EECS 348 Introduction to Artificial Intelligence

Practice Midterm Exam - Solution

May 12, 2010

Partial credit will be given, so feel free to explain your work.

- 1. (10 points total) You have a list of the top 10 restaurants in the country, and want to eat at all of them. You don't want to visit two steakhouses in a row, but otherwise simply want to minimize your total travel time.
 - (a) (6 points) Express your situation as a search problem. That is, define the states, operators, and goal test you will use.

Solution: States = Partial paths through the restaurants starting at my home city. Operators = adding a new restaurant, or a return to my home city, to the current path. Goal test = true if I've visited all 10 cities and returned home.

(b) (1 point) What search algorithm would you use for this problem?Solution: I'd make a heuristic and use A* search. Note that I need an optimal solution, according to the problem statement (path cost is travel time).

(c) (3 points) Assume you're using A* search. Devise a heuristic that is admissible, and another that is inadmissible. With which will you obtain optimal solutions?

Solution: Admissible heuristic: straight line travel time from the city at the end of the path to the farthest restaurant city, then back home (I'll assume I can get straight-line travel time easily). Inadmissible heuristic: the travel time if I visit all unvisited restaurants in alphabetical order, then return home. With A*, the admissible heuristic will give optimal solutions.

2. (10 points total) Consider the Tic-Tac-Toe board in Figure 1.

X	0	X
Ο	0	
	Х	



- (a) (6 points) X is to move. Draw the full minimax tree for this problem.
 Number the moves #1-3, from top to bottom and left to right. Then the minimax tree is shown in Figure 2.
- (b) (2 points) According to the minimax algorithm, which move should the maximizing player make? Move #1, to block the O player from winning.
- (c) (2 point) Describe behavior for the O player in which the Minimax move given above is not optimal. If the Min player were suboptimal such that if Max makes move #3, then Min (perhaps losing concentration due to the potential for impending glorious victory) will erroneously make



Figure 2: Minimax Example

move #2, then Max would actually win by making move #3 instead of typing with #1.

3. (4 points total) Describe four problems, where each one is best solved by a different technique from the following list: Constraint Satisfaction with Arc Consistency, Uninformed Search, A* Search, and Minimax. Each problem should be described in two-three sentences each, and invent these (don't use standard examples from class).

Solution:

CSP with Arc Consistency: Consider trying to assign students to project teams in which some number of programmers must be in each group, some students don't want to work together, and others really do want to work together. What's an assignment that meets all these constraints?

Uninformed Search: There aren't very many cases in which you'd want to use uninformed search. Here's one: say you have a binary program that is corrupted somehow – a few of its bits are wrong – but you don't know which ones. You want to find the smallest change you can make to the individual bits such that you get a program that runs without failure (and all you can do to check the code is running it to see if it fails).

A* Search: Assume you're trying to plan a path of a Mars rover back to base, and you're using photographs to determine the shortest path that avoids several regions of impassable terrian.

Minimax: Building a game player for Go (technically we did discuss this briefly in class).

4. (6 points) Say you had a meeting to convince Loebner to make one *incremental* change to the Loebner prize that would improve the contest. In about five sentences, what would your change be, and why?

The Loebner prize would have additional value if the annual winners produced usable technology. To get closer to this goal, I would propose changing the existing contest such that an additional bonus prize be awarded to the best dialogue system in computer games released each year. While game dialogue systems do not require the general AI capabilities needed to pass the Turing test, they do address a very similar problem (typically *restricted* dialogue with human beings). Thus, achievements in game dialogue systems are incremental steps toward passing the Turing test; further, unlike Loebner prize winners, they are guaranteed to be technology already in use in commercial software. Rewarding the best AI in game dialogue systems would share many of the virtues of the Loebner prize, such as adding visibility to the persuit of AI, and focusing scientists and engineers on fundamental AI challenges.