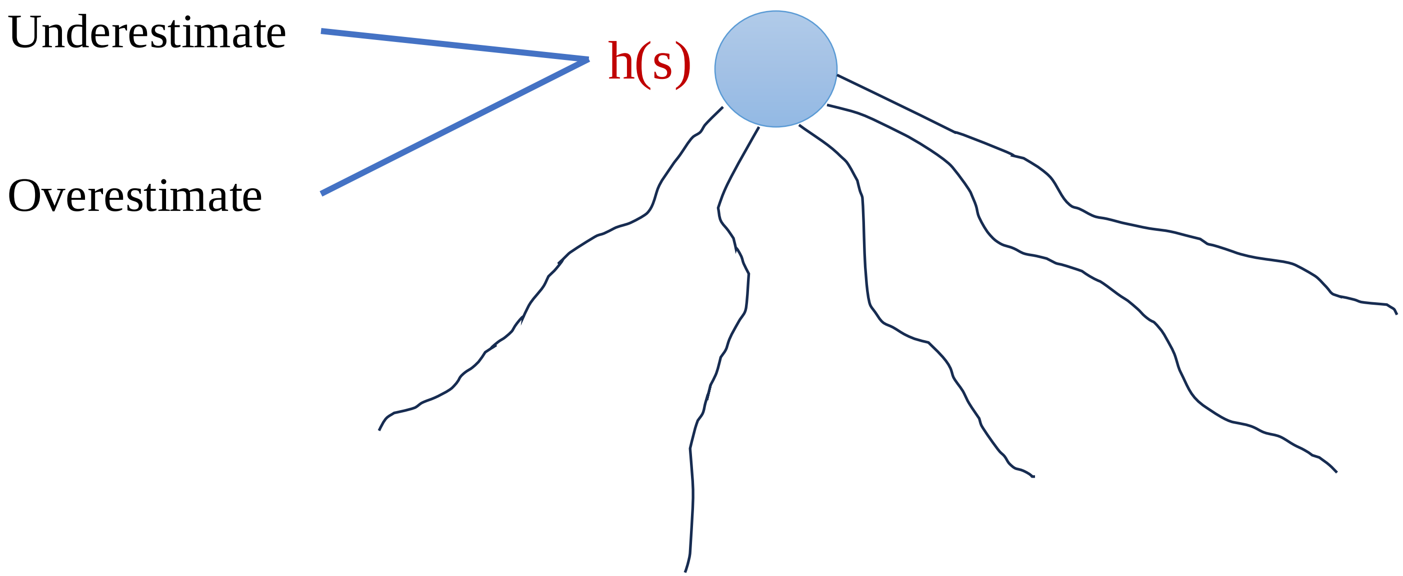
Informed State Space Search

14/01/2025

Koustav Rudra

The notion of heuristics

- Heuristics use domain specific knowledge to estimate the quality or potential of partial solutions



