Heuristic Search: BestFS and A*

Computer Science cpsc322, Lecture 8

(Textbook Chpt 3.6)

January, 20, 20010

Course Announcements

Posted on WebCT

- Second Practice Exercise (uninformed Search)
- Assignment 1

DEPARTMENT OF COMPUTER SCIENCE

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Distinguished Lecture Series 2008 - 2009

Speaker: Michael Littman Rutgers University

Date: Thursday, January 22, 2009 Time: 3:30 - 4:50pm

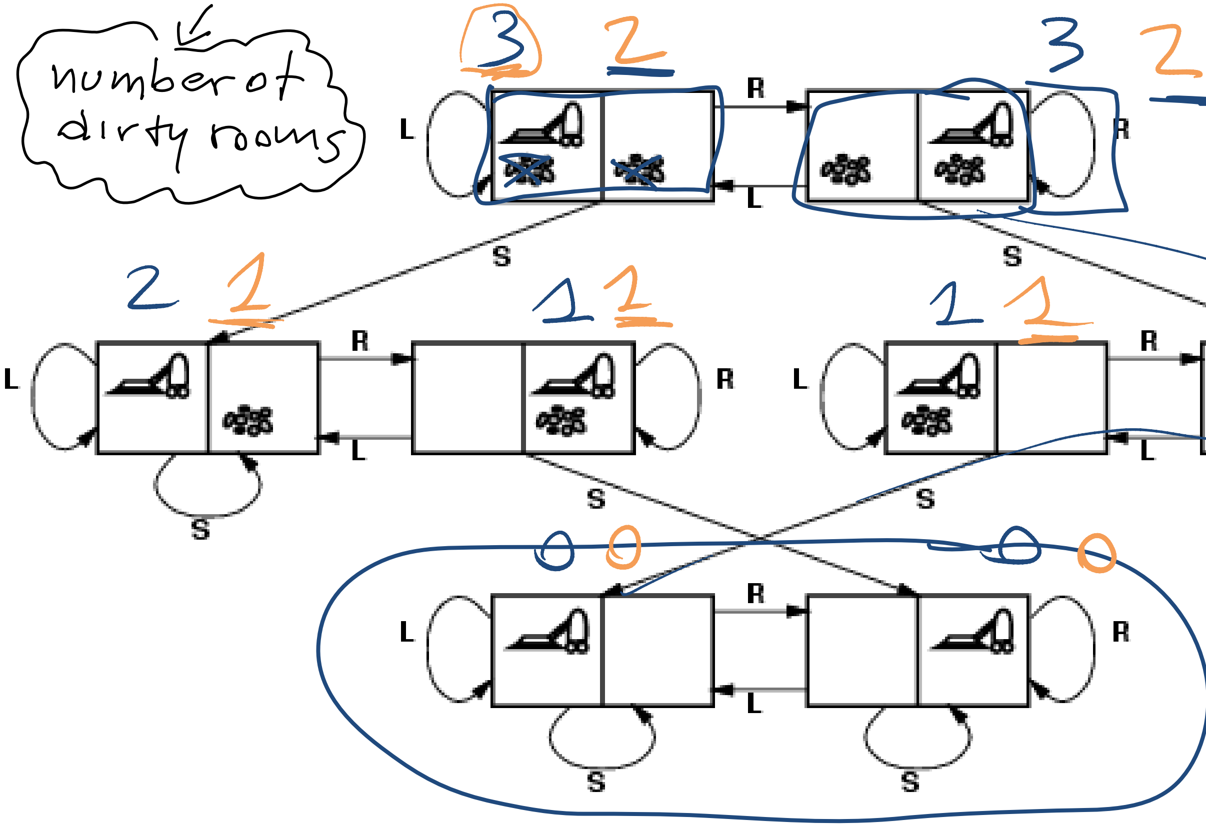
Venue: Hugh Dempster Pavilion Room 310

Title: Efficiently Learning to Behave Efficiently
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

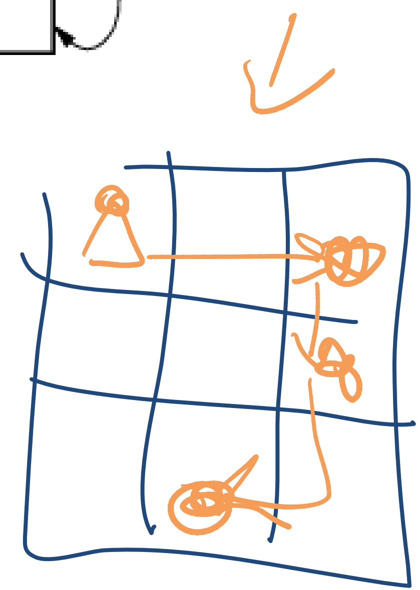
Lecture Overview

- Recap Heuristic Function
- Best First Search
- A*

Admissible heuristic for Vacuum world?



