

# Principles of Artificial Intelligence

Vasant Honavar

Department of Computer Science

Iowa State University

Problem set 2, Fall 2009

Due September 23, 2009

**Note:** The problems marked with \*\* are optional for ComS 472 students. The rest of the problems are optional for ComS 572 students. Please feel free to consult with the instructor or the TA if you have questions.

1. (10 pts.) Solve problem 3.6 from the Russell and Norvig text.
2. (10 pts.) Solve problem 3.7 from the Russell and Norvig text.
3. (10 pts.) Solve problem 3.13 from the Russell and Norvig text.
4. (10 pts.) Solve problem 3.17 from the Russell and Norvig text.
5. (10 pts.) Solve problem 4.2 from the Russell and Norvig text.
6. (10 pts.) Solve problem 4.3 from the Russell and Norvig text.
7. (10 pts.) Solve problem 4.7 from the Russell and Norvig text.
8. (10 pts.) Solve problem 4.8 (parts a and b) from the Russell and Norvig text.
9. (10 pts.) Solve problem 4.9 from the Russell and Norvig text.
10. (10 pts.) Suppose you are a student at an unnamed midwestern university with a really laid back schedule (Of course, we all know that the unnamed university can't be ISU! :-). Every morning after breakfast at the dormitory, you walk to visit once each of the following locations: the gym, the union, the computer lab (mostly to surf the web), the fastfood joint on campus for lunch, the library (mostly to read some magazines and to take a nap), the raquetball court, and back to the dormitory in time for dinner. Suppose you want to compute a path that minimizes the distance that you have to walk every day. (Roughly) rank order the following search procedures in terms of their desirability (as a function of completeness, admissibility, search effort likely to be needed) and justify your answer:
  - (a) Best first search
  - (b) Hillclimbing search
  - (c)  $A^*$  search
  - (d) Branch-and-bound search with dynamic programming
  - (e) Bidirectional Best first search

- (f) Bidirectional Hillclimbing search
- (g) Bidirectional  $A^*$  search

11. (10 pts.)

Consider a search space implicitly specified by a set of rules given below. A rule  $p \rightarrow q_1, q_2, \dots, q_n$  means problem  $p$  is solved if *each subproblem*  $q_i$  (where  $i = 1 \dots n$ ) is solved. A heuristic function for this problem is specified below. Assume that  $A$  is the start node; the cost of a  $k$ -connector is  $k$ ; for a state representing a primitive problem, the  $h$  value is 0; and any problem or subproblem that has a non-zero  $h$  value and cannot be decomposed into subproblems is unsolvable.

**Rules specifying the search space:**

$A \rightarrow B$	$A \rightarrow C$	$B \rightarrow D$	$B \rightarrow E$	$C \rightarrow F, G$	$D \rightarrow H, I$
$E \rightarrow J, K, L$	$F \rightarrow M$	$G \rightarrow N$	$G \rightarrow O$	$G \rightarrow P$	$H \rightarrow Q$
$H \rightarrow R$	$I \rightarrow R$	$L \rightarrow J, S, M$	$N \rightarrow U$	$O \rightarrow T, U$	$P \rightarrow U$

The value  $v$  returned by heuristic function for a node  $x$  is specified by entries of the form  $x(v)$  in the following list:

$A(1), B(3), C(2), D(3), E(4), F(1), G(10), H(2), I(3), J(0), K(0)$   
 $L(2), M(0), N(16), O(5), P(27), Q(50), R(30), S(0), T(90), U(200)$ .

Draw the search space generated by  $A^*$  by showing successive snap-shots of the search tree after each of the FIRST FOUR node expansions. (Expanding a node means applying all applicable rules at the node and generating the node's successors. For example, expanding  $A$  yields two successors  $B$  and  $C$  and  $A$  can be solved by solving either  $B$  OR  $C$ .) In *each* snap shot, include the *current* estimated cost of the minimum cost solution rooted at each node.

12. \*\* (20 pts.) Solve problem 4.12 from the Russell and Norvig text.

13. \*\* (20 pts.) Solve problem 4.13 from the Russell and Norvig text.

14. \*\* (20 pts.) In this problem, assume that the search space is a tree (the extension to graphs is straightforward) in which every arc is of unit cost. Assume that  $f(n) = g(n) + h(n)$  (where  $f$ ,  $g$ , and  $h$  have the usual meaning). Consider the following algorithm: Algorithm  $A_d$ :

1. Set  $c=1$ ; this is the current depth cutoff.
2. Initialize the list  $L$  of nodes that have been generated so far (but have not yet been expanded) to contain the start node  $s$ .
3. If  $L$  is empty, increment  $c$  and return to step 2.  
Let  $n$  be the first node on  $L$ .
4. If  $n$  is a goal node, stop and return the path from  $s$  to  $n$ .
5. Otherwise, remove  $n$  from  $L$ . Add to the front of  $L$ , every child  $m$  of  $n$  for which  $f(m)$  is less than or equal to  $c$ . Return to step 3.

(a) (5 pts.) Prove that the memory required by the algorithm  $A_d$  is LINEAR in the depth of the shallowest goal node.

(b) (15 pts.) Prove that the algorithm  $A_d$  is admissible when using an admissible  $h$  function.

15. \*\* (20 pts.) Prove the following results:

- (a) If the heuristic function  $h(n)$  used by  $A^*$  is monotone, if  $A^*$  selects a partial path  $(s \cdots n)$  for expansion,  $g(n) = g^*(n)$ .
- (b) If the heuristic function  $h(n)$  used by  $A^*$  is monotone, the  $f$  values of partial paths expanded by  $A^*$  form a non-decreasing sequence.