50.021 Artificial Intelligence

Theory Homework 6

Due: every Monday, 4PM before class starts

[Q1]. Refer to figure 1 for this question. S indicates the start node and G indicates the goal node. The numbers indicate the path cost.

1. Run UCS with graph search. What is your final solution? Show your steps.
   Solution:

$$[SA(1), SG(12)]$$
$$[SAC(2), SAB(4), SG(12)]$$
$$[SACD(3), SACG(4), SAB(4), SG(12)]$$
$$[SACG(4), SAB(4), SACDG(5), SG(12)]$$

   Solution is $SACG$.

2. How would you modify the graph in figure 1 such that the solution we obtain if we run BFS with graph search, we have the same solution as part (1)? Draw the modified graph. Prove your answer by running BFS on the modified graph from S. Show your steps.
   Solution:
   We can remove edge SG. Running BFS will result in,

$$[SA]$$
$$[SAB, SAC]$$
$$[SAC, SABD]$$
$$[SABD, SACG]$$

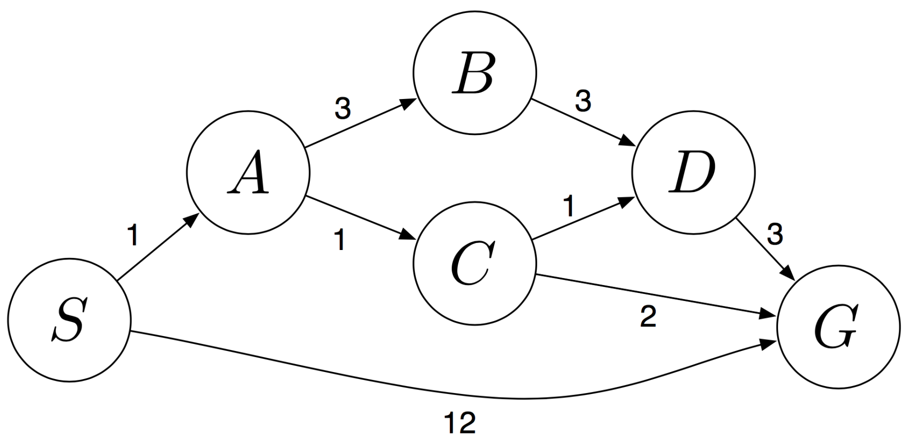

Figure 1: A graph

[Q2]. Suppose we have an arbitrary directed graph, with start node S and goal node G, and each node has at most b children. The path cost from any node i to any other node j is always greater than 1, except from the *outgoing* edges from the start node S to *all* of its children (b many nodes). These outgoing edges from S to all of its b children have negative path cost. In this case,

1. Will UCS with graph search still be able to find the optimum (least) cost from S to G? Explain your answer in not more than 5 sentences.
   Solution: Yes. Assume that node S can go to node G at current minimum cost c in the queue and suppose there's a shorter path through node B. However since all the other edges are positive, the cost to node B to G is > 0 and S to B is larger than c. If negative paths are only at the outgoing edges from S, then it will not violate the above optimality property.

2. Will UCS with plain tree search (without cycle pruning) still be able to find the optimum (least) cost from S to G? Explain your answer in not more than 5 sentences.
   Solution: No, because if there's incoming edges to S, it will cause a negative cycle and UCS will not terminate.

[Q3]. For the following questions, assume these properties:

- All algorithms are graph search, applied to undirected graph with alphabetical nodes, similar to the nodes at Figure 1.
- $c_{i,j} > 0$ denotes the raw path cost to go from node i to node j
- $d_{i,j}$ denotes the *action* cost to go from node i to node j. In other words, $d_{i,j} = f(c_{i,j})$ where f is some function. This is the cost incurred in the search algorithms. In the examples in class, our heuristic function is mostly just a unity function and therefore $d_{i,j} = c_{i,j}$.
- There is only one goal node G.
- Children nodes are explored alphabetically.

Definition of equivalence: Two search algorithms are defined to be **equivalent** if and only if they expand the same nodes in the same exact order and return the same solution.

1. Mark all the choices for costs $d_{i,j}$, that make running BFS algorithm on arbitrary graph with these costs $d_{i,j}$ *equivalent* to running UCS. For each choice you choose, state in your reason in not more than 2 sentences.
   (a) $d_{i,j} = 0$
   (b) $d_{i,j} = \alpha,\ \alpha > 0$ **Solution:** True, BFS is equivalent with UCS if the cost of each path is equal, because they will just explore the children nodes alphabetically.
   (c) $d_{i,j} = \alpha,\ \alpha < 0$
   (d) $d_{i,j} = 1$ **Solution:** True, same reason as the previous answer
   (e) $d_{i,j} = -1$
   (f) None of the above

2. Mark all the choices for costs $d_{i,j}$, that make running DFS algorithm on arbitrary graph with these costs $d_{i,j}$ *equivalent* to running UCS. For each choice you choose, state in your reason in not more than 2 sentences.

