Name: Solutions UW CSE 473 Midterm, Fall 2014

Instructions

Please answer clearly and succinctly. If an explanation is requested, think carefully before writing. Points may be removed for rambling answers. If a question is unclear or ambiguous, feel free to make the additional assumptions necessary to produce the answer. State these assumptions clearly; you will be graded on the basis of the assumption as well as subsequent reasoning. **On multiple choice questions, incorrect answers will incur negative points** proportional to the number of choices. For example a 1 point true-false question will receive 1 point if correct, -1 if incorrect, and zero if left blank. Only make informed guesses.

There are XXXX problems worth KKK points on PPP pages.

***1.* (1 point) Who are you?** Write your name at the top of every page.

2. (1 point each – total of 4). Types of Agents You are developing an agent that solves crossword puzzle (like the one pictured to the right) using an exhaustive dictionary of possible words. States are partially-completed puzzles and actions place a word on the puzzle. On each line below, we've listed two possible environmental aspects; circle the one which better describes the crossword-puzzle environment.

a) (fully observable) vs. partially observable

b) (single agent) vs. multiagent,

c) stochastic vs. (deterministic,)

d) (discrete) vs. continuous

3. (1 point each – total of 13) True / False Circle the correct answer.

(a) (T) F Iterative deepening search is guaranteed to expand more nodes than breadth-first search (on any graph whose root is not the goal). boundary case

(b) (T) F A* search with a heuristic that is not completely admissible may still find the shortest path to the goal state.

(c) T (F) Consider a finite, acyclic search space where depth-first search is guaranteed to eventually find a solution and the root is not a goal. In this situation iterative deepening search will always explore more nodes than depth-first.

(d) T (F) A pattern database helps an agent avoid wasting time in cycles by storing previously-expanded states.

(e) (T) F Random restarts are often used in local search to diminish the problem of local maxima.

(f) T (F) Doubling your computer's speed allows you to double the depth of a tree search given the same amount of time

(g) T (F) Every CSP with higher order constraints can be rewritten as a binary CSP with the same number of variables.

(h) (T) F If a binary CSP has a tree-structured constraint graph, we can find a satisfying assignment (or prove no satisfying assignment exists) in time that is linear in the number of variables.

(i) (T) F Backtracking search on CSPs, while generally much faster than general purpose search algorithms like A*, still requires exponential time in the worst case.

(j) (T) F One reason to use forward checking in a CSP problem is in order to detect failures quickly and backtrack earlier.

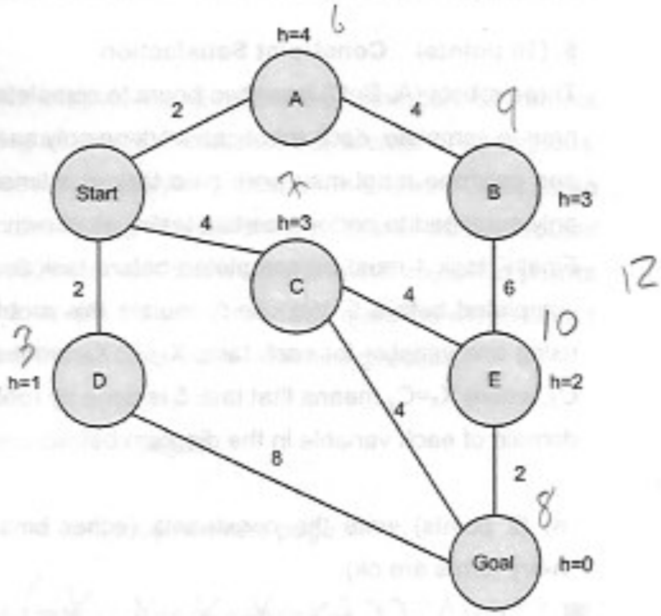
(k) (T) F An agent that uses Minimax search, which assumes an enemy behaves optimally, may well achieve a better score when playing against a suboptimal enemy than the agent would against an optimal enemy.

(l) (T) F All other things being equal, value iteration will converge in fewer iterations, when the discount factor, gamma, is smaller.

(m) (T) F Expectimax search can be used to solve an MDP in a finite horizon setting.

(n) T (F) The optimal policy for an MDP depends on the MDP's start state.

4. (2 points each – total of 8) Search. Given the graph to the right, write down the order in which the states are visited by the following search algorithms. If a state is visited more than once, write it each time. Ties (e.g., which child to first explore in depth-first search) should be resolved according to alphabetic order (i.e. prefer A before Z). Remember to include the start and goal states in your answer. Treat the goal state as G when you break ties. Assume that algorithms execute the goal check when nodes are visited, not when their parent is expanded to create them as children.