- cost of optimal solution in blue
- value of admissible heuristic in orange



states? Where it is dirty and robot location

actions? *Left*, *Right*, *Suck*

Possible goal test? no dirt at all locations

Lecture Overview

- Recap Heuristic Function
- Best First Search
- A*

Best-First Search

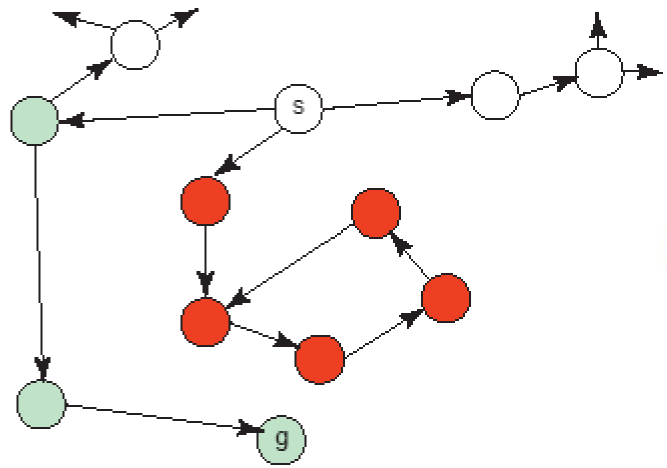
- **Idea:** select the path whose end is closest to a goal according to the heuristic function.

- **Best-First search** selects a path on the frontier with minimal *h*-value (for the end node).

- It treats the frontier as a priority queue ordered by *h*. (similar to ?) LCFS → Cost
- This is a greedy approach: it always takes the path which appears locally best

Analysis of Best-First Search

- Complete no: a low heuristic value can mean that a cycle gets followed forever.

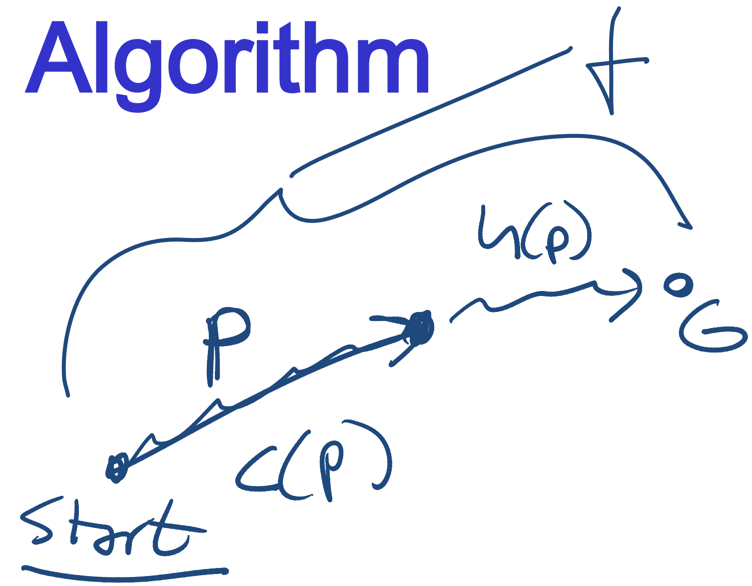


- Optimal: no (why not?)
- Time complexity is $O(b^m)$
- Space complexity is $O(b^m)$

Lecture Overview

- Recap Heuristic Function
- Best First Search
- A* Search Strategy

A^* Search Algorithm



- A^* is a mix of:
  - **lowest-cost-first** and
  - **best-first search**

- A^* treats the frontier as a priority queue ordered by $f(p) = c(p) + h(p)$

- It always selects the node on the frontier with the lowest estimated total distance.

Analysis of A^*

Let's assume that arc costs are strictly positive.

