Homework 1: Blind and Heuristic Search

Due: 5:00 PM, Feb 9, 2009

Contents

1 Maze **1**

2 Admissible **2**

3 Cats and Mice **2**

4 Weighted A* **3**

5 Beam Search **3**

6 Greedy Search **3**

7 Towers of Hanoi **3**

Objectives

By the end of this homework, you will know:

1. beam search, greedy search, and A* search
2. what an admissible heuristic is

By the end of this homework, you will be able to:

1. formalize a puzzle as a search problem
2. trace blind and heuristic search algorithms

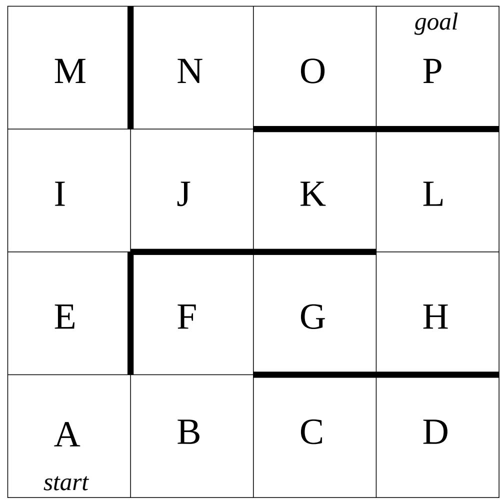
Practice

1 Maze

In the following maze, boldface lines indicate impassable walls. Search for a path from the start state to the goal state visiting neighbors in the following order: EAST, WEST, NORTH, SOUTH.

(a) In what order does DFS visit the cells in this grid? Is DFS optimal?

(b) In what order does BFS visit the cells in this grid? Is BFS optimal?



2 Admissible

Which of the following are admissible, given admissible heuristics h_1, h_2?

- $h(n) = \min\{h_1(n), h_2(n)\}$
- $h(n) = wh_1(n) + (1-w)h_2(n)$, where $0 \leq w \leq 1$
- $h(n) = \max\{h_1(n), h_2(n)\}$

Problems

3 Cats and Mice

Three cats and three mice are on one side of a river, along with a row boat that is capable of transporting either 1 or 2 occupants. We seek a way to move all the animals to the opposite side of the river, without ever allowing the number of cats on a river bank to exceed the number of mice—otherwise the cats would eat the mice. (Imagine that cats and mice can control a row boat.)

(a) State this problem as a formal search problem in terms of $\langle Q, S, F, \delta \rangle$. You need not define the transition function δ in full detail; instead list the possible operators which induce transitions: *e.g.*, 1 cat crosses the river in the boat, 2 cats cross the river in the boat, etc.

(b) Solve the cat-and-mouse problem by drawing the search space, and finding an optimal path to a goal. Limit the size of the search by disallowing moves that revisit past states: *e.g.*, initially, 1 cat can cross the river, but his return is not allowed, since the initial state would be revisited.

4 Weighted A*

Consider the algorithm wA*, which is a variant of A* search that uses the following weighted cost function: for some $w \geq 1$,

$$f_w(n) = g(n) + wh(n)$$

As usual, g is the cost from the root node to n and h is an admissible heuristic.

(a) Prove that the goal node m^* returned by wA* search on trees is within a factor of w of the optimal goal n^*: *i.e.*, $g(m^*) \leq wg(n^*)$.

(b) The wA* algorithm uses a weighted cost function that increases the value of the heuristic function h, hoping to proceed more directly towards a goal. An alternative is to ignore g entirely, simply letting $f = h$. Give two advantages of wA* over this alternative.

(c) An *anytime* algorithm is one whose solution quality improves over time, but can nonetheless be interrupted to return a (perhaps suboptimal) solution at any time. A* search is not an anytime algorithm. Design an anytime search algorithm based on wA*.

5 Beam Search

The pseudocode in Table 1 implements **beam** search (on trees), a memory-bounded heuristic variant of breadth-first search.

(a) Give the time and space complexity of beam search in terms of branching factor b, depth d, and beam width w.

(b) Argue whether or not beam search is optimal and complete.

6 Greedy Search

Beam search with beam width 1 is called **greedy** search. Give an example of a search problem in which the behaviors of greedy search, beam search (with beam width strictly greater than 1), and best-h search all differ.

7 Towers of Hanoi

Towers of Hanoi: given a stack of n disks of distinct sizes on a peg with no larger disk on top of a smaller disk, move the disks one at a time from one peg to a second goal peg, making use of a third peg when necessary, but never stacking a larger disk on top of a smaller disk.

(a) Represent the Towers of Hanoi problem for $n = 2$ as a formal search problem by drawing the search space and highlighting the start and final states.

(b) State whether the following heuristics are admissible. Where admissible, state why, and where inadmissible, give counterexample. [Hint: The optimal number of moves from the start state is $2^n - 1$].

BEAM$(X, S, G, \delta, c, w, h)$	
Inputs	search problem
	beam width w
	heuristic h
Output	(path to) goal node
Initialize	$O = S$ is the list of open nodes
	P is the beam (*i.e.*, priority queue)

```
while (O is not empty) do

    P = O, O = ∅

    while (P is not empty) do

      1. delete node n ∈ P s.t. f(n) is minimal
      2. if n ∈ G, return (path to) n
      3. for all m ∈ δ(n)
         (a) compute h(m)
         (b) g(m) = g(n) + c(m, n)
         (c) f(m) = g(m) + h(m)
         (d) insert node m into O with priority f(m)
         (e) truncate O to maximum beam width w

fail
```

Table 1: Beam Search.

1. the number of disks on a peg other than the goal peg
2. the number of disks on top of the largest disk
3. $2^k - 1$, where k is the largest misplaced disk
4. 2^{k-1}, where k is the largest misplaced disk