(a) $d_{i,j} = 0$

(b) $d_{i,j} = \alpha,\ \alpha > 0$

(c) $d_{i,j} = \alpha,\ \alpha < 0$ **Solution:** True, with negative edges, UCS will expand the nodes in the DFS fashion down the chain because they have the least cost.

(d) $d_{i,j} = 1$

(e) $d_{i,j} = -1$ **Solution:** True. Same reason as the previous answer.

(f) None of the above

3. Mark all the choices for costs $d_{i,j}$, that make running UCS algorithm on arbitrary graph with these action costs $d_{i,j}$ *equivalent* to running UCS with the original cost $c_{i,j}$. For each choice of you choose, state in your reason in not more than 2 sentences.

   (a) $d_{i,j} = c_{i,j}^2$

   (b) $d_{i,j} = \frac{1}{c_{i,j}}$

   (c) $d_{i,j} = \alpha c_{i,j},\ \alpha < 0$

   (d) $d_{i,j} = c_{i,j} + \alpha,\ \alpha > 0$

   (e) $d_{i,j} = \alpha c_{i,j} + 10,\ \alpha > 0$

   (f) None of the above **Solution:** True. Adding a certain amount of weight to all edges does not product the same solution (like for example, to eliminate the issue of negative weights by adding a constant value). For paths with many edges, its total weight will increase unfairly.

[Q4] Refer to Figure 2 for this question. S and G are the start and goal nodes respectively. Suppose that we use graph search and we explore the children nodes alphabetically. Notice that all other nodes are not yet labeled. In this question you will have to label them. The path cost for any edge is the 1.

1. What is the minimum cost from S to G? (do this manually, an answer without explanation is sufficient)
   Solution:
   4

2. How would you alphabetically label the nodes, such that running BFS from S to G results in the solution with minimum cost in part (1)? Draw out the labeled graph and show your steps running BFS from S to G. (there are more than 1 answers for this, just pick 1).
   Solution:
   There are multiple answers here. Just ensure that any nodes in that path of cost 4 will be expanded first among its neighbours of the same level.

3. How would you alphabetically label the nodes, such that running DFS from S to G results in the solution with minimum cost in part (1)? Draw out the labeled graph and show your steps running DFS from S to G. (there are more than 1 answers for this, just pick 1).
   Solution:
   There are multiple answers here. Just ensure that any three nodes in that path of cost 4 is labeled A, B, C.

4. How would you alphabetically label the nodes, such that running IDS from S to G results in the solution with minimum cost in part (1)? Draw out the labeled graph and show your steps running IDS from S to G. (there are more than 1 answers for this, just pick 1).
Solution:
There are multiple answers here. Just ensure that any nodes in that path of cost 4 will be expanded first among its neighbours of the same level. (same as running BFS).

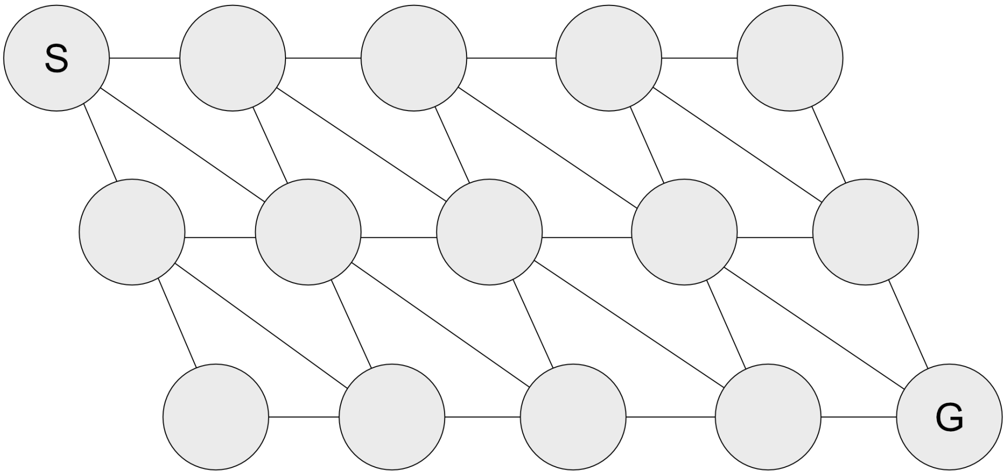

Figure 2: An undirected graph