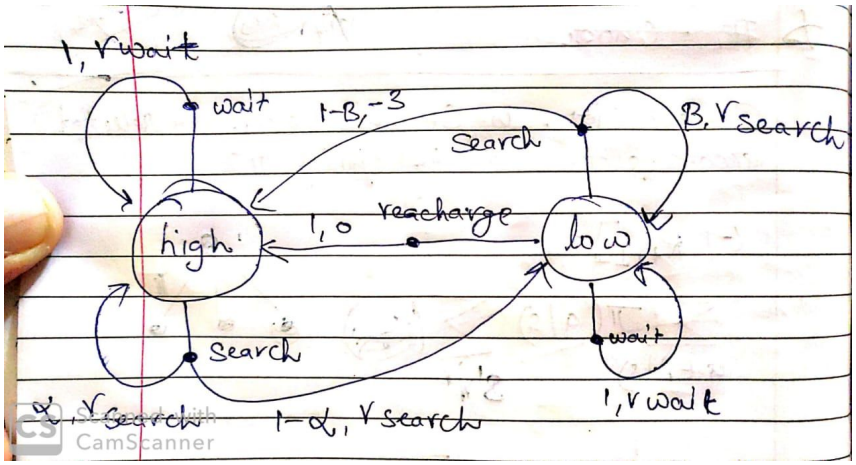
Reinforcement Learning

Assignment 2 (MDP and Dynamic Programming)

Prashant Pathak (MT19051)

Question 1

Below we have the MDP of the problem.



For making the table we have to consider all the possibility of the current states (s), actions that can be taken(a), next sates (s') and rewards(r).

As explained in the book, r_{search} and r_{wait} are the expected reward that an agent receives by taking a particular action. So to actually make the table off p(s',r | s, a) we have to consider the distribution (i.e. PMF) of the reward (i.e. r_{search} and r_{wait}) and include all the individual values for making the table. For instance, if r_search can take values r_search 1,r_search 2 and r_search 3 with probability p1,p2 and p3 then all these values will come in the table.

For making the below table I have considered r_{search} and r_{wait} as a constant reward.

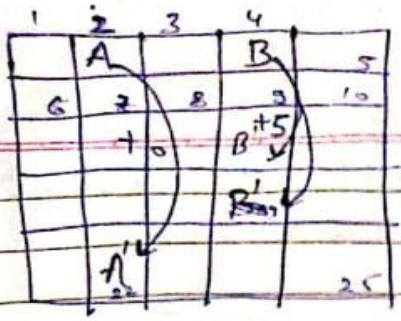
Table for p(s',r | s,a)

s	a	s'	r	p(s',r | s,a)
high	search	high	r_{search}	α
high	search	low	r_{search}	(1-α)
high	wait	high	r_{wait}	1
high	wait	low	---	0
low	search	high	-3	β
low	search	low	r_{search}	(1-β)
low	wait	high	---	0

low	wait	low	r_{wait}	1
low	recharge	high	0	1
low	recharge	low	---	0

Question 2

We have to solve this RL problem by solvinf the linear equations. So First we have found the linear equation and solveed it using matrix opeation.



→ States all grid cell are states we can denote them by
$S_1, S_2 \ldots S_{25}$

→ Action → Four action at each cell. N, S, E, W i.e. North, South, East, West

→ Reward → ~~from~~ ~~S_2~~
~~$r(s, a, s_2)$~~ ~~S_2, any action~~) -
→ Action that take off grid reward is -1
→ Action from A to A' = 10 (despite any action taken
→ Action from B to B' · 5 (despite any actio

We have to find the value function for each state given the policy that all action are equiprobable

$$V_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)\left[r + \gamma V_\pi(s')\right]$$

for all $s \in S$

Here from a state and action pair we can go to only one other state & get definite reward.

Scanned with CamScanner

Equation for each state:

$$V(S_1) = \frac{1}{1}\left[-1 + .9 * V(S_1)\right]$$

$$+ 0.25[-1 + .9 * V(S_1)] + 0.25[0 + .9 * V(S_2)$$
$$+ 0.25[0 + .9 * V(S_6)]$$

$$= -0.5 + 0.45\,V(S_1) + .225\,V(S_2) + .225\,V(S_6)$$

$$0.55\,V(S_1) - .225\,V(S_2) - .225\,V(S_6) + .5 = 0$$

↳ Eq (1)

