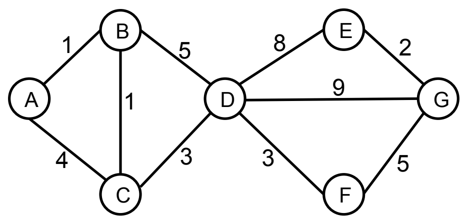
CS188: Exam Practice Session 1 Solutions

Q1. Heuristics (Fall 2013)



Consider the state space graph shown above. 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.

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

State	A	B	C	D	E	F	G
h	10	?	9	7	1.5	4.5	0

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

(a) What values of $h(B)$ make h admissible?

To make h admissible, $h(B)$ has to be less than or equal to the actual optimal cost from B to goal G, which is the cost of path B-C-D-F-G, i.e. 12. The answer is $0 \leq h(B) \leq 12$

(b) What values of $h(B)$ make h consistent?

All the other nodes except node B satisfy the consistency conditions. The consistency conditions that do involve the state B are:

$$h(A) \leq c(A,B) + h(B) \qquad h(B) \leq c(B,A) + h(A)$$
$$h(C) \leq c(C,B) + h(B) \qquad h(B) \leq c(B,C) + h(C)$$
$$h(D) \leq c(D,B) + h(B) \qquad h(B) \leq c(B,D) + h(D)$$

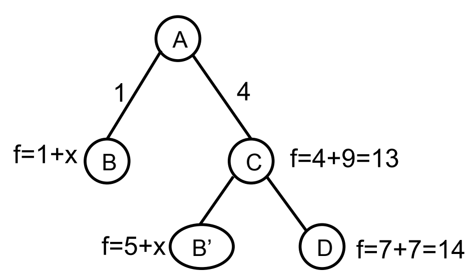
Filling in the numbers shows this results in the condition: $9 \leq h(B) \leq 10$

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

The A* search tree using heuristic h is on the right. In order to make A* graph search expand node A, then node C, then node B, suppose $h(B) = x$, we need

$$1 + x > 13$$
$$5 + x < 14 \quad (expand\ B') \quad or \quad 1 + x < 14 \quad (expand\ B)$$

so we can get $12 < h(B) < 13$



Q2. All Searches Lead to the Same Destination (Spring 2014)

For all the questions below assume :

- All search algorithms are *graph* search (as opposed to tree search).
- $c_{ij} > 0$ is the cost to go from node i to node j.
- There is only one goal state (as opposed to a set of goal states).
- All ties are broken alphabetically.
- Assume heuristics are consistent.

Definition: Two search algorithms are defined to be ***equivalent*** if and only if they expand the same states in the same order and return the same path.

In this question we study what happens if we run uniform cost search with action costs d_{ij} that are potentially different from the search problem's actual action costs c_{ij}. Concretely, we will study how this might, or might not, result in running uniform cost search (with these new choices of action costs) being equivalent to another search algorithm.

(a) Mark *all* choices for costs d_{ij} that make running **Uniform Cost Search** algorithm with these costs d_{ij} *equivalent* to running **Breadth-First Search**.

- ◯ $d_{ij} = 0$
- ● $d_{ij} = \alpha$, $\alpha > 0$
- ◯ $d_{ij} = \alpha$, $\alpha < 0$
- ● $d_{ij} = 1$
- ◯ $d_{ij} = -1$
- ◯ None of the above

Breadth-First Search expands the node at the shallowest depth first. Assigning a constant positive weight to all edges allows to weigh the nodes by their depth in the search tree.

(b) Mark *all* choices for costs d_{ij} that make running **Uniform Cost Search** algorithm with these costs d_{ij} *equivalent* to running **Depth-First Search**.

- ◯ $d_{ij} = 0$
- ◯ $d_{ij} = \alpha$, $\alpha > 0$
- ● $d_{ij} = \alpha$, $\alpha < 0$
- ◯ $d_{ij} = 1$
- ● $d_{ij} = -1$
- ◯ None of the above

Depth-First Search expands the nodes which were most recently added to the fringe first. Assigning a constant negative weight to all edges essentially allows to reduce the value of the most recently nodes by that constant, making them the nodes with the minimum value in the fringe when using uniform cost search.

