2-11-04

Informed Search Strategies

The node we select for expansion is chosen according to some evaluation function f(n).

Best-First-Search assumes f(n) is a cost and always expands node for which f(n) is smallest.

Problem: How to choose f?

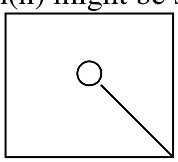
A Key component of f is usually some heuristic function.

H(n) := estimated cost from n to the goal.

Example:

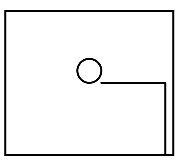
In maze program h(n) might be straight_line distance (straight line mathematical distance from current position to the goal)

Example



Manhattan distance (straight line_distance)

Must travel in STRAIGHT lines to get to final destination.



Example: Can use straight line with 8 puzzle.

7	2	1
5		6
8	3	4

For each piece, compute straight line distance to where it needs to be. Add up these values.

1 needs to move 1 space left,
2 needs to move 1 space right
3 needs to move one space up, then one left ….
4 needs to move up 1, and left 1
5 needs to move 2 right
6 needs to move 1 down, and 2 left
7 needs to move 2 down and 1 right
8 needs to move 2 right
total = 1 + 1 + 2 + 2 + 2 + 3 + 3 + 2 = 16

Greedy Best First Search

Choose f(n) = h(n)
Example from previous 8 puzzle

	1	8
2	3	4
5	6	7

H(puzzle) = 12

Take H(next possible boards), looking for the smallest cost, and proceed with that solution.

Problem with greedy best first search can get stuck with the same moves back and forth yielding the lowest cost.

A* Search

Choose f(n) = g(n) + h(n) and do best first earch

Here g(n) = the cost to reach node n
i.e. f(n) is estimated cost of total solution.

Turns out A* search is complete
i.e., given enough resources it will find a solution.

It is also in some sense optimal.
(among best first search algorithms)
& Provided h satisfies some constraints.

Call h admissable if it never overestimates the cost to a solution.

Proof of optimality

Suppose a suboptimal goal node G appears of fringe of nodes considered so far. (suboptimal means, there exists a goal G to go to).
Want to argue G won't be picked to expand next and hence see it's a solution).
Let C* be the cost of the optimal path.

So
F(G) = g(G) + h(G)
F(G) >= g(G) > C* (assuming costs are always positive, bigger than C* because we are assuming G is not optimal)
Since for any node n f(n) does not overestimate cost of solution through n.
This means there must be some n on fringe which is on path that gives C*. For this n
F(n) = g(n) + h(n) <= C* (Since h is an underestimate)
SO
F(n) <= C* < f(g)
So n will be chosen over g to be expanded next.