$$V(S_2) = 10 + 0.9\,V(S_{22})$$

$$V(S_2) - 0.9\,V(S_{22}) - 10 = 0 \quad \text{– Eq (2)}$$

$$V(S_3) = 0.25[0 + 0.9\,V(S_2)] + 0.25[-1 + 0.9\,V(S_3)] + 0.25[0 + 0.9\,V(S_4)] + 0.25[0 + 0.9\,V(S_8)]$$

$$\Rightarrow .225\,V(S_2) + .225\,V(S_3) + .225\,V(S_4) + 0.225\,V(S_8) - 0.25$$

$$0.775\,V(S_3) - 0.225\,V(S_2) - .225\,V(S_4) - 0.225\,V(S_8) + 0.25 \quad \text{– Eq (3)}$$

$$V(S_4) = 5 + 0.9\,V(S_{19}) \quad \text{– Eq (4)}$$

$$0.55\,V(S_5) - .225\,V(S_4) - 0.225\,V(S_{10}) + 0.5 = 0 \quad \text{– Eq (5)}$$

$$V(S_6) = 0.25[-1 + 0.9\,V(S_6)] + 0.25[0 + 0.9\,V(S_1)] + 0.25[0 + 0.9\,V(S_{11})] + 0.25[0 + 0.9\,V(S_7)]$$

$$V(S_6) = -0.25 + .225\,V(S_6) + 0.225\,V(S_1) + 0.225\,V(S_{11}) + 0.225\,V(S_7)$$

$$.775\,V(S_6) - 0.225\,V(S_1) - .225\,V(S_{11}) - 0.225\,V(S_7) + 0.25 = 0 \quad \text{– Eq (6)}$$

Scanned with CamScanner

$V(S_7) = 0.225\,V(S_2) + 0.225\,V(S_8) + 0.225\,V(S_{12}) + 0.225\,V(S_6) - \varepsilon_V(7)$

$V(S_8) = 0.225\,V(S_3) + 0.225\,V(9) + 0.225\,V(S_{13}) + 0.225\,V(S_7) - \varepsilon_0(8)$

$V(S_9) = 0.225\,V(S_4) + 0.225\,V(10) + 0.225\,V(S_{14}) + 0.225\,V(S_8) - \varepsilon_0(9)$

$V(S_{10}) = 0.225\,V(S_5) + 0.225\,V(10) + 0.25 + 0.225\,V(S_{15}) + 0.225\,V(S_9) - \varepsilon_0(10)$

$V(S_{11}) = 0.225\,V(S_6) + 0.225\,V(S_{12}) + 0.225\,V(S_{16}) + 0.225\,V(S_{11}) - 0.25 - \varepsilon_0(11)$

$V(S_{12}) = 0.225\,V(S_7) + 0.225\,V(S_{13}) + 0.225\,V(S_{17}) + 0.225\,V(S_{11}) - \varepsilon_0(12)$

$V(S_{13}) = 0.225\,V(S_8) + 0.225\,V(S_{14}) + 0.225\,V(S_{18}) + 0.225\,V(S_{12}) - \varepsilon_0(13)$

$V(S_{14}) = 0.225\,V(S_9) + 0.225\,V(15) + 0.225\,V(S_{19}) + 0.225\,V(S_{13}) - \varepsilon_0(14)$

$V(S_{15}) = 0.225\,V(S_{10}) + 0.225\,V(S_{15}) - 0.25 + 0.225\,V(S_{20}) + 0.225\,V(S_{14}) - \varepsilon_0(15)$

$V(S_{16}) = 0.225\,V(S_{11}) + 0.225\,V(S_{17}) + 0.225\,V(S_{21}) + 0.225\,V(S_{16}) - 0.25$

$V(S_{17}) = 0.225\,V(S_{12}) + 0.225\,V(S_{18}) + 0.225\,V(S_{22}) + 0.225\,V(S_{16})$

Scanned with CamScanner

$V(S_{18}) = 0.225V(S_{13}) + 0.225V(S_{19}) + 0.225V(S_{23}) + 0.225V(S_{17})$ — Eq (18)

$V(S_{19}) = 0.225V(S_{14}) + 0.225V(S_{20}) + 0.225V(S_{24}) + 0.225V(S_{18})$ — Eq (19)

$V(S_{20}) = 0.225V(S_{15}) + 0.225V(S_{20}) - 0.25 + 0.225V(S_{25}) + 0.225V(S_{19})$ — Eq (20)

$V(S_{21}) = 0.225V(S_{16}) + 0.225V(S_{22}) + .45V(S_{21}) - 0.5$ — Eq (21)

