Improving Search

1/29/16
Reading Quiz

Question 1: IDA* combines the advantages of A* and ________ searches.

a)  breadth first
b)  depth first

c)  uniform cost
d)  greedy
Reading Quiz

Question 2: Branch and Bound combines the advantages of A* and ________ searches.

a) breadth first  

b) depth first  

c) uniform cost  

d) greedy
Devising Heuristics (from Wednesday)

- Must be **admissible**: never overestimate the cost to reach the goal.
- Should strive for **consistency**: $h(s) + c(s)$ non-decreasing along paths.
- The higher the estimate (subject to admissibility), the better.

Key idea: simplify the problem.

- Traffic Jam: ignore some of the cars.
- Path Finding: assume straight roads.
Exercise

Devise a heuristic for the 8-puzzle game.
Why is $A^*$ complete and optimal?

- Let $C^*$ be the cost of the optimal solution path.
- $A^*$ will expand all nodes with $c(s) + h(s) < C^*$.
- $A^*$ will expand some nodes with $c(s) + h(s) = C^*$ until finding a goal node.
- With an admissible heuristic, $A^*$ is optimal because it can’t miss a better path.
- Given a positive step cost and a finite branching factor, $A^*$ is also complete.
Why is A* optimally efficient?

- For any given admissible heuristic, no other optimal algorithm will expand fewer nodes.

- Any algorithm that does NOT expand all nodes with $c(s) + h(s) < C^*$ runs the risk of missing the optimal solution.

- Only possible difference could be in which nodes are expanded when $c(s) + h(s) = C^*$. 
Iterative Deepening

- Inherits the completeness and shortest-path properties from BFS.
- Requires only the memory complexity of DFS.

Idea:

- Run a depth-limited DFS.
- Increase the depth limit if goal not found.
IDA*; Branch and Bound

- Use DFS, but with a bound on $c(s) + h(s)$.
- If bound < $c(goal)$, the search will fail and we’ll have to increase the bound.
  - IDA* starts with a low bound and gradually increases it.
- If bound > $c(goal)$, we may find a sub-optimal solution
  - We can re-run with $c(solution) - \epsilon$ as the new bound
  - Branch and bound starts with a high bound and lowers it each time a solution is found.
- We can alternate these two to narrow in on the right bound.
- With reasonable bounds, these will explore an asymptotically similar number of nodes to A*, with a lower memory overhead.
Multiple simultaneous searches

Bidirectional
Multiple simultaneous searches

Island-Driven
Multiple simultaneous searches

Hierarchy of Abstractions
Dynamic Programming

- Key idea: cache intermediate results.
- Applicable to much more than just state space search.
- The book glosses over its complexity.
  - Size of the state space graph IS NOT the right problem size.
- We’ll come back to this when we talk about MDPs (reinforcement learning).
Exercise: trace A*

Use the Manhattan distance heuristic.