CS4881 Artificial Intelligence

Problem Set 2

1) Intelligent Agents

For each of the following agents, develop a PEAS description of the task environment:

1. Playing tennis against a wall
2. Playing a tennis match (with another opponent : human or humanoid)
3. Hospital robot for dispensing medications to patients
4. Bidding on an item at an auction
5. Autonomous Mars rover

2) Uninformed Search

The missionaries and cannibals problem is usually stated as follows: Three missionaries and three cannibals are on one side of a river, along with a boat that can hold one or two people. Find a way to get everyone to the other side, without ever leaving a group of missionaries in one place outnumbered by the cannibals in that place.

1. Formulate the problem precisely, making only those distinctions necessary to ensure a valid solution. Draw a diagram of the complete state space (state diagram with nodes defining states and edges defining state transitions).
2. Solve the problem optimally using an appropriate search algorithm. Is it a good idea to check for repeated states? Try implementing this in code! Otherwise, you can solve this by hand, but you need to show how your optimal search algorithm visits the nodes in your state representation.
3. Why do you think people have a hard time solving this puzzle, given that the sate space is so simple?

3) Informed Search and Exploration

Generate a large number of random 8-puzzle instances and solve them using the following local search techniques (where possible) by:

- hill climbing using number of Misplaced Tiles heuristic – h(n)
- hill climbing using Manhattan Distance heuristic – h(n)
- Any search algorithm that performs better than the prior search algorithms with respect to higher percent of problems solved. Example: hill climbing with random restart, *simulated annealing, A*, etc.*

Measure the search cost and percentage of solved problems for each algorithm and plot a graph of these against the optimal solution cost. Comment on your results. Bonus: keep track of visited nodes to find optimal solution!

Note: Valid 8-puzzle problems must be generated from a solution of the 8-puzzle game, otherwise you could get unsolvable puzzles.

Submission:

Jay Urbain, PhD

Submit your source code and *one* PDF file to Blackboard. Name your archive and your report with your name, class, and problem set number. This is a 2-week lab. See the course outline for the assignment due date.