$V(S_{22}) = 0.225V(S_{17}) + 0.225V(S_{23}) + 0.225V(S_{22}) - 0.25 + 0.225V(S_{21})$ — Eq (22)

$V(S_{23}) = 0.225V(S_{18}) + 0.225V(S_{24}) + 0.225V(S_{23}) - 0.25 + 0.225V(S_{22})$ — Eq (23)

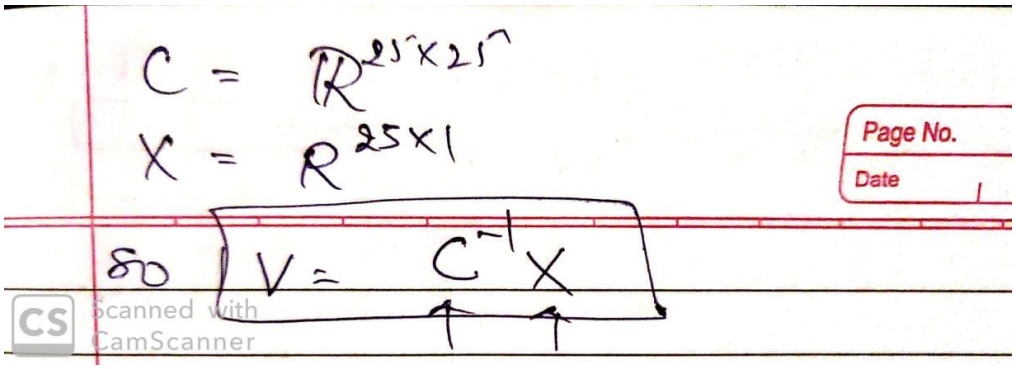
$V(S_{24}) = 0.225V(S_{19}) + 0.225V(S_{25}) + 0.225V(S_{24}) - 0.25 + 0.225V(S_{23})$ — Eq (24)

$V(S_{25}) = 0.225V(S_{20}) + 0.225V(S_{24}) + .45V(S_{25}) - 0.5$ — Eq (25)

Now we have 25 variable & 25 equation we can solve it to find value of value function.

$V * C = X$

V = row vector of state $R^{25\times 1}$



We use the equation above and coded it. Can be viewed in Question2.py

Values of the state after solving the equation:

```
array([[ 3.3,  8.8,  4.4,  5.3,  1.5],
       [ 1.5,  3. ,  2.3,  1.9,  0.5],
       [ 0.1,  0.7,  0.7,  0.4, -0.4],
       [-1. , -0.4, -0.4, -0.6, -1.2],
       [-1.9, -1.3, -1.2, -1.4, -2. ]])
```

Question 3

3.15

3.15 From Eq 3.8 of book we have,

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots$$

$$= \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

Now suppose we add a constant reward C to all the reward. So now

~~$G_t' = R_{t+1} + \gamma C$~~

$$G_t' = (R_{t+1} + C) + \gamma (R_{t+2} + C) + \gamma^2 (R_{t+3} + C) \cdots$$

$$= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} \cdots + C + \gamma C + \gamma^2 C$$

$$= \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} + \sum_{k=0}^{\infty} \gamma^k C$$

$$= G_t + \frac{C}{1-\gamma}$$

↓ This is from 3.8

↳ (Sum of G.P (infinte when $\gamma < 1$))

Now we have to calculate the value of each state.

Scanned with CamScanner

$$V_\pi(s) = E[G_t / S_t = s]$$

Page No.
Date

$$= E\left[\left(G_t + \frac{V_c}{1-\gamma}\right) \middle| S_t = s\right]$$

Using linearity of Expectation.

$$= E[G_t / S_t = s] + E\left[\frac{V_c}{1-\gamma} \middle| S_t = s\right]$$

As $\frac{V_c}{1-\gamma}$ is a constant

$$= E[G_t / S_t = s] + \frac{C}{1-\gamma}$$

So we can say that adding a constant change the final state values by a constant $V_c = \frac{C}{1-\gamma}$

So relative values are not changes between state.

Scanned with CamScanner

3.16 – In case of Episodic task

$G_t = R_{t+1} + R_{t+2} + R_{t+3} - - - - R_T$

where T is terminating time.

Now if we here add a constant C to all the reward

$G_t' = R_{t+1} + C + R_{t+2} + C + - - - R_T + C$

$= R_{t+1} + R_{t+2} + \ldots R_T + C + \ldots C$

Scanned with CamScanner

$G_t' = G_t + \text{Length of Episode} \times C$

Page No.
Date

→ Here length Episode is Not fixed.

