Best-First Search: The A* Algorithm

Keith L. Downing

August 22, 2017
Search is..

A (preferably methodical) process for finding something.

- Searching for:
  - pre-existing entities (information, objects, etc.)
  - strategies for creating/designing entities.
- Examples:
  - Web search (for just about anything)
  - AI search - creates something, doesn’t just find it.
  - Web search for AI search algorithms.
- Uninformed -vs- Informed: Do points in the search space give information that helps the searcher to determine the next step?
- Partial -vs- Complete solutions (i.e., attempts): Could the current state of search always be considered a complete solution, though not necessarily good or optimal? Or is it often a partial state that must be incrementally enhanced to become a solution?
  - E.g. 2 approaches to origami.
  - Closed-loop -vs- Open-loop control
The problem-solver works with partial -vs- complete solutions.
For this puzzle, a complete solution is a sequence of moves that achieves the goal configuration. Typically, this is solved using incremental methods that add one move at a time to a partial sequence.
The Horizon of a Search Graph

Root Node Contains Starting Search State
Expansion = Generating Child Nodes

Open, Unexpanded
Closed, Expanded

Keith L. Downing
Best-First Search: The A* Algorithm
3 Common Incremental Search Algorithms

 Depth-First

 Breadth-First

 Best-First

 General

 Problem-Specific

 "Intelligent"

 Keith L. Downing

 Best-First Search: The A* Algorithm
The key decision that differentiates depth-first, breadth-first and best-first search is: *What node on the OPEN list should be expanded next?*

- **Depth-First:** OPEN = a stack (LIFO - *Last in, first out*)
- **Breadth-First:** OPEN = a queue (FIFO - *First in, first out*)
- **Best-First:** OPEN = a sorted list known as the **agenda**, where sorting is based on the node’s **evaluation**, which reflects the algorithm’s knowledge about that search state and its potential to be extended into an optimal solution.
Needed a good route-planning algorithm.


Still used by the vehicle-navigation industry.

https://www.youtube.com/watch?v=mQ7M-zhiu7U
The Evaluation Function in Best-First Search

\[ f(x) = g(x) + h(x) \]

Concrete Estimate

Keith L. Downing
Best-First Search: The A* Algorithm
The heuristic function estimates the distance (typically in terms of the number of moves) from the current state to the goal.
DEFINE best-first-search()

1. CLOSED ← ∅; OPEN ← ∅  
2. Generate the initial node, n0, for the start state.  
3. g(n0) ← 0; Calculate h(n0)  
4. f(n0) ← g(n0) + h(n0);  
5. Push n0 onto OPEN.  
6. Do Agenda Loop
The Agenda Loop (Open List = Agenda)

- If OPEN = ∅ return FAIL
- X ← pop(OPEN)
- push(X,CLOSED)
- If X is a solution, return (X, SUCCEED)
- SUCC ← generate-all-successors(X)
- For each S ∈ SUCC do:
  - If node S* has previously been created, and if state(S*) = state(S), then S ← S*.
  - push(S,kids(X))
  - If not(S ∈ OPEN) and not(S ∈ CLOSED) do:
    - attach-and-eval(S,X)
    - insert(S,OPEN) ;; OPEN is sorted by ascending f value.
  - else if g(X) + arc-cost(X,S) < g(S) then (found cheaper path to S):
    - attach-and-eval(S,X)
    - If S ∈ CLOSED then propagate-path-improvements(S)
Node Expansion in A*

Expand D

Agenda
(Open Nodes)

Closed Nodes

Keith L. Downing  
Best-First Search: The A* Algorithm
Updating g values in A*

Before

- $P_1$
  - $g = 4$  
  - $h = 20$

- $C_1$
  - $g = 5$  
  - $h = 19$

- $C_2$
  - $g = 6$  
  - $h = 18$

After

- $P_2$
  - $g = 2$  
  - $h = 22$

- $C_1$
  - $g = 3$  
  - $h = 19$

- $C_2$
  - $g = 4$  
  - $h = 18$
An admissible heuristic NEVER overestimates distance to the goal.

**Admissable Heuristics for A***

- $h(v) \leq 2$
- $f(v) \leq 3 + 2$
- $f(w) = 5 + 0$

- $h(x) \leq 2$
- $f(x) \leq 2 + 2$
- $f(y) \leq 3 + 1$
- $f(z) = 4$

**Agenda**

```
W
X
Y
Z
```

**Solution**

```
W
X
Y
Z
```
Losing Admissability

If $h$ over-estimates for state $x$, node $X$ dies on the agenda despite being on optimal path.

\[
f(x) = 2 + 4
\]

\[
h(x) = 4
\]

\[
f(x) = 2 + 4
\]

\[
h(v) = 2
\]

\[
f(v) = 3 + 2
\]

\[
f(w) = 5 + 0
\]

\[
W
X
Y
Z
\]

Solution

Found

Agenda

Solution Found

Time
What is $g$?
Find an admissible $h$. 
Missionaries and Cannibals: A Classic AI Puzzle

3 x (M,C) => Boat holds 2
(4 or 5) x (M,C) => Boat holds 3
(6 or more) x (M,C) => Boat holds 4

What is g?
Find an admissible h.
Visit every peg from start to end. Neighbor distance is D, but each turn requires an extra amount, W, for wrapping.

- What is g?
- Find an admissible h.
Example Demos

- Missionaries and Cannibals
- Rush Hour
- Navigation