College of Engineering & Applied Sciences

CSPB 3202

Introduction To Artifical Intelligence

Assignment 2 - Informed Search

UNIVERSITY OF COLORADO

2024

Introduction To Artifical Intelligence - Assignment 2 - Informed Search

Assignment 2 - Informed Search

Assignment 2 - Informed Search

I have neither given nor received unauthorized assistance.

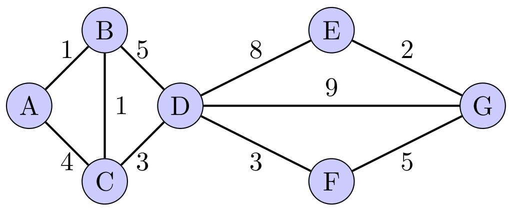
Taylor Larrechea

The original assignment can be found here.

Problem 1

Problem Statement

Consider the state space graph shown below. A is the start state and G is the goal state. The costs for each edge are shown on the graph. Each edge can be traversed in both directions. Note that the heuristic h_1 is consistent but the heuristic h_2 is not consistent.



Node	h_1	h_2
A	9.5	10
B	9	12
C	8	10
D	7	8
E	1.5	1
F	4	4.5
G	0	0

(a) Possible Paths Returned

For each of the following graph search strategies (*do not answer for tree search*), mark which, if any, of the listed paths it could return. Note that for some search strategies the specific path returned might depend on tie-breaking behavior. In any such cases, make sure to mark *all* paths that could be returned under some tie-breaking scheme.

Search Algorithm	**A-B-D-G**	**A-C-D-G**	**A-B-C-D-F-G**
Depth First Search			
Breadth First Search			
Uniform Cost Search			
A^* Search With Heuristic h_1			
A^* Search With Heuristic h_2			

(b) Heuristic Function Properties

Suppose you are completing the new heuristic function h_3 shown below. All the values are fixed except $h_3(B)$.

Node	A	B	C	D	E	F	G
h_3	10	?	9	7	1.5	4.5	0

For each of the following conditions, write the set of values that are possible for $h_3(B)$. For example, to denote all non-negative numbers, write $[0, \infty]$, to denote the empty set, write $\varnothing$, and so on.

(i) What values of $h_3(B)$ make h_3 admissible?

(ii) What values of $h_3(B)$ make h_3 consistent?

(iii) What values of $h_3(B)$ will cause A^* graph search to expand node A, then node C, then node B, then node D in order?

Part A - Solution

Depth First Search

All of the paths that are listed in this table are able to be explored using Depth First Search. This is because the graph that we are working with does not have any directions. Because this graph is undirected, the paths that are listed in the table are absolutely possible. For Depth First Search we can then say:

Search Algorithm	**A-B-D-G**	**A-C-D-G**	**A-B-C-D-F-G**
Depth First Search	Yes	Yes	Yes

Breadth First Search

The only path for Breadth First Search that is not possible to be traversed is **A-B-C-D-F-G**. This is because of the nature of Breadth First Search. Starting from A, BFS will then visit all children of A first. After the children are visited, the next level of children are visited after that. In the path **A-B-C-D-F-G**, if we first visited B, we would essentially be revisiting C for this path to be valid. And this is not a valid search done by BFS since we cannot revisit children. For Breadth First Search we can then say:

Search Algorithm	**A-B-D-G**	**A-C-D-G**	**A-B-C-D-F-G**
Breadth First Search	Yes	Yes	No

Uniform Cost Search

Uniform Cost Search is always going to return the path that has the least cumulative cost out of all possible paths. The first path, **A-B-D-G** is going to have a cumulative cost of 15, **A-C-D-G** is going to have a cumulative cost of 16, and the final path is going to have a cumulative cost of **A-B-C-D-F-G** will have a cumulative cost of 13. This means the only path that can be returned using UCS is going to be **A-B-C-D-F-G**. For Uniform Cost Search we can then say:

Search Algorithm	**A-B-D-G**	**A-C-D-G**	**A-B-C-D-F-G**
Uniform Cost Search	No	No	Yes

