Artificial Intelligence (CS 436)

Homework #1—Due Monday, September 22, 2008

All work must be individual effort, and use of the web is prohibited (unless otherwise indicated). Any external sources must be cited. All papers are due at the start of class on the due date.

1. R&N 4.11: Give the name of the algorithm that results from each of the following special cases:
   (a) [2.5 points] Local beam search with $k = 1$.
   (b) [2.5 points] Local beam search with $k = \infty$.
   (c) [2.5 points] Simulated annealing with $T = 0$ at all times.
   (d) [2.5 points] Genetic algorithm with population size $N = 1$.

2. An agent has to deliver identical packages to locations A, B, and C, in an office environment. Assume the agent starts off holding all three packages. The environment is represented as a grid of squares, some of which are free (so the agent can move into them) and some of which are occupied (by walls, doors, etc.). The agent can move into neighboring squares and can pick up and drop packages if the packages are in the same square as the agent.

   If we formulate the this problem as a state-space search problem, we can specifying the state space, action space, goal test, and cost function as follows:

   - State: $\{(x, y)$, deliveredA, deliveredB, deliveredC$\}$
   - Action: moveN, moveE, moveS, moveW, dropA, dropB, dropC
   - Goal: $s_i = \{(\#, \#), \text{True}, \text{True}, \text{True}\}$, where $(\#, \#)$ indicates any position
   - Cost: a cost of 1 for each action taken

   (a) [5 points] Give a non-trivial, polynomial-time, admissible heuristic for this domain, or argue why there isn't one.
   (b) [5 points] Package A has to be delivered to the boss, and it is important that it be delivered quickly. At the same time, the agent should try to deliver the other packages promptly as well. How should we encode this problem?
   (c) [5 points] Now consider the case where the agent does *not* start with all of the packages but has to pick up a package from location 1 and deliver it to location A, a package from location 2 to deliver to location B, and a package from location 3 to deliver to location C. What is an appropriate state space for this problem?
   (d) [5 points] What is a good admissible heuristic for the modified problem? Use a different heuristic from the one you used in part (a).

3. [10 points] Suppose we have a problem space with a uniform branching factor b and a single goal node at depth $m > 0$. What is the *minimum* number of nodes expanded for breadth first search and depth first search? Justify your answer.

4. A *SAT* or *Boolean satisfiability problem* is a decision problem regarding the existence of a complete assignment of truth values (*true* or *false*) to the variables in a clause that make the entire clause true. The clause in question consists of variables, parentheses, and Boolean connectives. Such a clause is *satisfiable* if there are one or more complete assignments that make the clause true and *unsatisfiable* if no such assignment exists.

   We would like to solve *SAT* using a greedy hill climbing algorithm. Each state corresponds to a complete assignment. The successor operator $Succ(s)$ generates all neighboring states of s, which we will define as all total assignments that differ by one. So, for example, given the state with total assignment $\{A \leftarrow true, B \leftarrow false\}$, the neighboring states would be $\{A \leftarrow false, B \leftarrow false\}$ and $\{A \leftarrow true, B \leftarrow true\}$. We can set the evaluation of a state to be the number of clauses that are satisfied, given the assignment of the state. This algorithm is usually called *GSAT*. Assume that ties in the evaluation function are broken randomly.

   (a) [5 points] If you have n variables, how many neighboring states does the $Succ(s)$ function produce?

   (b) [5 points] What is the total size of the search space, i.e., how many possible states are there total? Assume again that there are n variables.

   (c) [5 points] Consider the following set of clauses:

   $$\{\neg A \vee B \vee C\}, \{A \vee \neg B \vee C\}, \{A \vee B \vee \neg C\}, \{A \vee B \vee C\}.$$

   Come up with a non-goal state (i.e., a non-satisfying assignment) that is on a plateau in our hill climbing space.

   (d) [5 points] Consider the following *SAT* problem:

   $$(X \vee Y) \wedge (\neg X \vee Z) \wedge (\neg Y \vee Z) \wedge (\neg X \vee \neg Z) \wedge (X \vee U) \wedge (X \vee V)$$

   From the starting point below, present a sequence of moves that *GSAT* can execute that will result in the global maximum. For each move, give the resulting assignment and the value of the *GSAT* evaluation function for that assignment.

   $$\{X = true, Y = false, Z = true, U = false, V = false\}; val = 5.$$

5. If $h_1(s)$ and $h_2(s)$ are both ***admissible*** heuristic functions, which of the following are also admissible? Justify your answer. Among the admissible heuristics, state which you believe to be the "best" and explain why.

   (a) [5 points] $h_3(s) = h_1(s) + h_2(s)$

   (b) [5 points] $h_3(s) = |h_1(s) - h_2(s)|$, where $|\cdot|$ indicates absolute value.

   (c) [5 points] $h_3(s) = \max\{h_1(s), h_2(s)\}$

   (d) [5 points] $h_3(s) = \min\{h_1(s), h_2(s)\}$

6. [10 points] R&N 5.6 (modified): Solve the following cryptarithmetic problem by hand, using (simultaneously) backtracking, forward checking, and the MRV and least-constraining value heuristics.

   $$TWO + TWO = FOUR$$

7. [10 points] We wish to optimize a function of k variables, $f(x_1, \ldots, x_k)$, given $f : \Re^k \rightarrow \Re$. We assume we know the domains of the variables as $D_i \in [a_i, b_i] \subset \Re$. We assume, without loss of generality, that $f(x_1, \ldots x_k) > 0$, for all $x_i \in D_i$. Assume we want to represent our answer to n significant digits. Derive a general expression for determining the number of bits required in solving a function optimization problem with a standard (i.e., binary) genetic algorithm. Explain the encoding method to be used and derive a decoding expression for a chromosome encoded according to your method.