AI first class

September 10, 2019

To formalize or operationalize (informal problem description → formal problem description) Initial state to goal state.

8		6
5	4	7
2	3	1

Initial State

	1	2
3	4	5
6	7	8

Goal State

Figure 1: Problem solving state

For Above 8 puzzle problem,

- States:

  * A state descibes loacation of each 8 tiles.

- Operators:

  *Blank moves left, right up or down.

- Goal test:

  *State matches goal configuration

- Path cost:

  *Each step costs 1.

As one solve the above problem, it is equivlanent to view as going down the tree to reach goal state. We first construct a search tree, the branches are actions.

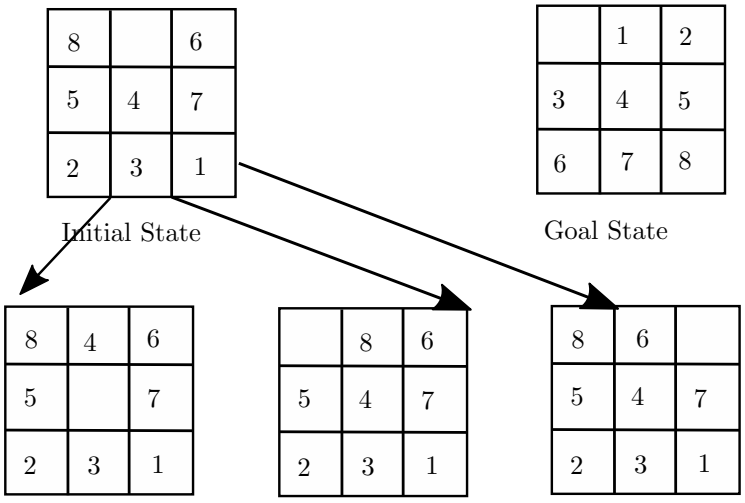

Figure 2: goaltree

In this manner, Traveling Salesman problem can be described formally with above notations. We begin with starting country, and every node is other country visted. However, in Teaveling Salesman problem there may be node with same values. In **Graph Search**, we solve this problem by making another data structure that checks visited states. In general, we have Infrastructire for search algorithm

- Child Node: takes parent node and action -¿ child node
- solution function returns the sequence of actions obtained by following parent poitners back to the
- the frontier is shtored in **queue**.
- explored nodes are stored in hash table

For search alogirithms, we have : Evalutaion criteria

1. Completeness: is it guaranteed to find a solution?
2. Optimality: does it find the optimal solution?
3. Time Complexity: ?
4. Space Complexity: ?

Some expressions to describle graph is:

- b: the branching factor or maximum number of successor of any node
- d: depth (smallest goal node)
- m: maximum length (of search tree)

1 Search

There are about two kind of Search:

- Uninformed search
- Informed search (this is more likely to be seen as "AI")

1.1 Uninformed Search

1.1.1 BFS

Expand the shallowest unexpandede nde in frontier.

Time Complexity: $1+b+b^2+b^3+...+b^d$

BFS has exponential time complexity!! very bad.

1.1.2 Uniform cost search

IT expand the least - cost unexpandede node on the frontier rather than shallowest node. Make a priority queue, ordered by path cost (allows redundant path if better than old) It finds the least-cost -optimal path overall. Time complexity is reduced just by little.

1.1.3 DFS

Expands the deepest unexpanded node int he frontier.

1.1.4 Depth limited search

Equal to the DFS with depth limit l Nodes at depth l has no successors. This does not have the completness guaranteed, yet, Once one have a idea of diameter, we

1.1.5 Iterative deepeinig Search

Evalutaion: Optimal and complete like breadth-first, and requires modest memory like depth first. however we see that