A* Search With Heuristic h_1

For each heuristic, both h_1 and h_2, we essentially need to calculate

$$f(\alpha) = g(\alpha) + h(\alpha)$$

where α is a node in the graph. Conversely, $g(\alpha)$ is the cost to arrive at a specific node. A* Search is going to prioritize the path that has the lowest f cost. Keeping this in mind, the path that has the lowest f cost of all three paths is **A-B-C-D-F-G** with $f(G) = 13$. Conversely **A-B-D-G** is $f(G) = 15$ and **A-C-D-G** is $f(G) = 16$. For A* Search with heuristic h_1 we can say:

Search Algorithm	**A-B-D-G**	**A-C-D-G**	**A-B-C-D-F-G**
A* Search With Heuristic h_1	No	No	Yes

A* Search With Heuristic h_2

Similar to heuristic h_1, we need to find the path with the smallest f value. For the path **A-B-D-G** we have $f(G) = 15$, **A-C-D-G** is $f(G) = 16$, and lastly **A-B-C-D-F-G** has an f value of $f(G) = 13$. A* will once again choose the path with the smallest f value.

Search Algorithm	**A-B-D-G**	**A-C-D-G**	**A-B-C-D-F-G**
A* Search With Heuristic h_2	No	No	Yes

Part B - Solution

(i) To determine what values of $h_3(B)$ make h_3 *admissible*, we need to not overestimate the cost using

$$\boxed{f(\alpha) = g(\alpha) + h(\alpha)}$$

where $g(\alpha)$ is the cost to get to a current node and $h(\alpha)$ is the estimated cost to get from the current node to the goal. We essentially need to find the most minimal cost for going from B to G out of all possible paths. The possible costs from B to G are:

- **B-D-G**: Cost is 14
- **B-D-E-G**: Cost is 15
- **B-D-F-G**: Cost is 13
- **B-C-D-G**: Cost is 13
- **B-C-D-E-G**: Cost is 14
- **B-C-D-F-G**: Cost is 12

Therefore, we must take the minimum of these possible costs. This means the *admissible* values for $h_3(B)$ are

$$\boxed{h_3(B) = (0, 12].}$$

(ii) To determine what values of $h_3(B)$ make $h_3(B)$ *consistent*, we need to take the minimal cost in terms of heuristic values for expanding from node B. The value that we are essentially calculating is

$$\boxed{h_3(n_1) - h_3(n_2) \leq g(n_1, n_2)}$$

The neighbors of B are A,C, and D. Going through these calculations for each B's neighbors we find

- $h_3(B) \leq g(B, A) + h_3(A) \leq 1 + 10 \leq 11 \qquad h_3(B) \geq h_3(A) - g(A, B) \geq 10 - 1 \geq 9$
- $h_3(B) \leq g(B, C) + h_3(C) \leq 1 + 9 \leq 10 \qquad h_3(B) \geq h_3(C) - g(C, B) \geq 9 - 1 \geq 8$
- $h_3(B) \leq g(B, D) + h_3(D) \leq 5 + 7 \leq 12 \qquad h_3(B) \geq h_3(D) - g(D, B) \geq 7 - 5 \geq 2$

This means for h_3 to be *consistent*, we must restrict its values to be

$$\boxed{h_3(B) = [9, 10].}$$

(iii) To determine the value of $h_3(B)$ that will generate the path **A-C-B-D** ... we first have to ensure that we expand C first from A. This is only possible if the corresponding f value that is produced when expanding to C is less than that of B. At this point in time, if we first expand to C then

$$f(C) = g(A, C) + h_3(C) = 4 + 9 = 13.$$

This in turn means that

$$f(B) > f(C) > g(A, B) + h_3(B) > 1 + h_3(B) > 13 \quad \rightarrow \quad h_3(B) > 12.$$

To now travel from C to B we constitute $f(B) < f(D)$. At this point in time we then have in our path

$$f(D) = g(C, D) + h_3(D) = 7 + 7 = 14.$$

So, now applying $f(B) < f(D)$ this means

$$f(B) < f(D) < g(C, B) + h_3(B) < 5 + h_3(B) < 14 \quad \rightarrow \quad h_3(B) < 9.$$

At this point we have a contradiction; $h_3(B)$ must be greater than 12 and less than 9 for this path to happen. **This is not possible**.