$V_\pi(s) = E[G_t' | S_t = s]$

$= E[G_t + \text{length of Episode} \times C | S_t = s]$

$= E[G_t | S_t = s] + \text{length of Ep} \times C$

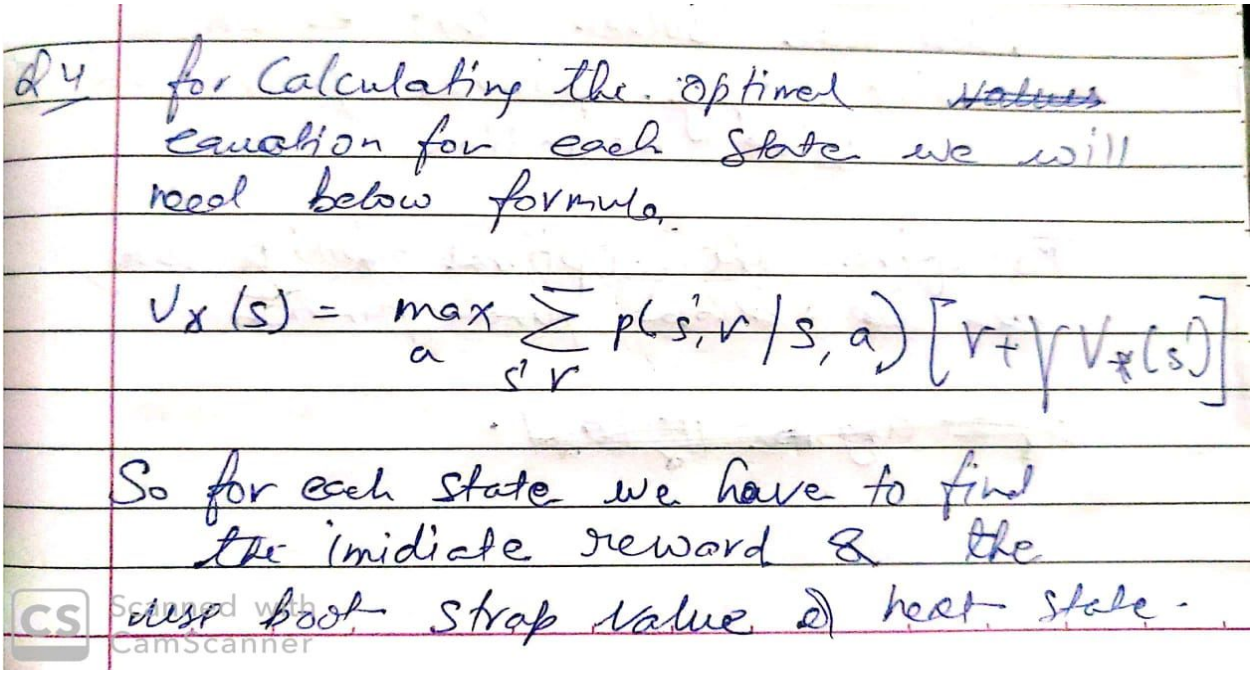
→ Here the value that is added to each episode is not constant so it can affect the relative ordering of the task.

→ Suppose by adding a Constant C to -ve reward it is ~~made~~ positive the our makes it positive the it would affect how agent moves So by both sigh & relative values are important in Episodic task.

Scanned with CamScanner

Question 4

In this, we have to find optimal values function and optimal policy for Previous grid word in question 2. But we have to find it by solving non-linear equations.



Ex for State 1

$$V_*(s_1) = \max_a \Big[\underbrace{(-1 + 0.9\, V_*(s_1))}_{a}, \underbrace{(-1 + 0.9 * V_*(s_1))}_{b}, \underbrace{(0.9 * V_*(s_2))}_{c}, \underbrace{0.9 * V_*(6)}_{d} \Big]$$

~~Similarly for~~

a → It is the case when I move west. receive reward of -1 & reach ~~$V_*(s_1)$~~ s_1

b → It is when we go north, receive reward of -1 & reach state s_1

c → It is when we move east & receive 0 & reach s_2

d → It is when Agent move South and receive 0 & reach s_6

Likewise we have equation for all the state . . .

Equation can be found in code Notebook then we solved this equation using scipy.optimize.fsolve to solve equation.

Scanned with CamScanner

To find the optimal policy we can use below function

$$a_s = \text{argmax}_a \sum_{s'} p(s', r | s, a) \big[r + V_*(s') \big]$$

Scanned with CamScanner

See Question4.py code for detail.

