AIFA
Searching With Costs

09/01/2024

Koustav Rudra

Search and Optimization

- Given: [S, s, O, G]

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

Search with Cost

- Initialize: Set OPEN={s}, CLOSED = {}, Set C(s)=0

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

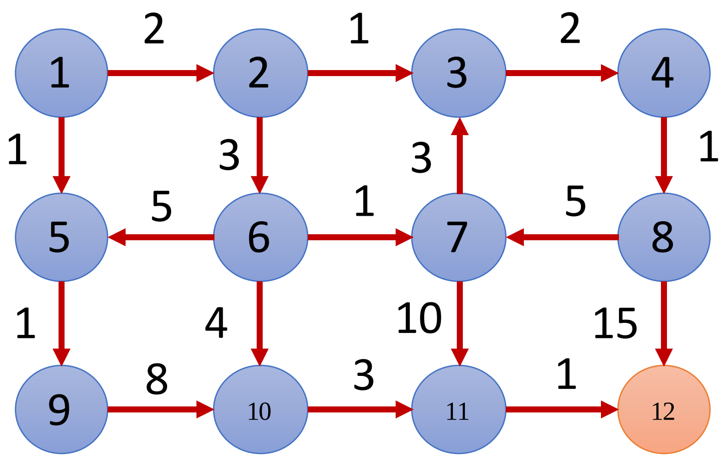
- Select: Select the minimum cost state, n, from OPEN and
  - Save n is CLOSED

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

Search with Cost

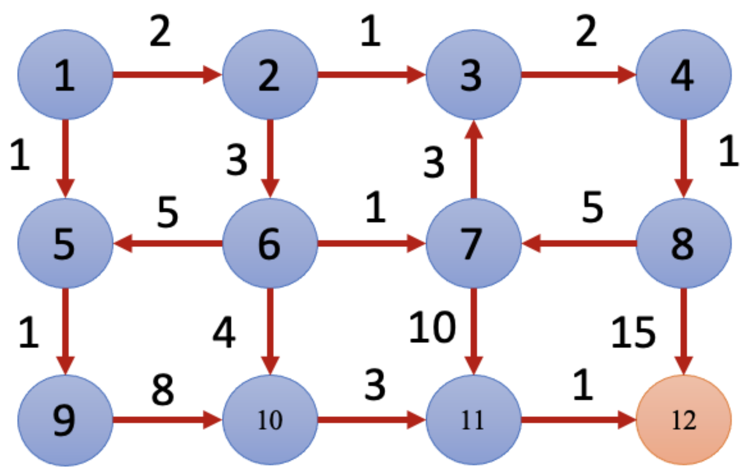
- Expand:
  - Generate successors of n using O
  - For each successor, m:
    - If m∉[OPEN∪CLOSED]
      - Set $C(m) = C(n) + C(n, m)$
      - Insert m in OPEN
    - If m∈[OPEN∪CLOSED]
      - Set C(m) = $min \begin{cases} C(m) \\ C(n) + C(n, m) \end{cases}$
      - If C(m) has decreased and $m \in CLOSED$
        - Move m to OPEN
- Loop:
  - Go to step 2

Search with Cost



OPEN SET	SELECT	GOAL	EXPANDED	CLOSED
[1(0)]	1(0)	N	[2(2),5(1)]	[1(0)]
[2(2),5(1)]	5(1)	N	[2(2),9(2)]	[1(0),5(1)]
[2(2),9(2)]	2(2)	N	[9(2),3(3),6(5)]	[1(0),5(1),2(2)]
[9(2),3(3),6(5)]	9(2)	N	[3(3),6(5),10(10)]	[1(0),5(1),2(2),9(2)]
[3(3),6(5),10(10)]	3(3)	N	[6(5),10(10),4(5)]	[1(0),5(1),2(2),9(2),3(3)]

