Search and HW#2 Q&A

Xiang Zhuoya

zxiang@cs.ust.hk

March 24, 2015

Exercise 1 - **Search a Maze**

Consider the 5 × 5 matrix below, '1' indicates that it is a wall, while '0' indicates that you can move to that grid. Your initial position is at the upper-left corner, and the exit at the lower-right corner (*the goal*). The cost for each move is 1.

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{matrix}$$

1. Run the A^* algorithm with the two heuristic functions:
   $h_1((x,y)) = 0$ and
   $h_2((x,y)) = Manhattan\_dist((x,y), goal) = height - x + width - y$
2. Design another heuristic h_3 and prove that it is admissible and dominates h_2

Solution 1 - Search a Maze

1. Label the explored nodes with '3'

For $h_1((x,y)) = 0$,

$$\begin{matrix} 3 & 3 & 3 & 3 & 3 \\ 1 & 1 & 3 & 1 & 1 \\ 3 & 3 & 3 & 3 & 3 \\ 0 & 1 & 3 & 1 & 1 \\ 0 & 1 & 3 & 3 & 3 \end{matrix}$$

For $h_2((x,y)) = Manhattan\_dist((x,y), goal) = height - x + width - y$,

$$\begin{matrix} 3 & 3 & 3 & 3 & 3 \\ 1 & 1 & 3 & 1 & 1 \\ \color{red}{0} & \color{red}{0} & 3 & 3 & 3 \\ 0 & 1 & 3 & 1 & 1 \\ 0 & 1 & 3 & 3 & 3 \end{matrix}$$

Solution 1 - Search a Maze Cont'd

2. $h_3((x,y))$ is given by

$$h_3 = \begin{cases} h_2 + 2, & \text{if both } (x+1,y) \text{ and } (x,y+1) \text{ are wall,} \\ h_2, & \text{otherwise.} \end{cases}$$

Since two more moves must be made by going up or left, the heuristic is still admissible and dominates h_2. The table below shows the number of expanded nodes on different size **sample mazes** using h_1, h_2 and h_3 *(Results from past assignment)*.

	*maze*1	*maze*2	*maze*3	*maze*4	*maze*5	*maze*6
size	29×29	29×29	49×49	49×49	99×99	99×99
h_1	442	447	1249	1247	4301	4966
h_2	257	439	1153	1190	1010	4667
h_3	225	438	1146	1188	993	4655

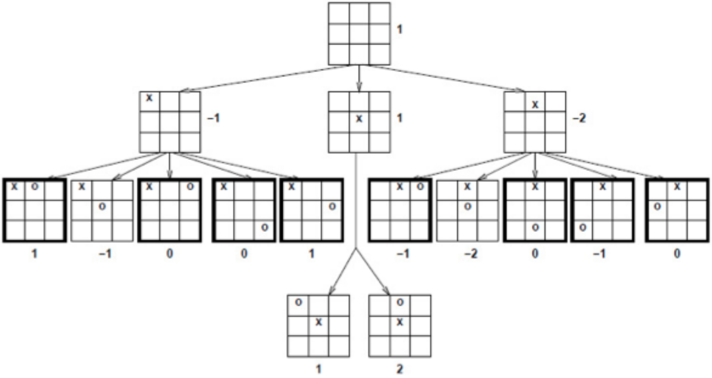
Exercise 2 - Tic-Tac-Toe

Consider *tic-tac-toe*. We define X_n as the number of rows, columns, or diagonals with exactly n X's and no O's. Similarly, O_n is the number of rows, columns, or diagonals with just n O's. The utility function assigns $+1$ to any position with $X_3 = 1$ and -1 to any position with $O_3 = 1$. All other terminal positions have utility 0. For non-terminal positions, we use a linear evaluation function defined as $Eval(s) = 3X_2(s) + X_1(s) - (3O_2(s) + O_1(s))$.

1. Show the whole game tree starting from an empty board down to depth 2, taking symmetry into account. Mark on your tree the evaluations of each position (use the minimax algorithm for depths 1 and 0).
2. What is the best starting move?
3. Circle the nodes at depth 2 that would not be evaluated if $\alpha - \beta$ pruning were applied, assuming the nodes are generated in the optimal order.

