Heuristic Search: BestFS and A*

Computer Science cpsc322, Lecture 8

(Textbook Chpt 3.6)

Sept, 21, 2010

Department of Computer Science
Undergraduate Events
More details @
https://www.cs.ubc.ca/students/undergrad/life/upcoming-events

Resume Drop-in Editing Session
Mon. Sept 24, 12 – 3 pm
Rm 253, ICICS/CS Bldg.

Global Relay Info Session
Mon. Sept 24, 5:30 pm
DMP 110

Google Info Session
Tues. Sept 25, 5:30 pm
Wesbrook 100

UBC Career Days
Wed. Sept 26 & Thurs. Sept 27
10 am – 4 pm
SUB

Microsoft Info Session
Wed. Sept 26, 5:30 pm
Wesbrook 100

Tri-mentoring Kick Off (invitation only)
Thurs. Sept 27, 5:30 pm
X860, ICICS/CS Bldg.

Avanade Info Session
Fri. Sept 28, 12 – 2 pm
Rm 206, ICICS/CS Bldg.

ACM Programming Team Try Out
Sat. Sept 29, 12 – 5 pm
Rm 005, Rm 011, ICICS/CS Bldg.

Course Announcements

Marks for Assignment0: posted on Connect

Assignment1: posted

If you are confused on basic search algorithm, different search strategies….. Check learning goals at the end of lectures. Work on the Practice Exercises and Please come to office hours

Giuseppe	Tue 2 pm, my office.
Nathaniel Tomer	Fri 11am, X150 (Learning Center)
Tatsuro Oya	**Wed 2 pm, X150 (Learning Center)**
Mehran Kazemi	Mon 11 am, X150 (Learning Center)