Search with Cost

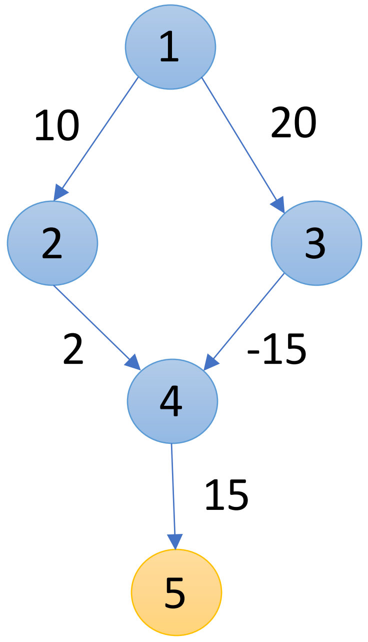


OPEN SET	SELECT	GOAL	EXPANDED	CLOSED
[3(3),6(5),10(10)]	3(3)	N	[6(5),10(10),4(5)]	[1(0),5(1),2(2),9(2),3(3)]
[6(5),10(10),4(5)]	6(5)	N	[10(9),4(5),7(6)]	[1(0),5(1),2(2),9(2),3(3),6(5)]
.				
.				
.				
[12(13)]	12(13)	Y		

Searching with Cost

- What are advantages of having positive cost?
- What will happen if operators have unit cost?
- What will happen if we have negative edge cost?

Searching with Cost



OPEN SET	SELECT	GOAL	EXPANDED	CLOSED
[1(0)]	1(0)	N	[2(10),3(20)]	[1(0)]
[2(10),3(20)]	2(10)	N	[3(20),4(12)]	[1(0),2(10)]
[3(20),4(12)]	4(12)	N	[3(20),5(27)]	[1(0),2(10),4(12)]
[3(20),5(27)]	3(20)	N	[4(5),5(27)]	[1(0),2(10),~~4(12)~~,3(20)]
[4(5),5(27)]	4(5)	N	[5(20)]	[1(0),2(10),~~4(12)~~,3(20),4(5)]
[5(20)]	5(20)	Y		

AIFA
Branch and Bound

09/01/2024

Koustav Rudra

Branch and Bound

- We know an upper bound on solution cost

- We can do BFS/DFS to find out a goal and its associated cost
  - But it is not guaranteed that it is least cost goal
  - However, it works as an upper bound

Branch and Bound

- Initialize: Set OPEN={s}, CLOSED = {}, Set C(s)=0, $C^* = \infty$
- Fail:
  - If OPEN={}, then return C^*
- Select: Select a state, n, from OPEN and save in CLOSED
- Terminate:
  - If n∈G, and C(n) < C^*, then
    - C^*=C(n) and Go To Step 2

Branch and Bound

- Expand:
  - If C(n) < C^* , Generate successors of n using O
  - For each successor, m:
    - If m∉[OPEN∪CLOSED]
      - Set $C(m) = C(n) + C(n,m)$
      - Insert m in OPEN
    - If m∈[OPEN∪CLOSED]
      - Set C(m) = $min \begin{cases} C(m) \\ C(n) + C(n,m) \end{cases}$
      - If C(m) has decreased and $m \in CLOSED$
        - Move m to OPEN