The notion of heuristics

- Examples:
  - Manhattan distance heuristic for 8 puzzle

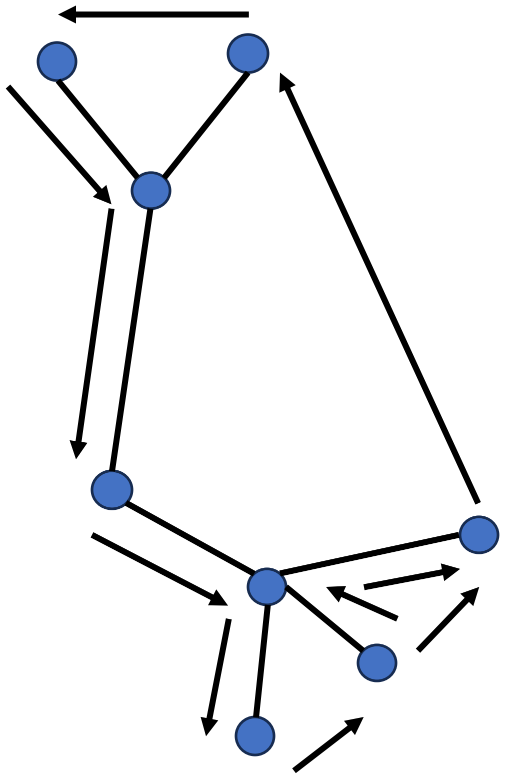
5	6	7
4	1	8
3	9	

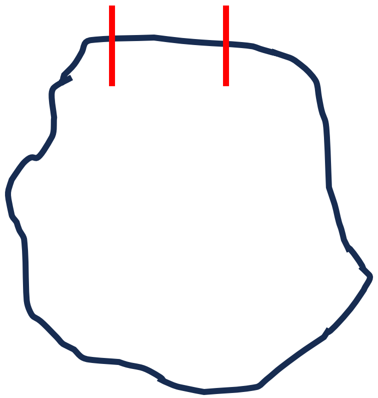
1	2	3
4	5	6
7	8	

2 + 0 + 4

The notion of heuristics

- Examples:
  - Minimum spanning tree heuristic for TSP





$C_S \quad < C^* < 2C_S$

The informed search problem

- Given: [S,s,O,G,h]   where
  - S is the (implicitly specified) set of states
  - s is the start state
  - O is the set of state transition operators each having some cost
  - G is the set of Goal states
  - h() is a heuristic function estimating the distance to a goal

- To find:
  - A minimum cost sequence of transitions to a goal state

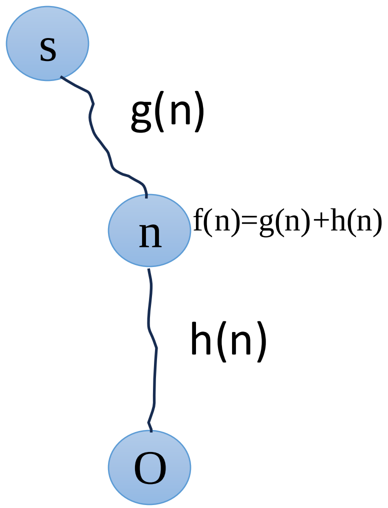
Algorithm A*

- Initialize: Set OPEN={s}, CLOSED = {}, g(s)=0, f(s) = h(s)

- Fail:
  - If OPEN={}, Terminate with failure

- Select: Select the minimum cost state, n, from OPEN and save in CLOSED

- Terminate:
  - If n∈G, terminate with success



Algorithm A*

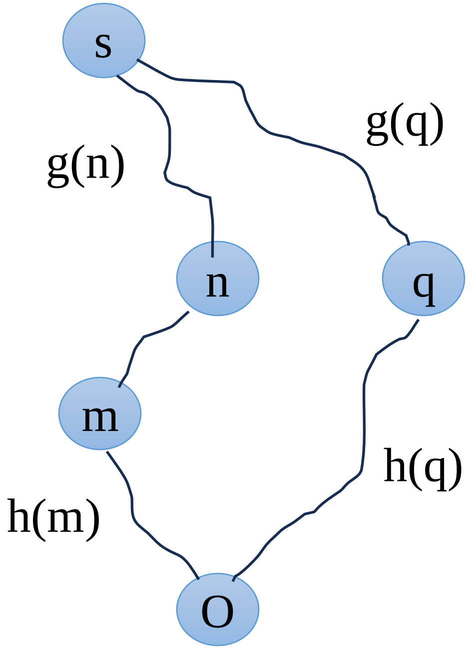
- Expand:
  - For each successor, m, of n:
    - If m∉[OPEN∪CLOSED]
      - Set $g(m) = g(n) + C(n, m)$
      - Set $f(m) = g(m) + h(m)$
      - Insert m in OPEN

    - If m∈[OPEN∪CLOSED]
      - Set g(m) = $min \begin{cases} g(m) \\ g(n) + C(n, m) \end{cases}$
      - Set $f(m) = g(m) + h(m)$
      - If f(m) has decreased and $m \in CLOSED$
        - Move m to OPEN

