11 Informed Search

Wednesday, March 5, 2025 3:15 PM

- HEURISTICS:

pg. 8, 9, 10 : Uninformed search.
No information about the goal is taken into account.

Sometimes we <u>Do</u> have hints about the goal

IDEA: Use information about the goal to select paths with better "potential"
"heuristic"

DEF heuristic function $h(n)$ nodes $\to$ value

value.- an estimate of the cost of the least-cost-path from n to the goal.

We can extend h to paths:

$$h(\langle n_0, n_1, n_2, n_3, \ldots n_k \rangle) = h(n_k)$$

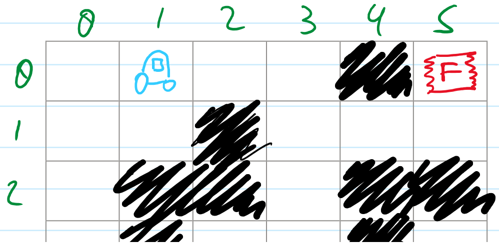
Note: $h(n)$ should be quick to compute.

Admissible Heuristic.- – Non-negative.
– Underestimate of true cost

underestimate .- no path from n to the goal can have cost less than $h(n)$

E.G.

E.G. "Grid World"



states: $\langle row, col \rangle$

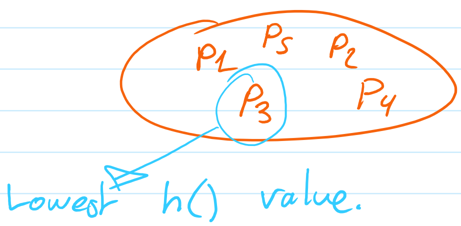
goal : $\langle 4,5 \rangle$

$h(n)$: distance between n and goal

h(n): distance between n and goal (disregarding walls)

- Developing a heuristic is Domain Dependent
    - Try solving a "simpler" problem.

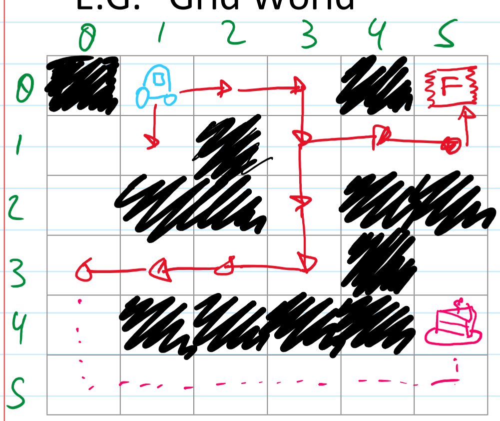
- SEARCH STRATEGY: Greedy Best-First Search
    - Treat Frontier as a priority queue where paths are ordered by their heuristic value.



P_3 is the path with the lowest heuristic value
P_3 is tested
If P_3 is not a solution extensions of P_3 are inserted in the frontier.

E.G.

E.G. "Grid World"



- Action Order. ↑ → ↓ ←
- Tiebreaker: prefer last added.

Frontier — Prio Queue.

~~⟨0,1⟩ 8~~
~~⟨0,1⟩⟨0,2⟩ 7~~
~~⟨0,1⟩⟨1,1⟩ 7~~
...⟨1,0⟩ 8
~~...⟨0,3⟩ 6~~
~~...⟨1,3⟩ 5~~
~~...⟨1,4⟩ 4~~
~~...⟨2,3⟩ 4~~
~~...⟨3,3⟩ 3~~
~~...⟨3,2⟩ 4~~
~~...⟨3,1⟩ 5~~
~~...⟨1,5⟩ 3~~

• Tiebreaker: prefer last added.

~~⟨3,1⟩ 5~~
~~⟨1,5⟩ 3~~
~~⟨0,5⟩ 4~~
~~⟨3,0⟩ 6~~
⟨2,0⟩ 7
⟨4,0⟩ 5

- SEARCH STRATEGY: A-star Search A^*

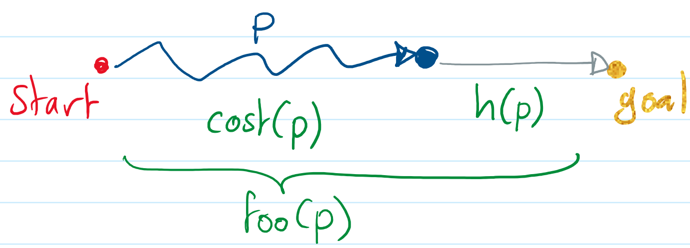
- A mix of Lowest-cost-first-Search & Greedy-best-first Search
  - Lowest-cost-first-Search: cost(p)
  - Greedy-best-first Search: h(p)

In A^*: $cost(p) + h(p)$

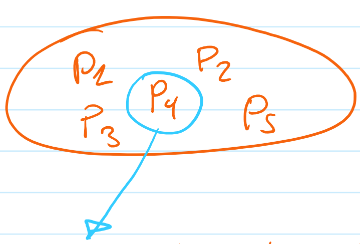
In A^* search strategy:
treat the frontier as a priority queue where the paths are ordered by $foo(p)$ value where

$$foo(p) = cost(p) + h(p)$$

$foo(p)$: an estimate of the lowest cost of a path to the goal through p



frontier



p_4 has the lowest foo() value.
p_4 is tested
Extensions of p_4 are inserted in the frontier.

