Search: A^* Optimal Efficiency

CPSC 322 Lecture 8

September 21, 2007
Textbook §3.6

Lecture Overview

1. Recap
2. Optimal Efficiency of A^*

A^* Search Algorithm

- A^* is a mix of lowest-cost-first and Best-First search.
- It treats the frontier as a priority queue ordered by $f(p) = cost(p) + h(p)$.
- 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.

- Completeness: yes.
- Time complexity: $O(b^m)$
  - 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: $O(b^m)$
  - like BFS, A^* maintains a frontier which grows with the size of the tree
- Optimality: yes.

Optimality[1] of A^*

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

- 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

[1] Some literature, and the textbook, uses the word "admissiblity" here.

Why is A^* optimal?

Theorem

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

- Assume for contradiction that some other path p' is actually the shortest path to a goal
- Consider the moment just before p is chosen from the frontier. Some part of path p' will also be on the frontier; let's call this partial path p''.
- Because p was expanded before p'', $f(p) \leq f(p'')$.
- Because p is a goal, $h(p) = 0$. Thus $cost(p) \leq cost(p'') + h(p'')$.
- Because h is admissible, $cost(p'') + h(p'') \leq cost(p')$ for any path p' to a goal that extends p''
- Thus $cost(p) \leq cost(p')$ for any other path p' to a goal. This contradicts our assumption that p' is the shortest path.

Lecture Overview

1. Recap
2. Optimal Efficiency of A^*

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.
  - problem: A^* could be unlucky about how it breaks ties.
  - So let's define optimal efficiency as expanding the minimal number of paths p for which $f(p) \neq f^*$, where f^* is the cost of the shortest path.

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' which has the same start node as A^*, uses the same heuristic and fails to expand some path p' expanded by A^* for which $cost(p') + h(p') < f^*$. Assume that A' is optimal.
- Consider a different search problem which is identical to the original and 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.
- 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.