(a) Iterative deepening depth first search

optimal → S S A C D S A B C E G ___

(b) A* search, where f(n)=g(n)+h(n)

S D A C G G ___ ___ ___ ___ ___

5. (10 points) Constraint Satisfaction

Three robots (A, B, C) have two hours to complete five tasks (1,2,3,4,5). Each task takes one hour to complete, each robot can work on only one task at a time, and only one robot may work on a task at a time. Each robot is only equipped to perform certain tasks, as shown in the table. Finally, task 1 must be completed before task 2, and 3 must be completed before 5. We can formulate this problem as a CSP, using one variable for each task: $X_1, \ldots, X_5$, whose possible values are a subset of $A_1, A_2, \ldots C_2$, where $X_5 = C_2$ means that task 5 is done by robot C and **Time(X_5)** is 2. We have written the domain of each variable in the diagram below.

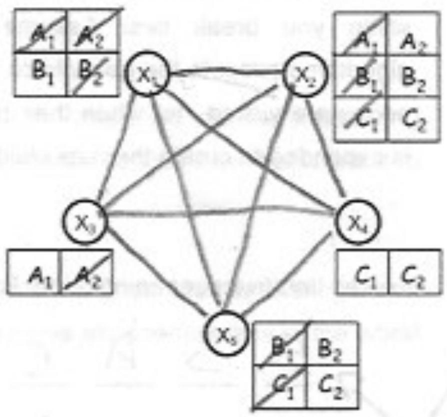
Robot	Tasks
A	1,2,3
B	1,2,5
C	2,4,5

a) (2 points) write the constraints (either binary or n-ary forms are ok).

Alldiff(X_1 X_2 X_3 X_4 X_5)

Time(X_1) < Time(X_2)

Time(X_3) < Time(X_5)



b) (1 point) Complete the drawing above to show the constraint graph.

All Diff means it is fully connected

d) (4 points) Is the initial state arc-consistent? No

If not, cross out the values for each variable that would be pruned by running AC-3 .

e) (4 points) Solve the (reduced, arc-consistent) CSP using backtracking search (without forward checking). Use the minimum remaining values (MRV) variable ordering (breaking ties in numerical order), and least constraining value (LCV) value ordering (breaking ties in alpha-numerical). What order are the first variables assigned, and what values are they given?

The first variable assigned is X_1 it's given value B_1

The second variable assigned is X_3 it's given value A_1

The third variable assigned is X_4 it's given value C_1

6 (9 points) MDPs. Consider a setting where every 6 months Apple decides whether to release a new version of the iPhone or not. Assume the problem can be represented as an MDP with states (G=Good, M = Mediocre, B = Bad), each referring to the public sentiment towards Apple. The actions are R=Release, D = Don't release. Taking an action that lands in state G (from any other state, including itself) receives reward 2, landing in state M receives reward 0, and landing in state B receives reward -1. The discount factor (γ) is 1. The transitions are as shown in the table:

	To G	**To M**	**To B**
From G Take R	0.1	0.9	0.0
From G Take D	0.2	0.8	0.0
From M Take R	0.1	0.9	0.0
From M Take D	0.0	0.3	0.7
From B Take R	0.9	0.0	0.1
From B Take D	0.0	0.5	0.5

a) For this MDP, fill in the blank spaces in the value iteration table (1 point for each Q entry)

	G	**M**	**B**
V_0(state)	0	0	0
Q_1(state, R)	0.1(2) + 0.9(0)= 0.2	0.1(2) + 0.9(0)= 0.2	0.9(2) + 0.1(-1)= 1.7
Q_1(state, D)	0.2(2) + 0.8(0)= 0.4	0.3(0) + 0.7(-1)= -0.7	0.5(0) + 0.5(-1)= -0.5
V_1(state)	max(0.2, 0.4)= 0.4	max(0.2, -0.7)= 0.2	max(1.7, -0.5)= 1.7
Q_2(state, R)	Not required	0.1*(2 + 1*0.4) + 0.9(0 + 1*0.2)= 0.24+0.18= 0.42	Not required
Q_2(state, D)	Not required	0.3*(0 + 1*0.2) + 0.7(-1 + 1*1.7)= 0.06+0.49= 0.55	Not required
V_2(state)	Not required	max(0.42, 0.55)= 0.55	Not required

b) (1 point) What should Apple do if it is in state M with a horizon of 2?

Action D, since it has a higher Q-value (0.55) than Action R (0.42).