COMP 3211 2015 Spring Semester Assignment #2

Date assigned: Wednesday, Mar 18, 2015

Due time and place: 17:00 on Friday, Mar 27, in a collection box outside Rm4215

Penalties on late papers: 20% off each day (anytime after the due time is considered late by one day)

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.

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.

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.

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.

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.

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.

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.

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.