Optimal Values of the state after solvong the non linear equation (It matches that of fig 3.5 of book):

```
[[22.  24.4 22.  19.4 17.5]
 [19.8 22.  19.8 17.8 16. ]
 [17.8 19.8 17.8 16.  14.4]
 [16.  17.8 16.  14.4 13. ]
 [14.4 16.  14.4 13.  11.7]]
```

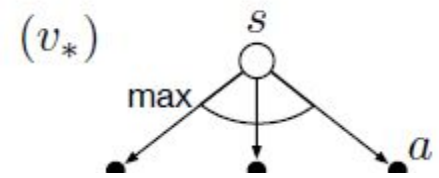
Optimal Policy after solving the equation:

Probability of each action in Cell(i, j) are: [N E S W]
Probability of each action in Cell(0 , 0) are: [0. 1. 0. 0.]
Probability of each action in Cell(0 , 1) are: [0.25 0.25 0.25 0.25]
Probability of each action in Cell(0 , 2) are: [0. 0. 0. 1.]
Probability of each action in Cell(0 , 3) are: [0.25 0.25 0.25 0.25]
Probability of each action in Cell(0 , 4) are: [0. 0. 0. 1.]
Probability of each action in Cell(1 , 0) are: [0.5 0.5 0. 0.]
Probability of each action in Cell(1 , 1) are: [1. 0. 0. 0.]
Probability of each action in Cell(1 , 2) are: [0.5 0. 0. 0.5]
Probability of each action in Cell(1 , 3) are: [0. 0. 0. 1.]
Probability of each action in Cell(1 , 4) are: [0. 0. 0. 1.]
Probability of each action in Cell(2 , 0) are: [0.5 0.5 0. 0.]
Probability of each action in Cell(2 , 1) are: [1. 0. 0. 0.]
Probability of each action in Cell(2 , 2) are: [0.5 0. 0. 0.5]
Probability of each action in Cell(2 , 3) are: [0.5 0. 0. 0.5]
Probability of each action in Cell(2 , 4) are: [0.5 0. 0. 0.5]
Probability of each action in Cell(3 , 0) are: [0.5 0.5 0. 0.]
Probability of each action in Cell(3 , 1) are: [1. 0. 0. 0.]
Probability of each action in Cell(3 , 2) are: [0.5 0. 0. 0.5]
Probability of each action in Cell(3 , 3) are: [0.5 0. 0. 0.5]
Probability of each action in Cell(3 , 4) are: [0.5 0. 0. 0.5]
Probability of each action in Cell(4 , 0) are: [0.5 0.5 0. 0.]
Probability of each action in Cell(4 , 1) are: [1. 0. 0. 0.]
Probability of each action in Cell(4 , 2) are: [0.5 0. 0. 0.5]
Probability of each action in Cell(4 , 3) are: [0.5 0. 0. 0.5]
Probability of each action in Cell(4 , 4) are: [0.5 0. 0. 0.5]

Question 5

Given an equation for v∗ in terms of q∗.

v∗(s) = max_a{q∗(s,a)}



From the backup diagram, we can see that to find the optimal state values function we have to take a single action and then find the maximum from all the q* values.

Question 6

Implementation of the Policy Iteration and Value Iteration can be found in Question6_PolicyIteration.ipynb and Question6_ValuesIteration.ipynb respectively. Sample values and policy are printed there.

State Value of Each iteration

Policy Evaluation : Iteration 1	
1) [[0. -1. -1. -1.] [-1. -2. -2. -2.] [-1. -2. -2. -2.] [-1. -2. -2. 0.]]	2) [[0. -7. -10. -11.] [-7. -10. -11. -11.] [-10. -11. -10. -8.] [-11. -11. -8. 0.]]
3) [[0. -13. -19. -21.] [-13. -17. -19. -19.] [-19. -19. -17. -13.] [-21. -19. -13. 0.]]	4) [[0. -14. -20. -22.] [-14. -18. -20. -20.] [-20. -20. -18. -14.] [-22. -20. -14. 0.]]
5) [[0. -14. -20. -22.] [-14. -18. -20. -20.] [-20. -20. -18. -14.] [-22. -20. -14. 0.]]	

Policy Evaluation : Iteration 2	
1) [[0. -1. -2. -3.] [-1. -2. -3. -15.] [-2. -3. -15. -1.] [-3. -15. -1. 0.]]	2) [[0. -1. -2. -3.] [-1. -2. -3. -2.] [-2. -3. -2. -1.] [-3. -2. -1. 0.]]
Policy Evaluation : Iteration 3	
1) [[0. -1. -2. -3.] [-1. -2. -3. -2.] [-2. -3. -2. -1.] [-3. -2. -1. 0.]]	