- Loop:
  - Go to step 2

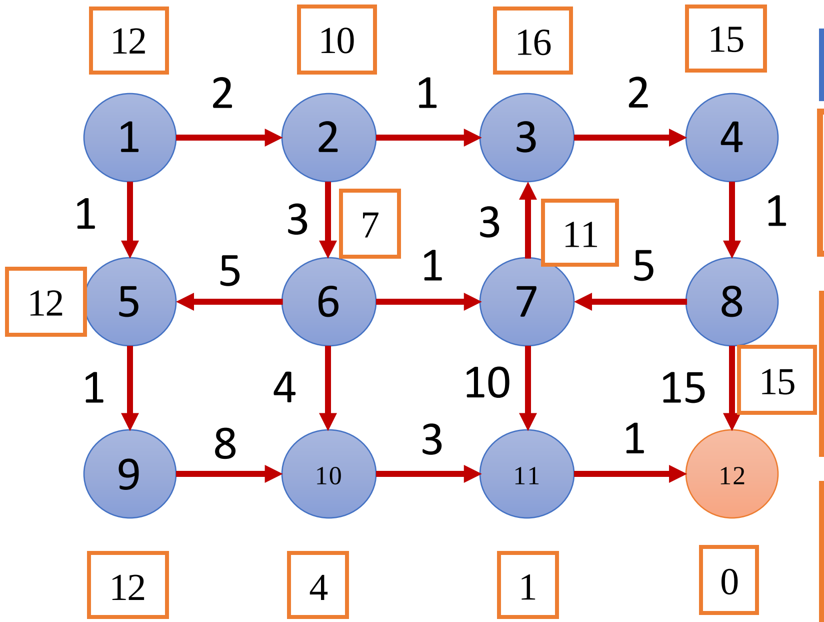
Algorithm A*

- How can a promising path become non-promising?
  - When moving from n to m, we may find a path with less heuristic cost



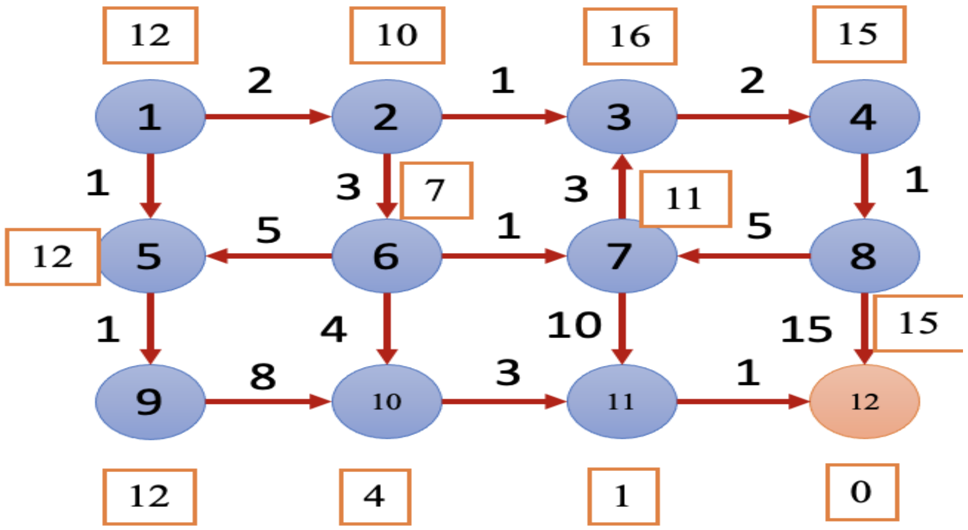
$$h(q) \gg h(m)$$

Algorithm A*



OPEN SET	SELECT	GOAL	EXPANDED	CLOSED
[1(12)]	1(12)	N	[2(12),5(13)]	[1(12)]
[2(12),5(13)]	2(12)	N	[5(13),3(19),6(12)]	[1(12),2(12)]
[5(13),3(19),6(12)]	6(12)	N	[5(13),3(19),7(17),10(13)]	[1(12),2(12),6(12)]
[5(13),3(19),7(17),10(13)]	5(13)	N	[3(19),7(17),10(13),9(14)]	[1(12),2(12),6(12),5(13)]
[3(19),7(17),10(13),9(14)]	10(13)	N	[3(19),7(17),9(14),11(13)]	[1(12),2(12),6(12),5(13),10(13)]

Algorithm A*



OPEN SET	SELECT	GOAL	EXPANDED	CLOSED
[3(19),7(17),10(13),9(14)]	10(13)	N	[3(19),7(17),9(14),11(13)]	[1(12),2(12),6(12),5(13),10(13)]
[3(19),7(17),9(14),11(13)]	11(13)	N	[3(19),7(17),9(14),12(13)]	[1(12),2(12),6(12),5(13),10(13),11(13)]
[3(19),7(17),9(14),12(13)]	12(13)	Y		

Algorithm A*: Benefit

- Reduces number of expanded nodes

- Performs the lookahead and tells us promising paths

- What about optimality?

