1 handout: slides
Are We Done?

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

GBFS
8-puzzle
Evaluating Greedy
Beam Search

Suboptimal Search

Anytime Search

Real-time Search

EOLQs
Greedy Best-first Search (BGFS)

Q ← an ordered list containing just the initial state.

Loop
  If Q is empty,
    then return failure.
  Node ← Pop(Q).
  If Node is a goal,
    then return Node (or path to it)
  else
    Children ← Expand (Node).
    Merge Children into Q, keeping sorted by heuristic.
GBFS on the 8-puzzle

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

\[
\begin{array}{ccc}
2 & 8 & 3 \\
\hline
1 & 2 & 3 \\
\end{array}
\]

Start state: 1 6 4 Goal state: 8 □ 4
7 □ 5
7 6 5

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

Completeness:

Time:

Space:

Admissibility:
Truncate queue to hold the most promising $k$ nodes. $k$ is the beam width.
Suboptimal Search

- Problem Settings
- wA*
- wA* Behavior
- Distance-to-go
- EES

Anytime Search

Real-time Search

EOLQs
optimal: minimize solution cost
        suffer all with $f(n) = g(n) + h(n) < f^*$

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
wA* Behavior

- Are We Done?
- Beyond A*

Suboptimal Search
- Problem Settings
- wA*
- wA* Behavior
  - Distance-to-go
  - EES

Anytime Search

Real-time Search

EOLQs

optimal: A*
bounded suboptimal: Weighted A*
how to minimize solving time?
how to minimize solving time?
how to minimize number of expansions?
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 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 = $d(n)$

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

Wheeler 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: $d(n) = d(n)$

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

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

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

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

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

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

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

- asst2
- scores and grades
- AAAI
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
Search Algorithms

- Are We Done?
- Beyond A*
- Suboptimal Search
- Anytime Search
- Real-time Search
  - RTA*
  - LSS-LRTA*
- Search Algorithms
- Other Algorithms
- EOLQs

Uninformed: DFS, UCS
Admissible: A*
Limited memory: iterative deepening (IDDFS, IDA*)
Satisficing: GBFS, Speedy, Beam
Bounded suboptimal: wA*, EES
Real-time: RTA*, LSS-LRTA*
Other Shortest-path Algorithms

- Are We Done?
- Beyond A*
- Suboptimal Search
- Anytime Search
- Real-time Search
- RTA*
- LSS-LRTA*
- Search Algorithms
- Other Algorithms
- SMA*, IE
- RBFS
- Bugsy
- RTAA*

Course projects!
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!