- Time complexity is $O(b^m)$
  - the heuristic could be completely uninformative and the edge costs could all be the same, meaning that A^* does the same thing as BFS
- Space complexity is $O(b^m)$ like BFS, A^* maintains a frontier which grows with the size of the tree
- Completeness: yes.
- Optimality: yes.

Optimality of A^*

If A^* returns a solution, that solution is guaranteed to be optimal, as long as

When

- the branching factor is finite
- arc costs are strictly positive
- *h(n)* is an underestimate of the length of the shortest path from *n* to a goal node, and is non-negative

admissible

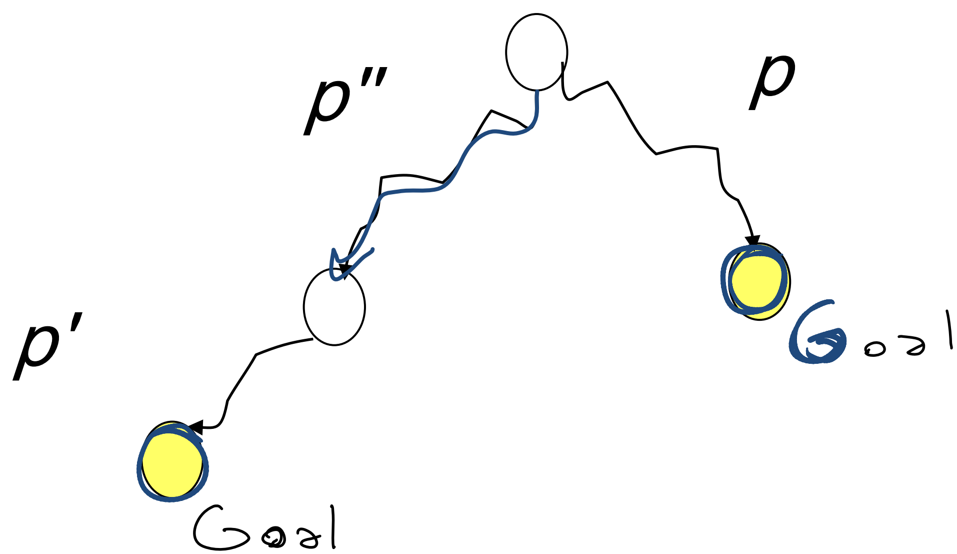
Theorem

If A^* selects a path *p*, *p* is the shortest (i.e., lowest-cost) path.

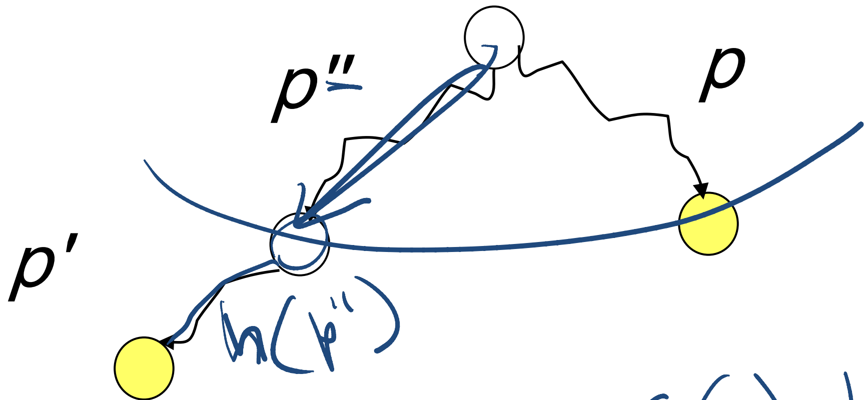
Why is A^* optimal?

$f = c + h$

- Assume for contradiction that some other path p' is actually the shortest path to a goal $cost(p') < cost(p)$
- Consider the moment when p is chosen from the frontier. Some part of path p' will also be on the frontier; let's call this partial path p''.



Why is A^* optimal? (cont')



$\underbrace{c(p)+h(p)}_{f(p)} \le \underbrace{c(p'')+h(p'')}_{f(p'')}$

- Because p was expanded before p'', $f(p) \le f(p'')$
- Because p is a goal, $h(p)=0$ Thus $\boxed{c(p) \le c(p'')+h(p'')}$
- Because h is admissible, $\boxed{cost(p'') + h(p'') \le cost(p')}$ for any path p' to a goal that extends p''
- Thus $cost(p) \le cost(p')$ for any other path p' to a goal.