Course Announcements

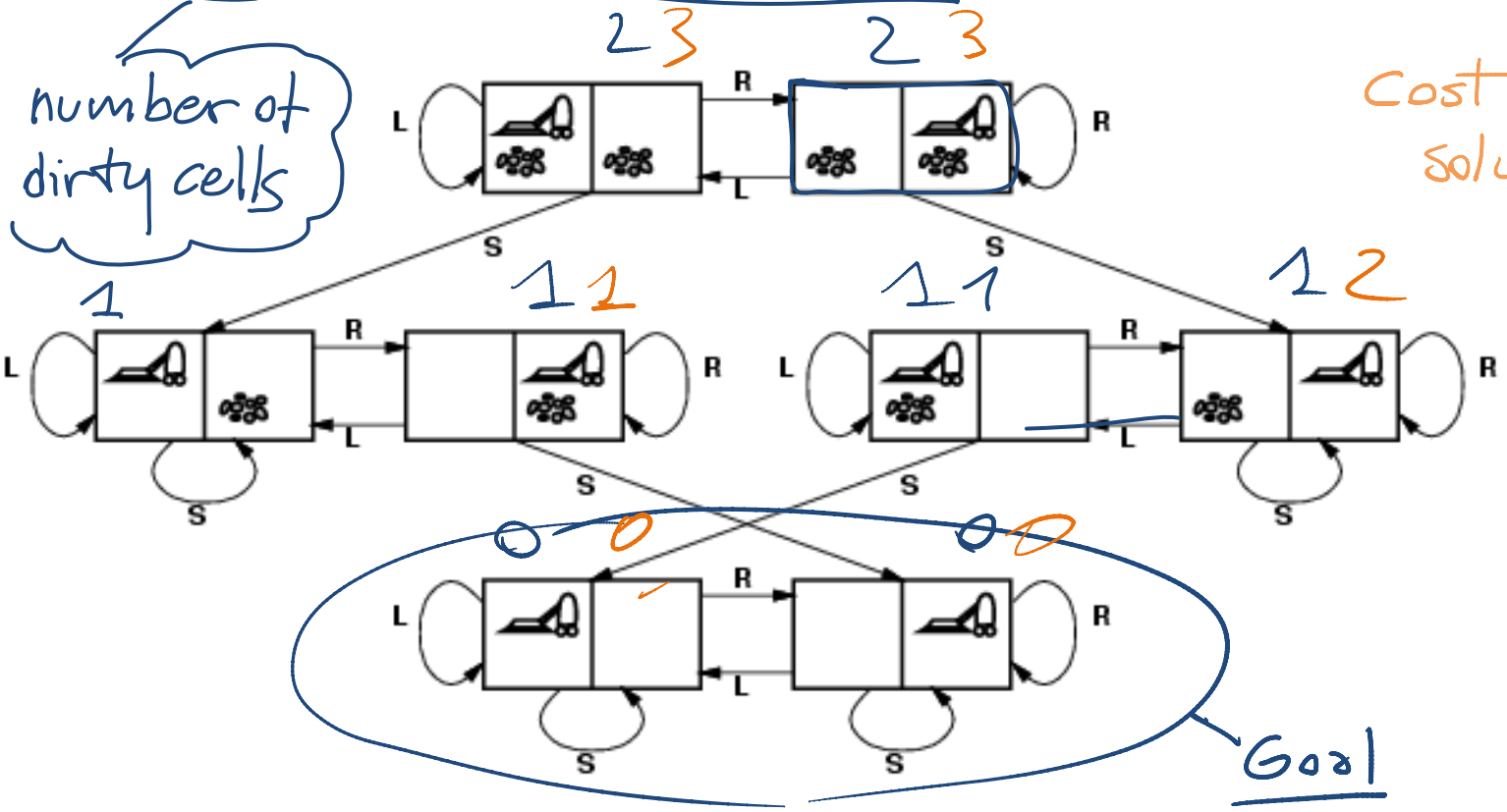
Inked Slides

- At the end of each lecture I revise/clean-up the slides. Adding comments, improving writing… make sure you check them out

Lecture Overview

- Recap Heuristic Function
- Best First Search
- A*

Admissible heuristic for Vacuum world?



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

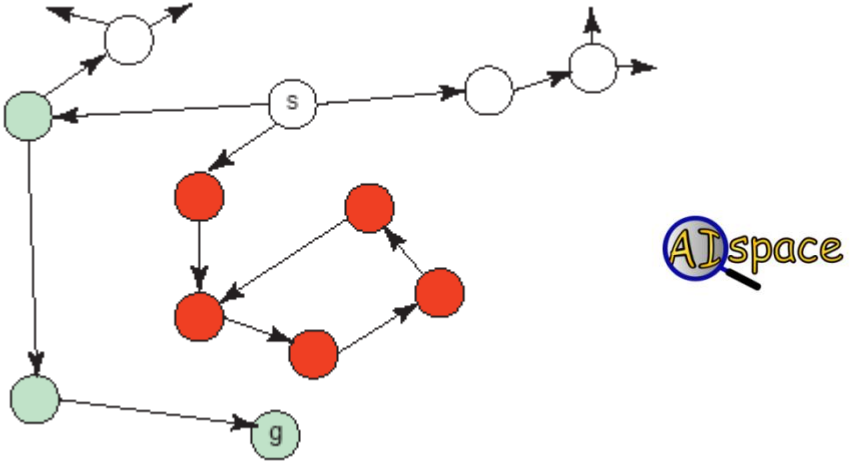
- **Idea:** select the path whose end is closest to a goal according to the heuristic function.

- **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 by 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 $O(b^m)$

