EECS 391: Introduction to AI (Spring 2012) Written Homework 2 (Max Points: 100)

Assigned Tuesday January 31, due 5pm Tuesday February 7. Write your answers neatly and remember to show all relevant work. Before turning in your work, staple your answer sheets together and write your name(s) and Case ID(s) on the front page.

1. Prove that every consistent heuristic is also admissible. (10 points)

2. In normal real-time strategy games, unlike in the programming assignment, the map is not static. New elements (such as buildings and walls) are being added and deleted over time. How would you adapt A* to perform pathfinding in such a dynamic environment? (10 points)

3. In class we discussed how to use genetic algorithms for optimization. Consider agent functions that control an agent's behavior, where each function is a collection of n IF-THEN rules. Here n is the number of actions, and each rule says something like "If the state has some properties P, perform action A." Specify state descriptions and fitness functions that you could use in a genetic algorithm to evolve such functions for an agent that (1) plays tic-tac-toe and (2) controls an elevator. (10 points)

4. Gradient ascent search is prone to local optima just like hill climbing. Describe how you might adapt simulated annealing to gradient ascent search to reduce this problem. (10 points)

5. Derive a condition on the class of functions f for which the Newton-Raphson method guarantees improvement at each step, i.e., each iteration of Newton-Raphson strictly decreases the function value. (10 points)

6. Describe how the minimax and alpha-beta algorithms change for two-player, *non-zero-sum* games in which each player has a distinct utility function and both functions are known to both players. If there are no constraints on the terminal utilities, is it possible for any node to be pruned by alpha-beta? What if the player's utility functions on any state differ at most by a constant k? (10 points)

7. Prove that a positive linear transformation of leaf values, where a leaf with value x gets a new value $ax+b$, $a > 0$, leaves the choice of moves unchanged in a game tree. Repeat for trees with chance nodes. (10 points)

8. Consider the following procedure for generating moves in trees with chance nodes:

- Generate some long dice roll sequences.
- Now traverse the game tree and whenever a chance node is encountered, use a pre-generated dice roll to proceed. In this way the game tree becomes deterministic.
- Solve the deterministic tree using alpha-beta.

- Use the results to estimate the value of each move and choose the best.

Will this procedure work well? Why or why not? (10 points)

9. Prove: For every game tree, the utility obtained by MAX using minimax decisions against a suboptimal MIN will never be lower than the utility obtained playing against an optimal MIN. Show a game tree where MAX can do better than the minimax strategy by playing a suboptimal strategy against a suboptimal MIN. (10 points)

10. By what names are these algorithms more commonly known? (10 points)

i. Beam search with k=1.
ii. Beam search with one initial state and no limit on k.
iii. Simulated annealing with T=0 (or very very close to zero) always.
iv. Simulated annealing with T=∞ (or very very large) always.
v. Genetic algorithm with population size=1.