Note A^* always finds the optimal solution when:

Note A^* always finds the optimal solution when:

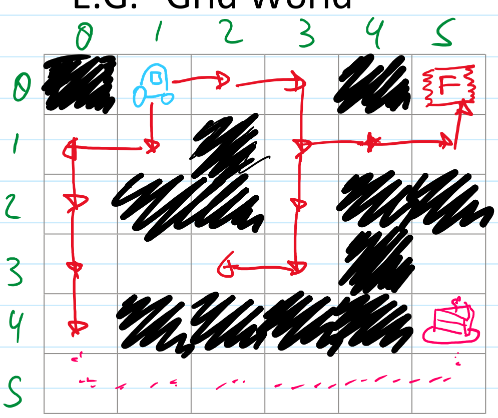
- The branching factor is finite
- The arc costs are "bounded"

  $\exists \alpha \geq 0$ all costs are greater than α

- $h(p)$ is non-negative and an underestimate.

E.G.

E.G. "Grid World"



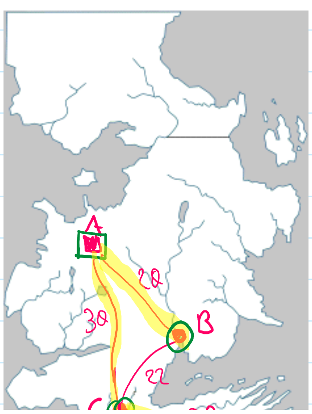
- Action Order. ↑ → ↓ ←
- Tiebreaker: prefer last added.

frontier (Priority Queue)

~~(0,1)~~ 0 + 8 = 8
~~(0,1)(0,2)~~ 1 + 7 = 8
~~(0,1)(1,1) 1 + 7 = 8~~
~~... (1,0) 2 + 8 = 10~~
~~... (0,3) 2 + 6 = 8~~
~~... (1,3) 3 + 5 = 8~~
~~.. (1,4) 4 + 4 = 8~~
~~.. (2,3) 4 + 4 = 8~~
~~.. (3,3) 5 + 3 = 8~~
~~.. (3,2) 6 + 4 = 10~~
~~.. (1,5) 5 + 3 = 8~~
~~.. (0,5) 6 + 4 = 10~~
.. (3,1) 7 + 5 = 12
~~.. (2,0) 3 + 7 = 10~~
~~.. (3,0) 4 + 6 = 10~~
.. (3,1) 5 + 5 = 10
~~.. (4,0) 5 + 5 = 10~~

EG #2

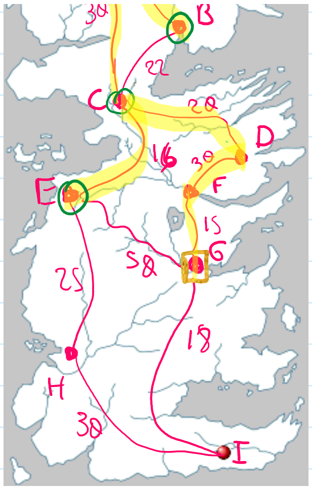
Euclidean Distance is an Acceptable Heuristic.



h(n)	
A	55
B	43
C	31
D	26
E	26
F	12
G	0
H	30

frontier

~~[A]~~ 0 + 55 = 55
~~[A,B]~~ 20 + 43 = 63
~~[A,C]~~ 30 + 31 = 61
[A,C,B] 52 + 43 = 95
~~[A,C,D]~~ 50 + 26 = 76
~~[A,C,E]~~ 46 + 26 = 72
[A,C,E,G] 96 + 0 = 96
[A,C,E,H] 71 + 30 = 101



G 0
H 30
I 15

[A,C,E,G] 96+0 = 96
[A,C,E,H] 71+30 = 101
~~[A,C,D,F]~~ 80+12 = 92
~~[A,C,D,F,G]~~ 95+0 = 95

[A,C,D,F,G]

- VIZUALISATIONS :

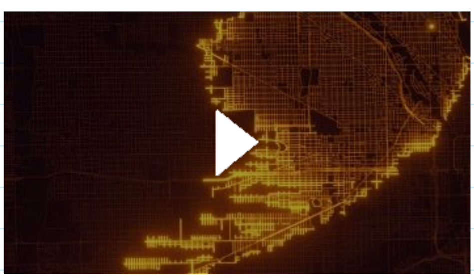
- Dijkstra vs A*
Lowest-cost-first cost + h

https://upload.wikimedia.org/wikipedia/commons/2/23/Dijkstras_progress_animation.gif

https://upload.wikimedia.org/wikipedia/commons/5/5d/Astar_progress_animation.gif

Pathfinding algorithm comparison: Dijkstra's

A* (A-Star) Pathfinding Algorithm Visualization on a Real Map



- VISUALIZING THE DIFFERENT SEARCH ALGORITHMS:

  https://qiao.github.io/PathFinding.js/visual/

EOF