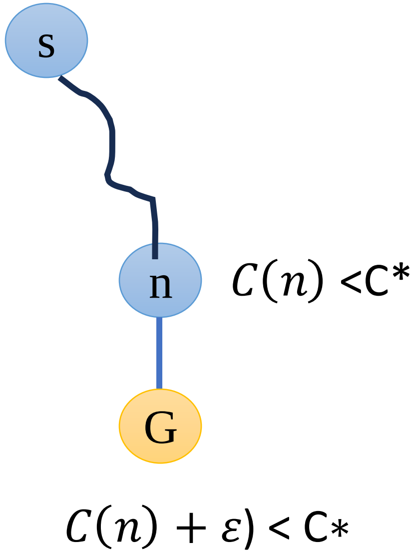
Uniform Cost Search

- Claim: If C(n)<C* (optimal cost) then n must be expanded
- Let algorithm A does not expand n

- For the class of algorithms without any heuristics
  - All states that have cost < $C*$ will have to be expanded

  - Always expands the minimum cost node in your frontier

  - When we find the goal
    - All the states that we have in the frontier have cost higher than the goal state



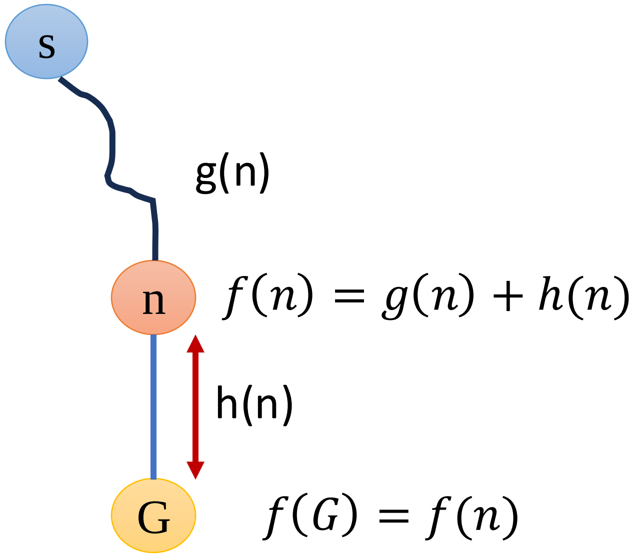
Algorithm A*: Benefit

- Claim: $f(n) < C^*$ then n must be expanded

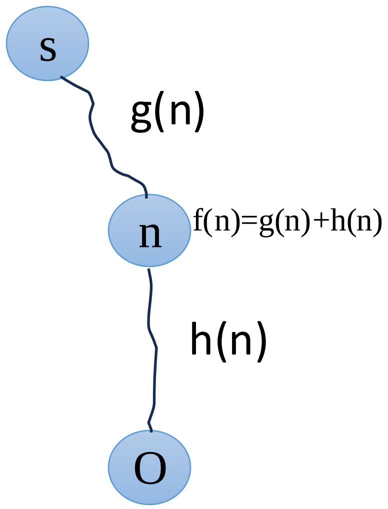
- The heuristic function underestimates
  - $h(n) \leq f^*(n)$
    - Cost of reaching goal from n
  - All costs are +ve

- If we do not expand n, we can't find the goal

- If we have a state whose cost is less than C*
  - Then every algorithm which guarantees finding optimal solution have to expand it



Algorithm A*



- Initialize: Set OPEN={s}, CLOSED = {}, g(s)=0, f(s) = h(s)
- Fail:
  - If OPEN={}, Terminate with failure
- Select: Select the minimum cost state, n, from OPEN and save in CLOSED
- Terminate:
  - If n∈G, terminate with success

How to break the tie?

Thank You