Solution 2 - **Tic-Tac-Toe**

The evaluation function values are shown below the terminal nodes and the backed-up values to the right of the non-terminal nodes. The values imply that the best starting move for X is to **take the center**. The terminal nodes with a bold outline are the ones that are pruned.



P1

Problem 1 (10%). (Russell and Norvig) The following payoff matrix shows a game between politicians and the Federal Reserve in the US:

	Fed: contract	Fed: do nothing	Fed: expand
Pol: contract	1,7	4,9	6,6
Pol: idle	2,8	5,5	9,4
Pol: expand	3,3	7,2	8,1

The politicians can expand (increase spending) or contract (cut spending) fiscal policy (or stay idle), while the Fed can expand (lower interest rate) or contract (increase interest rate) monetary policy (or do nothing). Find the Nash equilibria of this game.

P2

Problem 2 (10%). Consider the following zero-sum game:

	A	B
A	a,-a	b,-b
B	c, -c	d, -d

1. Formulate conditions under which (A, A) is a Nash equilibrium.
2. Formulate conditions under which (B, B) is a Nash equilibrium.
3. Formulate conditions under which both (A, A) and (B, B) are Nash equilibria.

P3

Problem 3 (10%). Consider an auction by first price with two agents, with ties broken randomly. It's a common knowledge that both agents have value 6 for the item, and they can bid only positive integers. Formulate this auction as a game and find all Nash equilibria if there is any.

P4

Problem 4 (10%). Given a graph and a node in it as the starting point, the traveling salesman problem (TSP) is about finding a least cost path that starts and ends at the starting node, and goes through each other node in the graph once and exactly once. Formulate this problem as a search problem by specifying the states, possible initial states, goal test, and operators.

P5

Problem 5. (10%) (Russell and Norvig) Consider the problem of constructing crossword puzzles: fitting words into a grid of intersecting horizontal and vertical squares. Assume that a list of words (i.e. dictionary) is provided, and that the task is to fill in the squares using any subset of this list. Formulate this problem as a search problem by specifying the states, possible initial states, goal test, and operators.

P6

Problem 6. (20%) Consider the blocks world where a block can be on the table or on another block. For our problem, suppose there are three blocks named B1, B2, and B3. The initial state can be any configuration of the three blocks, and the goal is to have B1 on B2, B2 on B3, and B3 on the table. The actions in this domain are:

- move(x,y,z), where x is a block, and y and z can be blocks or the table. The effect of this action is to move block x from y to z, provided x is clear (no other block on top of it), x is on y, and either z is the table (which always has room) or z is clear. Thus after this action, x is on z in the new state.

Now consider the following initial state:

- B1 is on the table; B3 is on B2, and B2 is on the table.

P6 Answer the following questions:

5.1 Draw a search tree according to the iterative deepening, begin with the depth bound set to 0, increase it by 1 on each iteration, and terminate when the depth bound is 2. Do not repeat a state if it has already been expanded in the same branch. You can assume an arbitrary order of operators. You should label a node with a number indicating the order that the node is selected for expansion.
5.2 Design an admissible heuristic function for this domain. It should be admissible for all possible states, not just those that can be reached from the above initial state. You should also briefly justify why your heuristic function is admissible.
5.3 Assume that each operator costs one unit, draw a search tree according to the A* search by tree using the heuristic function that you designed in (5.2). You should label each node by two things: its f(n) = h(n)+g(n) value, and a number indicating the order by which it is selected for expansion.

P7

Problem 7. (20%) Consider the following search problem:

- state space: $\{S, A, B, G\}$;
- operators: O_1 maps S to A with cost 4, O_2 maps S to B with cost 2, O_3 maps B to A with cost 1, and O_4 maps from A to G with cost 5;
- initial state S;
- goal state G;
- heuristic function h: $h(S) = 7$, $h(A) = 1$, $h(B) = 6$, and $h(G) = 0$.

Solve the problem in two ways:
1. A* by tree. Number the nodes according to their order of expansion, and label the nodes by their f(n) = g(n) + h(n) values.
2. A* by graph. Draw the sequence of graphs generated by the search, and in each graph, color by red those edges that are maintained as pointers, i.e. the least cost paths from the starting node to each node that has already been generated.