CSE 560 Midterm

Olcay Taner YILDIZ

I. Question (18 points)

Suppose you're trying to solve the following puzzle. The puzzle involves numbers from 100 to 999. You're given two numbers called S and G. You're also given a set of numbers called bad. A move consists of transforming one number into another by adding 1 to one of its digits or subtracting 1 from one of its digits; for instance, a move can take you from 678 to 679; or from 234 to 134. Moves are subject to the following constraints:

- You cannot add to the digit 9 or subtract from the digit 0. That is to say, no carries are allowed and the digits must remain in the range from 0 to 9.
- You cannot make a move which transforms your current number into one of the numbers in the bad set.
- You cannot change the same digit twice in two successive moves.

Since the numbers have only 3 digits, there are at most 6 possible moves at the start. And since all moves except the first are preceded by another move which uses one of the digits, after the start there are at most 4 possible moves per turn. You solve the puzzle by getting from to in the fewest possible moves. Your task is to use A* search to find a solution to the puzzle.

(a) Briefly list the information needed in the state description (not the node description) in order to apply A* to this problem.
(b) Find a heuristic for use with A* search in this problem which is admissible and which does not require extensive mathematical calculation (that is, you should be able to use it in solving the second part of this problem without needing a calculator!). Explain clearly why your heuristic is admissible. Try to find a heuristic which is as powerful as possible while still remaining admissible.
(c) Give a domain-independent description of what f, g, and h values represent in A* search in general.
(d) Use your heuristic to carry out an A* search to find a solution when G = 567, S = 777, and bad = {666, 667}. Draw the search tree as you go, showing for each node both the state it represents; (include all information needed to distinguish this state from other states) and the node's f, g, and h values. Also, whenever you expand a node (generate its successors), label it with a number indicating the order of expansion (the start node would be labeled 1, the next node to be expanded would be 2, etc.). Include in your tree all legal successors of each node you expand.

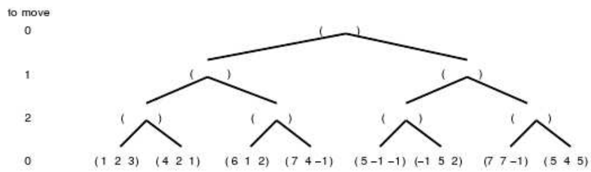
II. Question (18 points)

Decide if each of the following is true or false. Explain why?

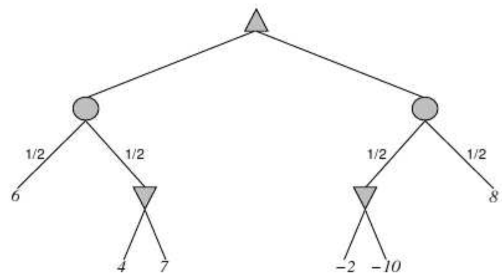
(a) Breadth-first is an optimal search algorithm.
(b) Simple reflex agents cope well with inaccessible environments.
(c) Minimax and alpha-beta can sometimes return different results.
(d) It is possible to write an exact evaluation function for chess.
(e) Assume that a king can move one square in any direction on a chessboard (8 directions in all). Manhattan distance is then an admissible heuristic for the problem of moving the king from square A to square B.
(f) It is possible to build a knowledge-based agent that is a pure reflex agent.
(g) A rational agent outperforms all nonrational agents because it knows the actual outcome of its actions.
(h) Simple hill-climbing is a complete algorithm for solving CSPs.
(i) h(n) = 0 is an admissible heuristic for 8-puzzle.
(j) Some pruning is possible in game trees with chance nodes.
(k) A perfectly rational backgammon agent never loses.
(l) Search algorithms cannot be applied in completely unobservable environments.

III. Question (18 points)

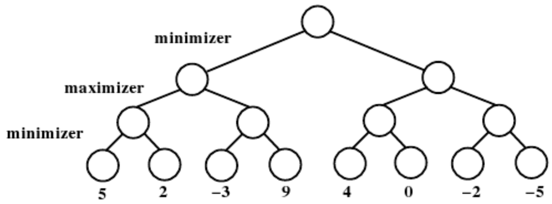
(a) Draw the smallest possible game tree on which alpha-beta will prune at least one leaf node. Make sure to label the leaves with values, and circle the leaf (or leaves) that will be pruned.
(b) Copy and complete the following game tree by filling in the backed-up value triples for all remaining nodes:



(c) Copy and complete the following game tree by filling in the values for all remaining nodes:



(d) Apply the minimax algorithm to the game tree below, where it is the minimizers turn to play. Report the estimated values of the intermediate nodes and indicate the proper move of the minimizer.



Indicate, by crossing out, one (1) unnecessary call to the static board evaluator. Explain why this call to the board evaluator is unnecessary.

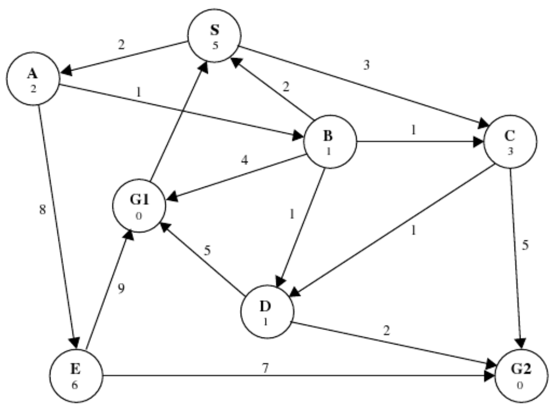
IV. Question (18 points)

Consider the problem of placing k knights on an n × n chessboard such that no two knights are attacking each other, where k is given and $k \leq n^2$.

(a) Choose a CSP formulation. In your formulation, what are the variables?
(b) What are the values of each variable?
(c) What sets of variables are constrained, and how?
(d) Now consider the problem of putting as many knights as possible on the board without any attacks. We will solve this using local search. Briefly describe in English a sensible successor function.

V. Question (18 points)

Consider the search space below, where S is the start node and G1 and G2 satisfy the goal test. Arcs are labeled with the cost of traversing them and the estimated cost to a goal is reported inside nodes. For each of the following search strategies, indicate which goal state is reached (if any) and list, in order, all the states popped off of the OPEN list. When all else is equal, nodes should be removed from OPEN in alphabetical order.



(a) Iterative Deepening
(b) A^*
(c) Hill climbing
(d) Breadth-first
(e) Depth-first

VI. Question (10 points)

Consider the following fitness function:

$$fitness = 5a + 3bcd + 2e \quad (1)$$

where a-e are all Boolean-valued parameters. Compute the fitness of each of the members of the initial population below. Also compute the probability that each member of the population will be selected during the fitness-proportional reproduction process.

a	b	c	d	e
1	1	0	1	1
0	1	1	0	1
1	1	0	0	0
1	0	1	1	1
1	0	0	0	0

Assuming the first two of members of the population are selected for reproduction, and the cross-over point is that between the b and the c, show the resulting children.