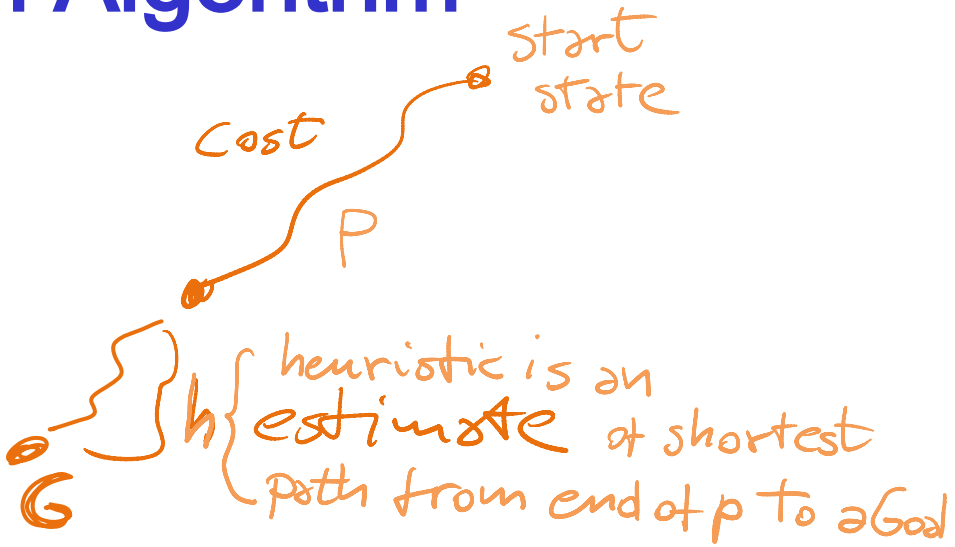
AIspace

worst case

Lecture Overview

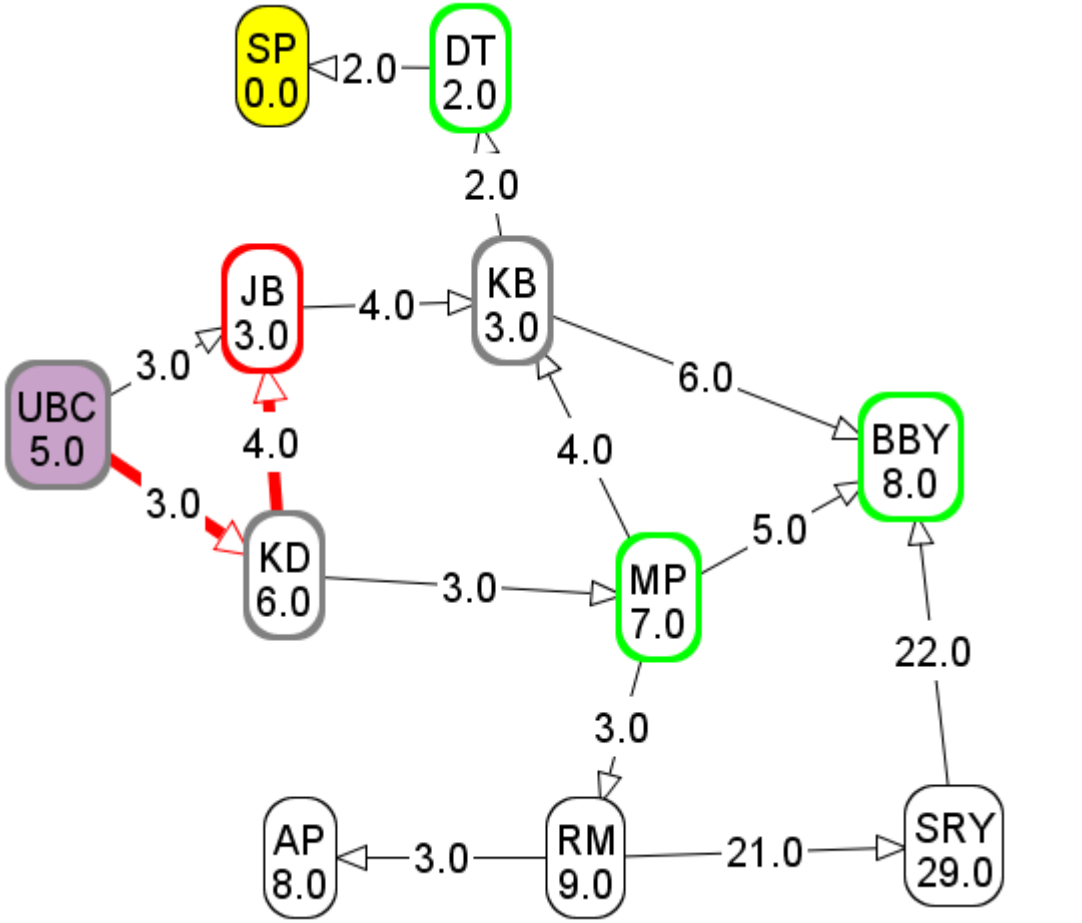
- Recap Heuristic Function
- Best First Search
- A* Search Strategy

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 $f(p) =$ cost(p) + h(p) is an estimate
- It always selects the node on the frontier with the lowest ..... estimated ..... total ..... distance.

Computing f-values



f-value of UBC → KD → JB?

6 9 10 11

Analysis of A^*

for all states heuristic is equal to 0

Let's assume that arc costs are strictly positive.

- Time complexity is $O(b^m)$ $\forall s \; h(s) = 0$
  - the heuristic could be completely uninformative and the edge costs could all be the same, meaning that A^* does the same thing as…. DFS BFS LCFS
- Space complexity is $O(b^m)$ like BFS, A^* maintains a frontier which grows with the size of the tree
- Completeness: yes.
- Optimality: ??

