorrect answer: Random restarts alone is incorrect because HC would always fall in the A-B region.