- Loop:
  - Go to step 2

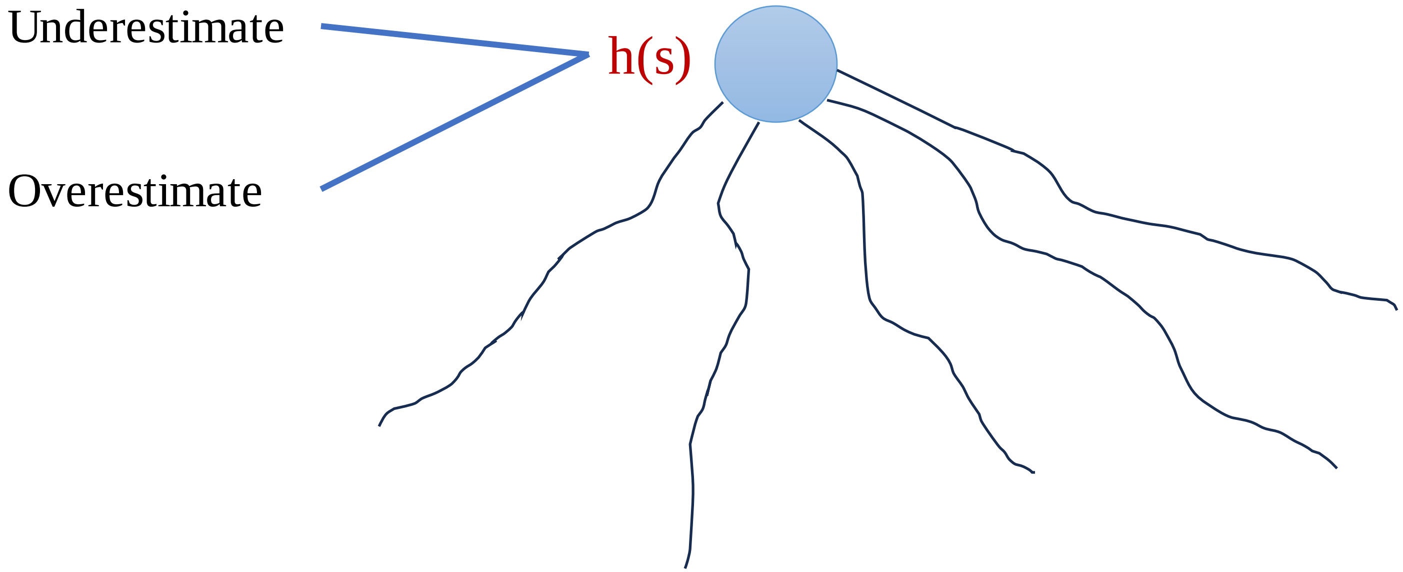
AIFA
Informed State Space Search

09/01/2024

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 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

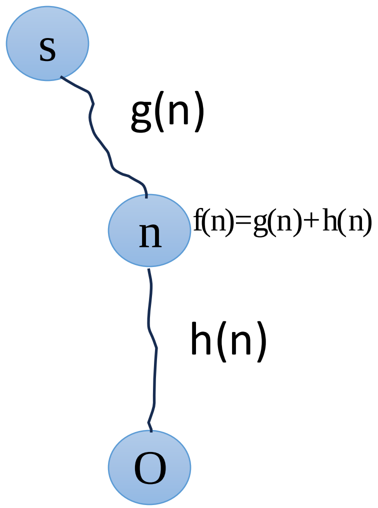
Algorithm A*

- 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

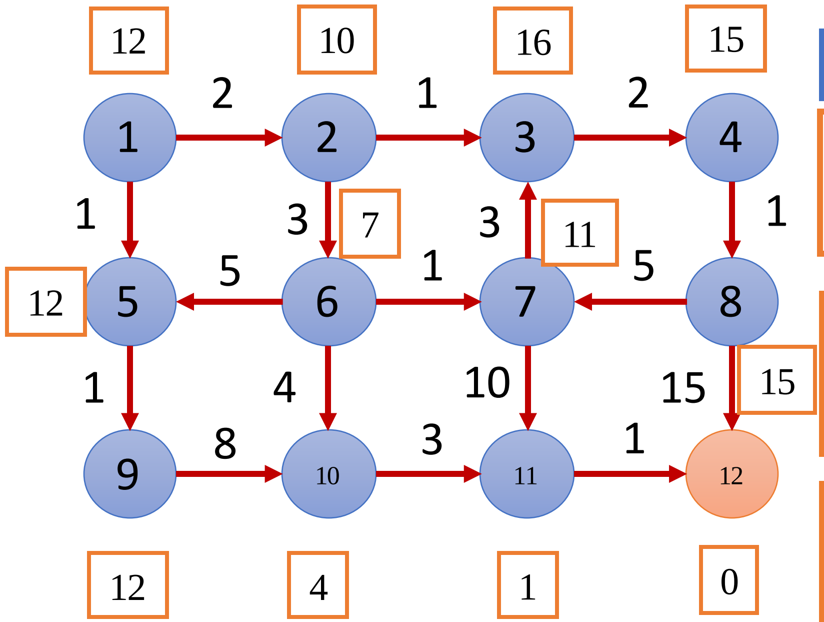
- Terminate:
  - If n∈G, terminate with success



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 $g(\mathrm{m}) = min \begin{cases} g(m) \\ g(n) + C(n,m) \end{cases}$
      - 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*



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)]

.

.

.

Thank You