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 as the solution,

p is the shortest (i.e., lowest-cost) path.

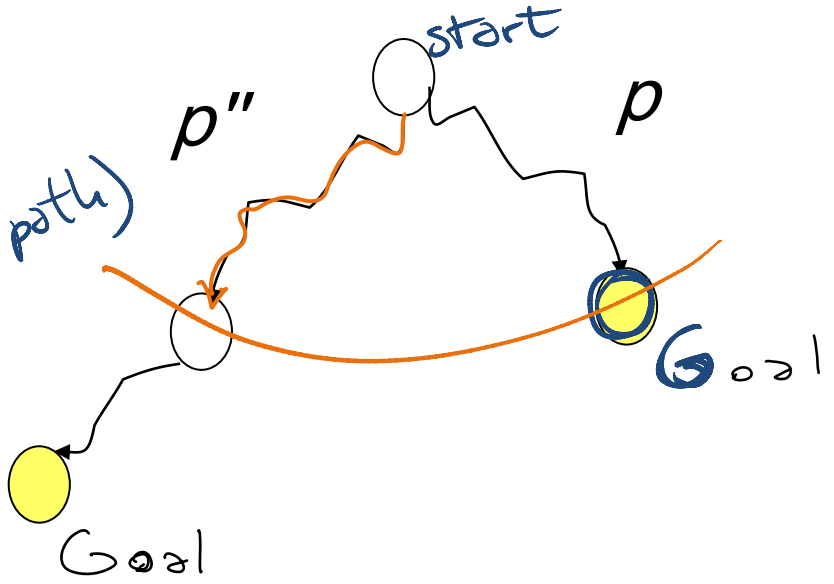
Why is A^* optimal?

- A* returns p
- Assume for contradiction that some other path p' is actually the shortest path to a goal
- 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''.

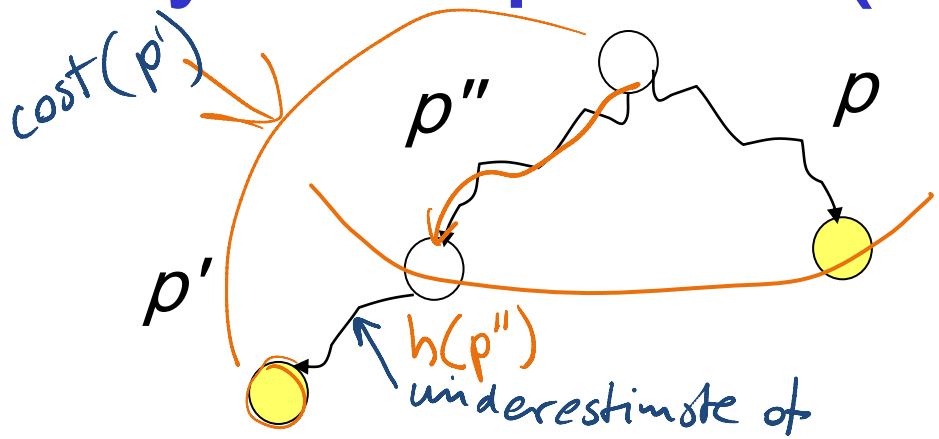
cost(p) > cost(p')

Think!

for any path from start to any state there is always a subpath (of that path) on the frontier



Why is A^* optimal? (cont')



$cost(p) + h(p) \leq cost(p'') + h(p'')$

$f(p) \leq f(p'')$

- Because p was expanded before p'',
- Because p is a goal, $h(p) = 0$ Thus $cost(p) \leq cost(p'') + h(p'')$ (1)
- Because h is admissible, $cost(p'') + h(p'') \leq cost(p')$ for any path p' to a goal that extends p'' (2)

combining (1) and (2)

- Thus $cost(p) \leq cost(p')$ for any other path p' to a goal.

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

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

Optimal efficiency of A^*

- 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.

