AI Theory, W4/L1

Review: State definitions

State:

"Summary of actions until a given time t".

Algorithms of Trees

Name	Action costs	Space	Time
BackTracking	any	$O(D)$	$O(b^D)$
DFS	Zero	$O(D)$	$O(b^D)$
BFS	≥ 0	$O(b^d)$	$O(b^d)$
DFS-ID	≥ 0	$O(d)$	$O(b^d)$

Review of Tree Algorithms.

Back to our problem:

Transportation Problem

dist: 1 . . . . . n

walk: cost 1 , St 1

Tram: cost 2 , s^*2

Let's add a constraint:

"Cant visit three odd cities in a row".

For $N=10$, how many states (represented by integers) will be

there with the actions?

Discrete Mathematics 101

keep track of all odd states
(again, actions don't matter)

<u>10</u> prev city

<u>10</u> next city

1 - 1
1 - 2
1 - 3
9 - 1

is also a state

Total states : n^2

$\Rightarrow$ we can try to keep a counter of odd states.
future is decided on basis of

counter.

0	1	2
ok	ok	N.ok
prev. city	curr. city	raise Exception

$\Rightarrow$ prev. city (odd/even), curr. city (even/odd)

will further filter out the states from n^2 to $2n$.

1 ..... n 1 ..... n

odd cities even cities

New constraint

visit atleast 3 odds before goal.

What is the minimal state?

=> We can use the counter approach here, will work.

State =

min((no. of odd cities visited, 3), curr. city)

<u>Constraint:</u>

Travel from 1 to n, but odd cities should be more than even.

Odd = even

Odd should be multiple of even

traverse multiples of three

exam pov important

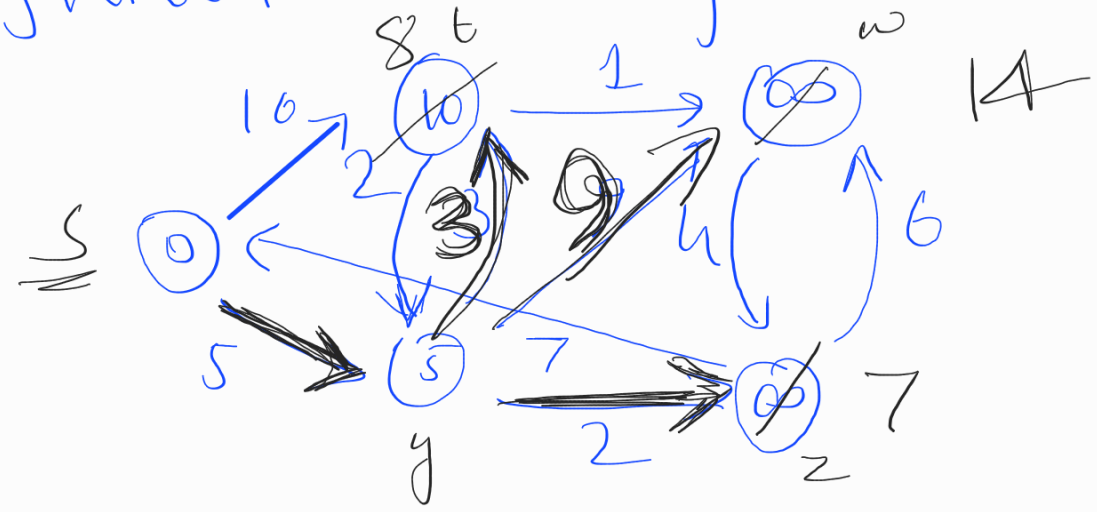
above questions inspired from Theory of Automata and have a high chance of papers.

Dynamic Programming has an issue:

it works for acyclic graphs, what if there are cycles?

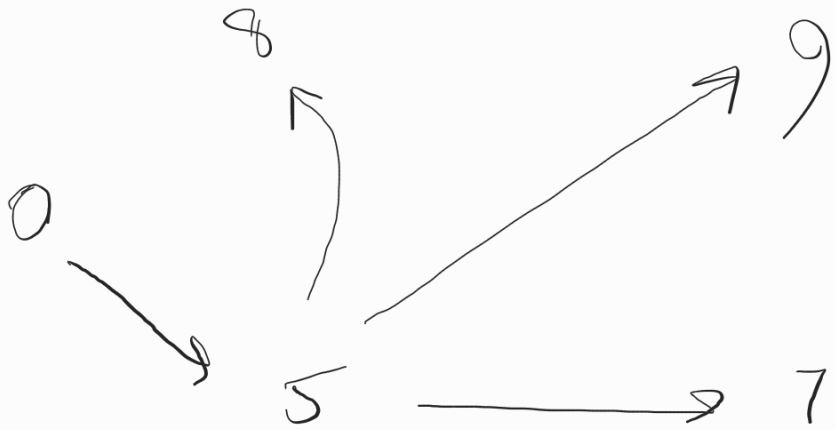
Uniform Cost Search

Intuition: Dijkstra's Algorithm



if node.cost + edge.cost < destination.cost

destination.cost = node.cost + edge.cost



Uniform Cost Search (UCS):

Dijkstra pseudocode

Add start-st to priorityQueue

Repeat until frontier is empty.