Policy Improvement : Iteration 1	
state	[Up, down, left, right]
1	[(1, 0.25), (2, 0.25), (3, 0.25), (4, 0.25)]
2	[(1, 0.25), (2, 0.25), (3, 0.25), (4, 0.25)]
3	[(1, 0.25), (2, 0.25), (3, 0.25), (4, 0.25)]
4	[(1, 0.25), (2, 0.25), (3, 0.25), (4, 0.25)]
5	[(1, 0.25), (2, 0.25), (3, 0.25), (4, 0.25)]
6	[(1, 0.25), (2, 0.25), (3, 0.25), (4, 0.25)]
7	[(1, 0.25), (2, 0.25), (3, 0.25), (4, 0.25)]
8	[(1, 0.25), (2, 0.25), (3, 0.25), (4, 0.25)]
9	[(1, 0.25), (2, 0.25), (3, 0.25), (4, 0.25)]
10	[(1, 0.25), (2, 0.25), (3, 0.25), (4, 0.25)]
11	[(1, 0.25), (2, 0.25), (3, 0.25), (4, 0.25)]
12	[(1, 0.25), (2, 0.25), (3, 0.25), (4, 0.25)]
13	[(1, 0.25), (2, 0.25), (3, 0.25), (4, 0.25)]
14	[(1, 0.25), (2, 0.25), (3, 0.25), (4, 0.25)]
Policy Improvement : Iteration 2	
state	[Up, down, left, right]
1	[(1, 0.0), (2, 0.0), (3, 1.0), (4, 0.0)]
2	[(1, 0.0), (2, 0.0), (3, 1.0), (4, 0.0)]
3	[(1, 0.0), (2, 0.0), (3, 1.0), (4, 0.0)]
4	[(1, 1.0), (2, 0.0), (3, 0.0), (4, 0.0)]
5	[(1, 0.5), (2, 0.0), (3, 0.5), (4, 0.0)]
6	[(1, 0.0), (2, 0.0), (3, 1.0), (4, 0.0)]
7	[(1, 0.0), (2, 1.0), (3, 0.0), (4, 0.0)]
8	[(1, 1.0), (2, 0.0), (3, 0.0), (4, 0.0)]
9	[(1, 1.0), (2, 0.0), (3, 0.0), (4, 0.0)]
10	[(1, 0.0), (2, 1.0), (3, 0.0), (4, 0.0)]
11	[(1, 0.0), (2, 1.0), (3, 0.0), (4, 0.0)]

12	[(1, 1.0), (2, 0.0), (3, 0.0), (4, 0.0)]
13	[(1, 0.0), (2, 0.0), (3, 0.0), (4, 1.0)]
14	[(1, 0.0), (2, 0.0), (3, 0.0), (4, 1.0)]

Policy Improvement : Iteration 3

state	[Up, down, left, right]
1	[(1, 0.0), (2, 0.0), (3, 1.0), (4, 0.0)]
2	[(1, 0.0), (2, 0.0), (3, 1.0), (4, 0.0)]
3	[(1, 0.0), (2, 0.5), (3, 0.5), (4, 0.0)]
4	[(1, 1.0), (2, 0.0), (3, 0.0), (4, 0.0)]
5	[(1, 0.5), (2, 0.0), (3, 0.5), (4, 0.0)]
6	[(1, 0.25), (2, 0.25), (3, 0.25), (4, 0.25)]
7	[(1, 0.0), (2, 1.0), (3, 0.0), (4, 0.0)]
8	[(1, 1.0), (2, 0.0), (3, 0.0), (4, 0.0)]
9	[(1, 0.25), (2, 0.25), (3, 0.25), (4, 0.25)]
10	[(1, 0.0), (2, 0.5), (3, 0.0), (4, 0.5)]
11	[(1, 0.0), (2, 1.0), (3, 0.0), (4, 0.0)]
12	[(1, 0.5), (2, 0.0), (3, 0.0), (4, 0.5)]
13	[(1, 0.0), (2, 0.0), (3, 0.0), (4, 1.0)]
14	[(1, 0.0), (2, 0.0), (3, 0.0), (4, 1.0)]