Sample A* applications

- **An Efficient A* Search Algorithm For Statistical** Machine Translation. 2001
- **The Generalized A* Architecture**. Journal of Artificial Intelligence Research (2007)
  - Machine Vision … Here we consider a new compositional model for finding salient curves.
- **Factored A*search for models over sequences and trees** International Conference on AI. 2003…. It starts saying… *The primary challenge when using A* search is to find heuristic functions that simultaneously are admissible, close to actual completion costs, and efficient to calculate…* applied to NLP and BioInformatics

(Natural Language Processing)

Sample A* applications (cont')

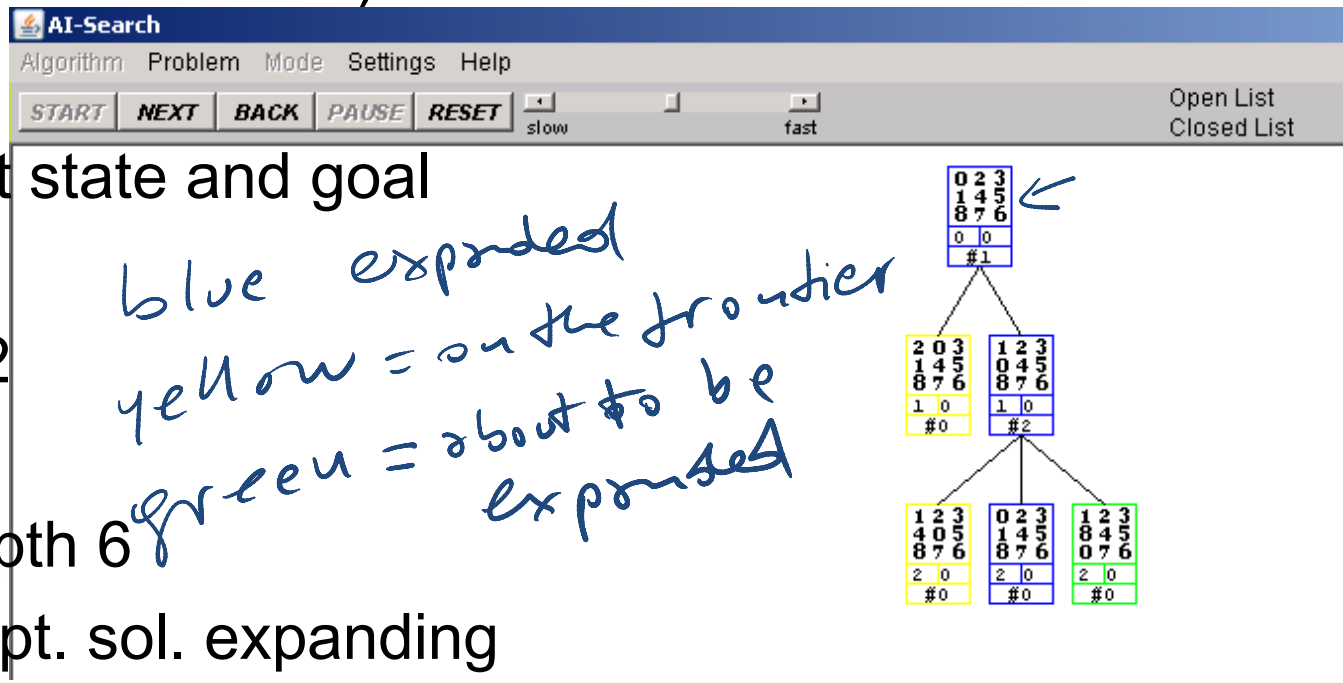
Aker, A., Cohn, T., Gaizauskas, R.: **Multi-document summarization using A* search and discriminative training.** Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing.. ACL (2010)

*DFS, BFS, A** Animation Example

- The AI-Search animation system

http://www.cs.rmit.edu.au/AI-Search/Product/

- *To examine Search strategies when they are applied to the 8puzzle*
- Compare only DFS, BFS and A* (with only the two heuristics we saw in class)



- With default start state and goal
- DFS will find

Solution at depth 32

- BFS will find

Optimal solution depth 6

- A* will also find opt. sol. expanding much less nodes

nPuzzles are not always solvable

Half of the starting positions for the *n*-puzzle are impossible to resolve (for more info on 8puzzle)
http://www.isle.org/~sbay/ics171/project/unsolvable

- So experiment with the AI-Search animation system with the default configurations.
- If you want to try new ones keep in mind that you may pick unsolvable problems

Learning Goals for today's class

- **Define/read/write/trace/debug** & **Compare** different search algorithms
  - With / Without cost
  - Informed / Uninformed
- Formally prove A* optimality.

Next class

Finish Search (finish Chpt 3)

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