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
3 Common Incremental Search Algorithms

Depth-First

Breadth-First

Best-First

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: these algorithms: 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.
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 = \(\emptyset\) return FAIL
- \(X \leftarrow \text{pop}(\text{OPEN})\)
- push(\(X\),CLOSED)
- If \(X\) is a solution, return (\(X\), SUCCEED)
- SUCC \(\leftarrow\) generate-all-successors(\(X\))
- For each \(S \in\) SUCC do:
  - If node \(S^*\) has previously been created, and if state(\(S^*\)) = state(\(S\)), then \(S \leftarrow S^*\).
  - push(\(S\),kids(\(X\)))
  - If not(\(S \in\) OPEN) and not(\(S \in\) 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 \(S \in\) CLOSED then propagate-path-improvements(\(S\))
Node Expansion in A*
Updating $g$ values in A*

Before

$P_1$
$g = 4 \quad h = 20$

$C_1$
$g = 5 \quad h = 19$

$C_2$
$g = 6 \quad h = 18$

After

$P_2$
$g = 2 \quad h = 22$

$C_1$
$g = 3 \quad h = 19$

$C_2$
$g = 4 \quad h = 18$
Admissable Heuristics for A*

An admissable heuristic NEVER overestimates distance to the goal.

\[ f(x) \leq 2 + 2 \]
\[ f(y) \leq 3 + 1 \]
\[ f(z) = 4 + 0 \]

\[ f(v) \leq 3 + 2 \]
\[ f(w) = 5 + 0 \]

Color Coding: 
- Blue: Known
- Red: To Be Explored
- Black: Explored

Agenda

Keith L. Downing

Best-First Search: The A* Algorithm
What is $g$?

Find an admissible $h$. 
Missionaries and Cannibals: A Classic AI Puzzle

What is $g$?
Find an admissible $h$. 

Keith L. Downing
Best-First Search: The $A^*$ Algorithm
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