ECS 170 Homework 1

Hardy Jones
999397426
Professor Davidson
Winter 2014

1. What is the difference among BFS, DFS, and uniform-cost search (Dijkstra's algorithm) with respect to their implementations in the generic tree search algorithm?

   (a) BFS uses a FIFO structure for the fringe.

   (b) DFS uses a LIFO structure for the fringe.

   (c) Uniform-cost search uses a priority FIFO structure for the fringe.

2. Assuming there aren't any programming bugs, what is the most likely reason uniform cost search will sometimes return a greater cost path than depth first search?

   If the branching factor is large, most of the path costs are small, but a then UCS will explore many small-cost branches before exploring a any large-cost branches. If the small-cost branches find a path to the goal, UCS will choose this path as a solution, even though the overall cost would be greater than any large-cost branches

3. Given this information (and assuming there aren't any programming bugs) what is the most likely reason A* would return a greater cost path than Dijkstra's Algorithm?

   A* can have a greater cost path than UCS if we choose a heuristic function which overestimates the cost to reach the goal. It's possible that there is some path on the optimal solution that would have an overestimation for the heuristic. A* would not expand this path, and instead attempt to find some other path. UCS on the other hand would just expand the path like any other.

4. Consider the two statements (a) BFS is a special case of uniform cost search, and (b) uniform cost search is a special case of A*. Under what conditions are they true?

   (a) BFS is a special case of UCS when the cost function is constant.

   (b) UCS is a special case of A* when the heuristic function is constant.

5. Sudoku is a popular game in which the player tries to fill in all blank cells so that the resulting board contains 1 to 9 on each row, each column, and each 3 x 3 block.

   (a) Formulate it as a graph search problem. What are the state space, goal state, successor function, and the path costs?

      i. The state space is any arrangement of the numbers 1-9 in each cell without repetition in the column, row, or 9x9 cell.

      ii. The goal state is all 81 cells are filled in with the numbers 1-9 with no repetition in each column, row or 9x9 cell.

      iii. The successor function chooses a cell and inputs a number from 1-9 where the number chosen is not also in the same row, column, or 9x9 cell.

      iv. The path cost is 1.

   (b) Assume we use uninformed search. Which method would you prefer? Why?

   With uninformed search, it is preferable to use DFS.

   The branching factor can be huge. In the worst case, the branching factor b is 9. The depth of the shallowest goal node depends on the initial state. With an entirely empty board the maximum depth d is 81. This is the same as the the maximum length of any path in the state space, $m = 81$

   In BFS this means we need to store around a maximum of $b^{d+1} = 9^82 = 1.77x10^78$ nodes in memory. In DFS this means we would only store around a maximum of $bm = 9 \cdot 81 = 729$ nodes in memory.

   The two algorithms have similar time complexity, so there is not much difference there.

   Finally, the main issue with using DFS is that it is possible for the search to never end. However, since we have a finite d and m, we are guaranteed to find a solution.

   One is prohibitively expensive, while the other is an actual usable solution. For these reasons, DFS is recommended.

   (c) Is a good heuristic possible in this case? If so, provide one. If not, why not?

   There are quite a few heuristics available for this problem. One such is choosing the cell with the least number of valid numbers, and inserting a number there. We end up with a greater probability of chosing the correct number to insert into a cell.

6. Consider the classic farmer, fox, goose, and grain problem. It turns out that you can pose this as a graph search problem.

   (a) Describe a representation of the state space for this problem. What would the goal state look like in this representation.

   The state space would be each of the farmer, fox, goose, and grain in one of two locations: the west bank, and the east bank.

   The goal state would have each of the farmer, fox, goose, and grain on the east bank.

   (b) What are the possible actions at a particular state?

   There are 8 possible actions:

   i. Move just the farmer to the east bank.
   ii. Move just the farmer to the west bank.
   iii. Move the fox to the east bank.
   iv. Move the fox to the west bank.
   v. Move the goose to the east bank.
   vi. Move the goose to the west bank.
   vii. Move the grain to the east bank.
   viii. Move the grain to the west bank.

   Of course, each possible action involving moving something other than the farmer assumes that said item is on the appropriate bank.

   (c) Our goal is to get everything to the other side in one piece, so what are the constraints on the state space that we need for the actions in part b?

   We have two constraints:

   i. The fox and goose cannot be on one bank without the farmer also on that bank.
   ii. The goose and grain cannot be on one bank without the farmer also on that bank.

   (d) Suppose we want to use A* here. Describe a non-trivial admissible heuristic that we could use.

   One heuristic would be the number of items on the east bank. If a move would put another item on the east bank, it should be preferable to one that would remove an item from the east bank.