1 handout: slides
Beyond A∗
Greedy Best-first Search (GBFS)

\[ Q \leftarrow \text{an ordered list containing just the initial state.} \]

Loop

If \( Q \) is empty,

then return failure.

\( Node \leftarrow \text{Pop}(Q). \)

If \( Node \) is a goal,

then return \( Node \) (or path to it)

else

\( Children \leftarrow \text{Expand (Node).} \)

Merge \( Children \) into \( Q \), keeping \text{sorted by heuristic}.\]
GBFS on the 8-puzzle

$h(n) = \text{number of tiles out of place. (The blank is not a tile.)}$

\[
\begin{array}{ccc}
2 & 8 & 3 \\
1 & 2 & 3 \\
7 & \Box & 5 \\
\end{array}
\]

Start state: \hspace{1cm} Goal state: \hspace{1cm}

\[
\begin{array}{ccc}
1 & 6 & 4 \\
8 & \Box & 4 \\
7 & 6 & 5 \\
\end{array}
\]

Please draw the tree resulting from the first two node expansions.
Assume branching factor $b$ and solution at depth $d$.

Completeness:

Time:

Space:

Admissibility:
Beam Search

Truncate queue to hold the most promising $k$ nodes. $k$ is the beam width.

Works best with breadth-first search!
Suboptimal Search

- Problem Settings
- \(wA^*\)
- \(wA^*\) Behavior
- Distance-to-go
- RR-\(d\)

Anytime Search

Real-time Search

EOLQs
optimal: minimize solution cost

\[ f(n) = g(n) + h(n) < f^* \]

 greedy: minimize solving time

bounded suboptimal: minimize time subject to relative cost bound (factor of optimal)

bounded cost: minimize time subject to absolute cost bound

contract: minimize cost subject to absolute time bound

anytime: iteratively converge to optimal

utility: maximize given function of cost and time
Weighted A*

\[ f'(n) = g(n) + w \cdot h(n) \]

- nodes with high \( h(n) \) look even worse
- no infinite rabbit holes
- suboptimality bounded: within a factor of \( w \) of optimal!
optimal: uniform-cost search
optimal: A*
bounded suboptimal: Weighted A*
For Speed: Distance-to-go, Not Cost-to-go

how to minimize solving time?
how to minimize solving time?
how to minimize number of expansions?
how to minimize solving time?
how to minimize number of expansions?
take the shortest path to a goal
how to minimize solving time?
how to minimize number of expansions?
take the shortest path to a goal
for domains with costs, this is not $h(n)$

new information source: $d(n)$

```
        n
       /\     /
      /  \   /  \
     h=4 d=2 h=5 d=1
```

Wheeeler Ruml (UNH)
For Speed: Distance-to-go, Not Cost-to-go

how to minimize solving time?
how to minimize number of expansions?
take the shortest path to a goal
for domains with costs, this is not $h(n)$

new information source: distance-to-go = $d(n)$

Speedy: best-first search on $d$
bounded-suboptimal using $h, \hat{h}, d$

optimal: uniform-cost
bounded-suboptimal using $h, \hat{h}, d$

optimal: $A^*$
bounded-suboptimal using $h, \hat{h}, d$

bounded suboptimal: Weighted A*
bounded-suboptimal using $h, \widehat{h}, d$

bounded suboptimal: Optimistic Search (ICAPS, 2008)
bounded-suboptimal using $h, \hat{h}, d$

bounded suboptimal: Explicit Estimation Search (IJCAI, 2011)
Anytime Search
Anytime A*

1. run weighted A*
2. keep going after finding a goal
3. keep best goal found (can test at generation)
4. prune anything with $f(n) > incumbent$

Anytime Restarting A* (ARA*): lower weight after finding each solution

Anytime EES
Break

- EOLQs
- Are We Done?
- Beyond A*
- Suboptimal Search
- Anytime Search
- Anytime A*
- Break
- Real-time Search
- EOLQs

- asst2 (asst8), asst 3
- scores and grades
Real-time Search
keep hash table of $h$ values for visited states

1. for each neighbor of current state $s$
2. either find $h$ in table or do some lookahead
3. add edge cost to get $f$
4. update $h(s)$ to second-best $f$ value
5. move to best neighbor
1. single A* lookahead (LSS)
2. update all $h$ values in LSS
3. move to frontier
Uninformed: DFS, UCS
Admissible: A*
Limited memory: iterative deepening (IDDFS, IDA*)
Satisficing: GBFS, Speedy, Beam
Bounded suboptimal: wA*, RR-\(d\)
Real-time: RTA*, LSS-LRTA*
Other Shortest-path Algorithms

- EOLQs
- Are We Done?

Beyond A*
- Suboptimal Search
  - SMA*, IE
  - RBFS
  - Bugsy
  - RTAA*

Anytime Search
- Real-time Search
  - RTA*
  - LSS-LRTA*

Search Algorithms
- Other Algorithms

Course projects!
EOLQs

Beyond A*
Suboptimal Search
Anytime Search
Real-time Search
EOLQs

EOLQs
Please write down the most pressing question you have about the course material covered so far and put it in the box on your way out.

*Thanks!*