This contradicts our assumption that p' is the shortest path.

that $cost(p') < cost(p)$

Optimal efficiency of A^*

next class

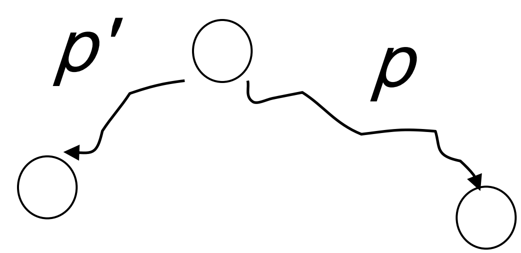
- In fact, we can prove something even stronger about A^*: in a sense (given the particular heuristic that is available) **no search algorithm could do better!**

- Optimal Efficiency: Among all optimal algorithms that **start from the same start node** and **use the same heuristic** h, A^* expands the minimal number of paths.

Why is A^* optimally efficient?

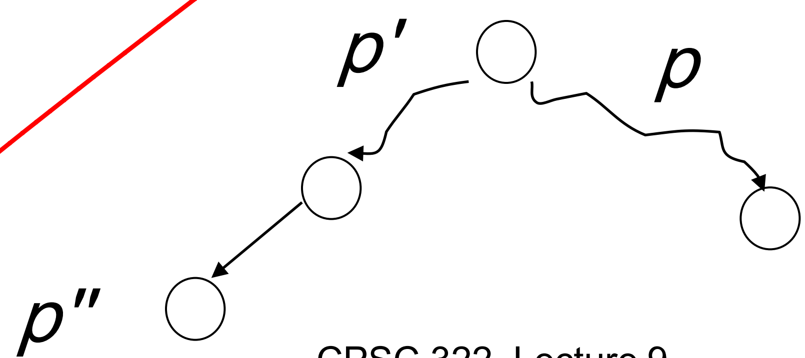
Theorem: A^* is optimally efficient.

- Let f^* be the cost of the shortest path to a goal.
- Consider any algorithm A'
  - the same start node as A^*,
  - uses the same heuristic
  - fails to expand some path p' expanded by A^*, for which $f(p') < f^*$.
- Assume that A' is optimal.



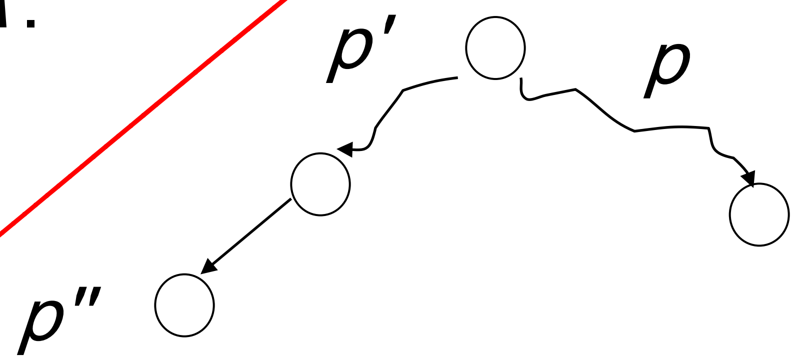
Why is A^* optimally efficient? (cont')

- Consider a different search problem
  - identical to the original
  - on which h returns the same estimate for each path
  - except that p' has a child path p'' which is a goal node, and the true cost of the path to p'' is $f(p')$.
  - that is, the edge from p' to p'' has a cost of $h(p')$: the heuristic is exactly right about the cost of getting from p' to a goal.



Why is A^* optimally efficient? (cont')

- A' would behave identically on this new problem.
  - The only difference between the new problem and the original problem is beyond path p', which A' does not expand.
- Cost of the path to p'' is lower than cost of the path found by A'.



- This violates our assumption that A' is optimal.

Learning Goals for today's class

- Define/read/write/trace/debug different search algorithms
    - With / Without cost
    - Informed / Uninformed

they use h

Best First $\min h$

A* min c+h

- Formally prove A* optimality.

- Define optimally efficient and formally prove that A* is optimally efficient

to be done

Next class

Finish Search (finish Chpt 3)

- Branch-and-Bound
- A* enhancements
- Non-heuristic Pruning
- Backward Search
- Dynamic Programming