CS 188 Summer 2021 Introduction to Artificial Intelligence — Exam Prep 1B Solutions

Course Overview

Here are some questions for you:

1. What is AI?
2. What can AI do?
3. What do you want to learn from this course?

There are two types of discussion sections:

1. Regular Discussion
2. Exam Prep
3. LOST

There are 5 graded components:

1. Programming Assignments (25%)
2. Electronic Homework Assignments (10%)
3. Written Homework Assignments (10%)
4. Midterm (20%)
5. Check-in quizzes + Final exam (35%)

Check-in Quizzes boost: Your percentage grade on the final = max(percentage grade on the final, 1/4 * percentage grade on check-in quizzes + 3/4 * percentage grade on the final) For example, if you completed 80% of all check-in quizzes and got 60% on the final, your grade on the final will be 1/4* 80% + 3/4*60% = 65%.

Q1. Search

For this problem, assume that all of our search algorithms use tree search, unless specified otherwise.

(a) For each algorithm below, indicate whether the path returned after the modification to the search tree is guaranteed to be identical to the unmodified algorithm. Assume all edge weights are non-negative before modifications.

(i) Adding additional cost $c > 0$ to every edge weight.

	Yes	No
BFS	●	○
DFS	●	○
UCS	○	●

(ii) Multiplying a constant $w > 0$ to every edge weight.

	Yes	No
BFS	●	○
DFS	●	○
UCS	●	○

(b) **For part (b)**, 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. **Assume all graphs are directed and acyclic.**

(i) Assume we have access to costs c_{ij} that make running UCS algorithm with these costs c_{ij} equivalent to running BFS. How can we construct new costs c'_{ij} such that running UCS with these costs is equivalent to running DFS?

○ $c'_{ij} = 0$ ○ $c'_{ij} = 1$ ○ $c'_{ij} = c_{ij}$

● $c'_{ij} = -c_{ij}$ ○ $c'_{ij} = c_{ij} + \alpha$ ○ Not possible

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. 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. Hence, we can construct new costs c'_{ij} by flipping the sign of the original costs c_{ij}.

(ii) Given edge weight $c_{ij} = h(j) - h(i)$, where $h(n)$ is the value of the heuristic function at node n, running UCS on this graph is equivalent to running which of the following algorithm on the same graph?

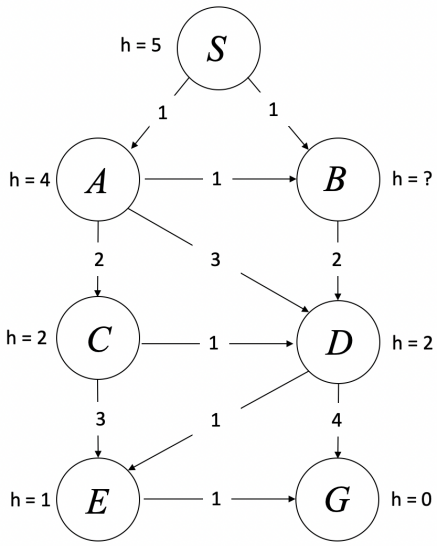
○ DFS ○ BFS ○ Iterative Deepening

● Greedy ● A* ○ None of the above.

Greedy Search expands the node with the lowest heuristic function value $h(n)$. If $c_{ij} = h(j) - h(i)$, then the cost of a node n on the fringe when running uniform-cost search will be $\sum_{ij} c_{ij} = h(1) - h(start) + h(2) - h(1) + ... + h(n) - h(n-1) = h(n) - h(\text{start})$. As $h(\text{start})$ 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.

A* expands the node with the lowest $f(n) = g(n) + h(n)$ value. Since $g(n) = h(n) - h(\text{start})$, $f(n) = 2h(n) - h(\text{start})$.

(c) Consider the following graph. $h(n)$ denotes the heuristic function evaluated at node n.



(i) Given that G is the goal node, and heuristic values are fixed for all nodes other than B, for which values of $h(B)$ will A* tree search be guaranteed to return the optimal path? Fill in the lower and upper bounds or select "impossible."

__0__ $\leq h(B) \leq$ __4__ ○ Impossible

An admissible heuristic is sufficient for A* tree search to be optimal. Recall the constraint for an admissible heuristic: $\forall n, 0 \leq h(n) \leq h^*(n)$. Since all other nodes satisfy the constraint above, we just need to enforce the constraint on node B, so that $0 \leq h(B) \leq h^*(B) = 4$.

(ii) With the heuristic values fixed for all nodes other than B, for which values of $h(B)$ will A* graph search be guaranteed to return the optimal path? Either fill in the lower and upper bound or select "impossible."

__4__ $\leq h(B) \leq$ __4__ ○ Impossible

We need a consistent heuristic for A* tree search to be optimal. Recall the constraint for a consistent heuristic: $\forall A, C \quad h(A) - h(C) \leq \text{cost}(A, C)$. Since all other nodes satisfy the constraint above, we just need to enforce the constraint on node B. Enforcing consistency along $S - B$, gives $h(B) \geq 4$, $A - B$ gives $h(B) \geq 3$, $B - D$ gives $h(B) \leq 4$.