Policy Improvement : Iteration 4

state	[Up, down, left, right]
1	[(1, 0.0), (2, 0.0), (3, 1.0), (4, 0.0)]
2	[(1, 0.0), (2, 0.0), (3, 1.0), (4, 0.0)]
3	[(1, 0.0), (2, 0.5), (3, 0.5), (4, 0.0)]
4	[(1, 1.0), (2, 0.0), (3, 0.0), (4, 0.0)]
5	[(1, 0.5), (2, 0.0), (3, 0.5), (4, 0.0)]
6	[(1, 0.25), (2, 0.25), (3, 0.25), (4, 0.25)]
7	[(1, 0.0), (2, 1.0), (3, 0.0), (4, 0.0)]
8	[(1, 1.0), (2, 0.0), (3, 0.0), (4, 0.0)]
9	[(1, 0.25), (2, 0.25), (3, 0.25), (4, 0.25)]
10	[(1, 0.0), (2, 0.5), (3, 0.0), (4, 0.5)]
11	[(1, 0.0), (2, 1.0), (3, 0.0), (4, 0.0)]
12	[(1, 0.5), (2, 0.0), (3, 0.0), (4, 0.5)]
13	[(1, 0.0), (2, 0.0), (3, 0.0), (4, 1.0)]
14	[(1, 0.0), (2, 0.0), (3, 0.0), (4, 1.0)]

Values Iteration's Iteration

1) [[0. -1. -1. -1.] [-1. -1. -1. -1.] [-1. -1. -1. -1.] [-1. -1. -1. 0.]]	2) [[0. -1. -2. -2.] [-1. -2. -2. -2.] [-2. -2. -2. -1.] [-2. -2. -1. 0.]]

```
3) [[ 0. -1. -2. -3.]
    [-1. -2. -3. -2.]
    [-2. -3. -2. -1.]
    [-3. -2. -1.  0.]]
```

Value Iteration Policy

state	[Up, down, left, right]
1	[(1, 0.0), (2, 0.0), (3, 1.0), (4, 0.0)]
2	[(1, 0.0), (2, 0.0), (3, 1.0), (4, 0.0)]
3	[(1, 0.0), (2, 0.5), (3, 0.5), (4, 0.0)]
4	[(1, 1.0), (2, 0.0), (3, 0.0), (4, 0.0)]
5	[(1, 0.5), (2, 0.0), (3, 0.5), (4, 0.0)]
6	[(1, 0.25), (2, 0.25), (3, 0.25), (4, 0.25)]
7	[(1, 0.0), (2, 1.0), (3, 0.0), (4, 0.0)]
8	[(1, 1.0), (2, 0.0), (3, 0.0), (4, 0.0)]
9	[(1, 0.25), (2, 0.25), (3, 0.25), (4, 0.25)]
10	[(1, 0.0), (2, 0.5), (3, 0.0), (4, 0.5)]
11	[(1, 0.0), (2, 1.0), (3, 0.0), (4, 0.0)]
12	[(1, 0.5), (2, 0.0), (3, 0.0), (4, 0.5)]
13	[(1, 0.0), (2, 0.0), (3, 0.0), (4, 1.0)]
14	[(1, 0.0), (2, 0.0), (3, 0.0), (4, 1.0)]