(c) Mark *all* choices for costs d_{ij} that make running **Uniform Cost Search** algorithm with these costs d_{ij} *equivalent* to running **Uniform Cost Search** with the original costs c_{ij}.

- ○ $d_{ij} = c_{ij}^2$
- ○ $d_{ij} = 1/c_{ij}$
- ● $d_{ij} = \alpha\ c_{ij}, \quad \alpha > 0$
- ○ $d_{ij} = c_{ij} + \alpha, \quad \alpha > 0$
- ○ $d_{ij} = \alpha\ c_{ij} + \beta, \quad \alpha > 0,\ \beta > 0$
- ○ None of the above

Uniform Cost Search expands the node with the lowest cost-so-far $= \sum_{ij} c_{ij}$ on the fringe. Hence, the relative ordering between two nodes is determined by the value of $\sum_{ij} c_{ij}$ for a given node. Amongst the above given choices, only for $d_{ij} = \alpha\ c_{ij},\ \alpha > 0$, can we conclude,
$\sum_{ij \in \text{ path}(n)} d_{ij} \geq \sum_{ij \in \text{ path}(m)} d_{ij} \iff \sum_{ij \in \text{ path}(n)} c_{ij} \geq \sum_{ij \in \text{ path}(m)} c_{ij}$, for some nodes n and m.

(d) Let $h(n)$ be the value of the heuristic function at node n.

(i) Mark *all* choices for costs d_{ij} that make running **Uniform Cost Search** algorithm with these costs d_{ij} *equivalent* to running **Greedy Search** with the original costs c_{ij} and heuristic function h.

- ○ $d_{ij} = h(i) - h(j)$
- ● $d_{ij} = h(j) - h(i)$
- ○ $d_{ij} = \alpha\ h(i), \quad \alpha > 0$
- ○ $d_{ij} = \alpha\ h(j), \quad \alpha > 0$
- ○ $d_{ij} = c_{ij} + h(j) + h(i)$
- ○ None of the above

Greedy Search expands the node with the lowest heuristic function value $h(n)$. If $d_{ij} = h(j) - h(i)$, then the cost of a node n on the fringe when running uniform-cost search will be $\sum_{ij} d_{ij} = h(n) - h(start)$. As $h(start)$ is a common constant subtracted from the cost of all nodes on the fringe, the relative ordering of the nodes on the fringe is still determined by $h(n)$, i.e. their heuristic values.

(ii) Mark *all* choices for costs d_{ij} that make running **Uniform Cost Search** algorithm with these costs d_{ij} *equivalent* to running **A* Search** with the original costs c_{ij} and heuristic function h.

- ○ $d_{ij} = \alpha\ h(i), \quad \alpha > 0$
- ○ $d_{ij} = \alpha\ h(j), \quad \alpha > 0$
- ○ $d_{ij} = c_{ij} + h(i)$
- ○ $d_{ij} = c_{ij} + h(j)$
- ○ $d_{ij} = c_{ij} + h(i) - h(j)$
- ● $d_{ij} = c_{ij} + h(j) - h(i)$
- ○ None of the above

A^* search expands the node with the lowest $f(n) + h(n)$ value, where $f(n) = \sum_{ij} c_{ij}$ is the cost-so-far and h is the heuristic value. If $d_{ij} = c_{ij} + h(j) - h(i)$, then the cost of a node n on the fringe when running uniform-cost search will be $\sum_{ij} d_{ij} = \sum_{ij} c_{ij} + h(n) - h(start) = f(n) + h(n) - h(start)$. As $h(start)$ is a common constant subtracted from the cost of all nodes on the fringe, the relative ordering of the nodes on the fringe is still determined by $f(n) + h(n)$.