on which you are processing

Remove s with smallest priority p from frontier

UCS change

If isGoal(s) ← return solution

Add s to explored

Foreach action $\in$ Actions(s)

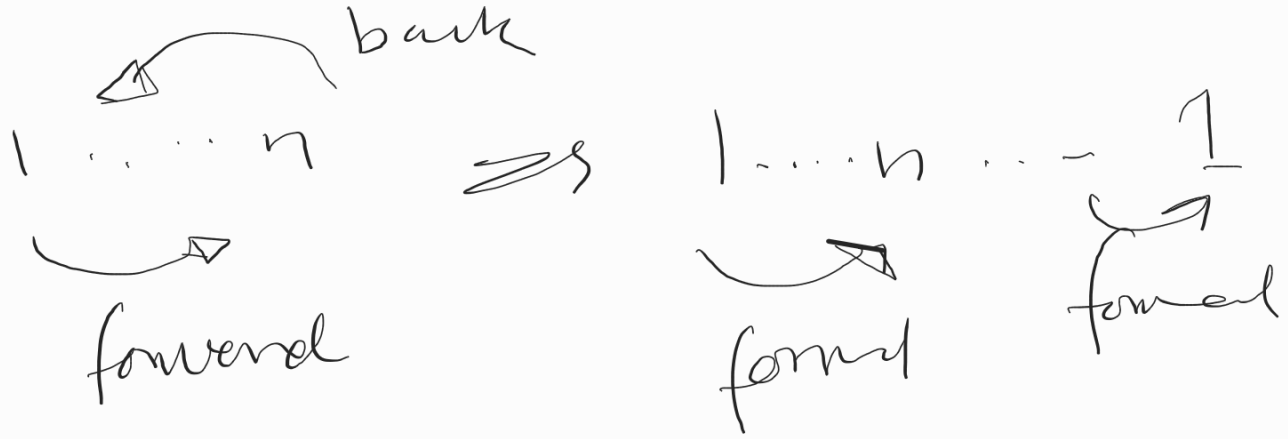
Get NextState s' from Transition(s, a)

If s' in explored ← Continue

Add s' to frontier with priority s' + cost(s', a)

More towards UCS

Dijkstra + Stop on Goal → Uniform Cost Search



is how we model this problem.

Learning as an Inverse Problem:

prev. data $\longrightarrow$ learner $\longrightarrow$ learns new data

(it doesn't know cost of actions, only end answer).

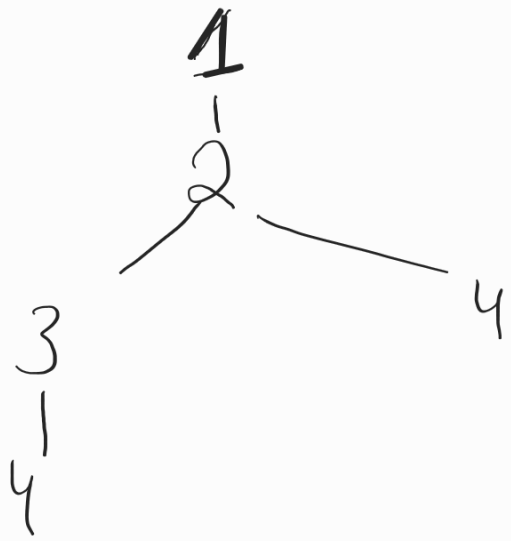
We try to figure out action cost in learner problem, given action.

↳ Back problem

But given action cost, when we try to figure action.

↳ Forward problem

Input: Search without costs



Assume Cost

walk = 3

tram = 2

Solution: Optimal solution:

expert told us → walk x 3

According to our solution, our assumption is wrong.

walk walk walk

So we adjust our assumptions..

y = Actual = walk walk walk

y' = Predicted = walk tram

Update costs:

$$\text{Cost} = \underbrace{\text{prev walk}}_{3} - \underbrace{1 - 1 - 1 + 1}_{} = 1$$

when we see it in predicted path, we +1
when we see it in actual path, −1

$$\text{Tram cost} = \underbrace{\text{prev tram}}_{2} + 1 = 3$$

(walk, 1
tram, 3) are the correct learning rates.

Algorithm: Structured Perceptron

for each action $w[a] \leftarrow 0$;

foreach iteration $t = 1 \ldots T$

for each training example $(x, y) \in D_{train}$

→ Compute minimum cost-w

→ for correct pred. $w+1$

→ for wrong pred $w-1$

This is kind of the crux.

Learning Problem requires voluminous amounts of data.

Say that we have a data generator that gives us data in the form of:

(totalcost, [actions ...])

What is the cost of $\boxed{\text{individual actions?}}$

ON SLIDES is the working of

StructuredPerceptron

Types of Data

many types

Text

Image/video

Speech

Sensors

$\rightarrow$ features

AI Term Project Blueprint

→ Real world problem

→ Type of data
↳ more complexity indicates more challenge !!!

→ Data
↳ Authentic
↳ Should be close to Pakistan

<u>Proposal</u>: How are you going to solve the problem? How

are you going to effect it in works?

1 person

2 person

3 person

more person == more complexity

2 slides + documents for proposals.

Final proposals for exhibition + AI Theory