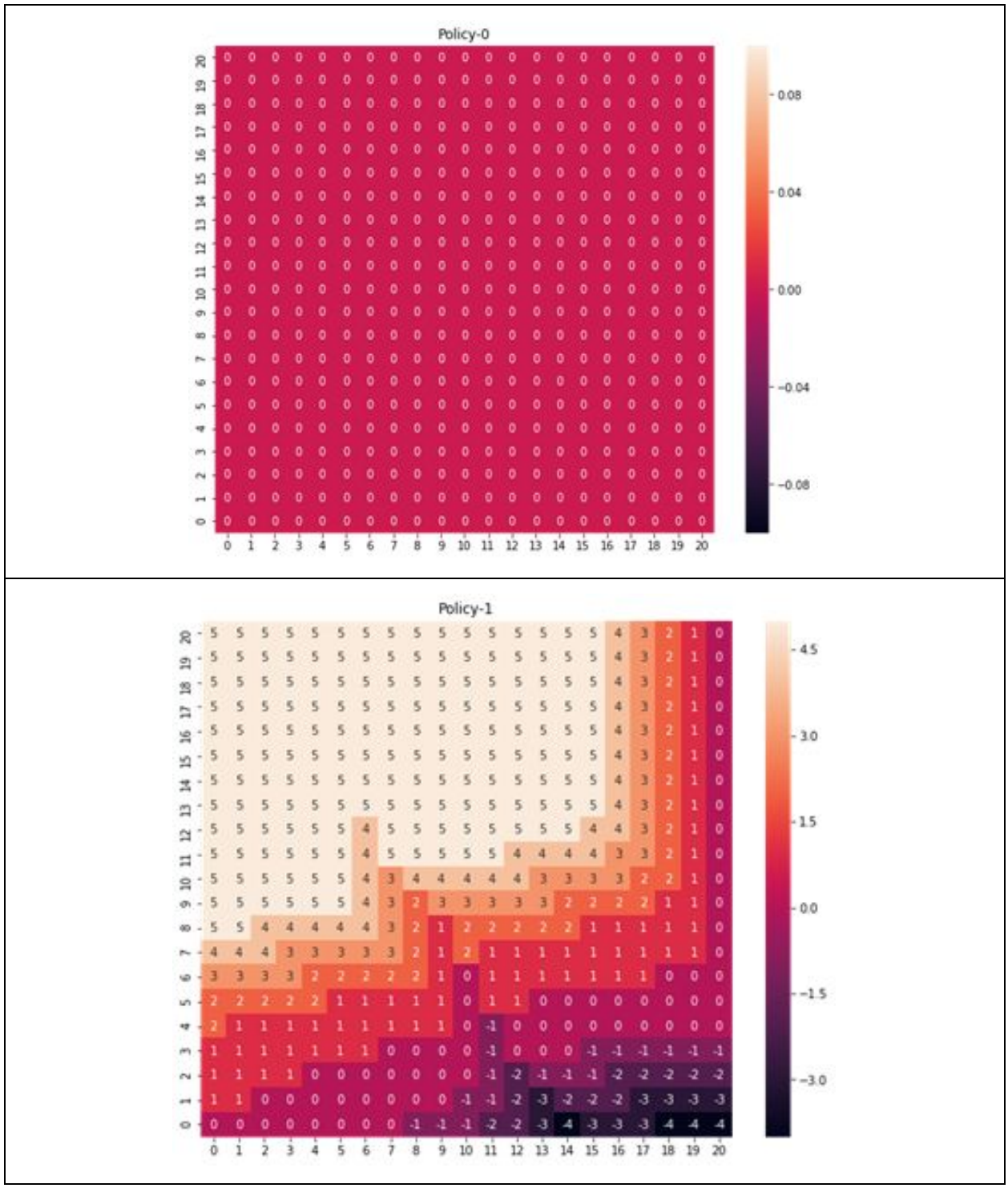
Question 7

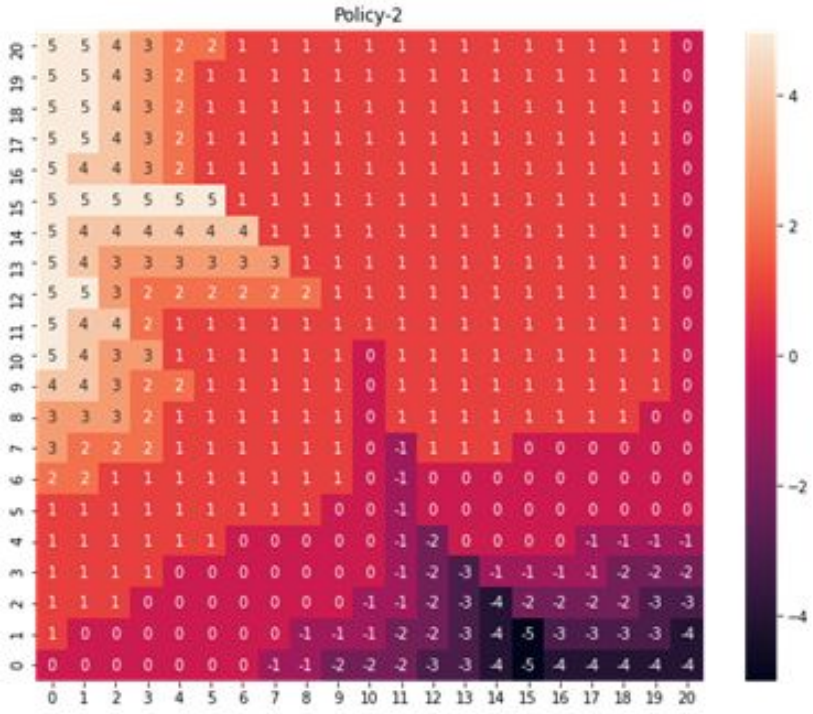
Policy Iteration for Jack Rental Car Problem.

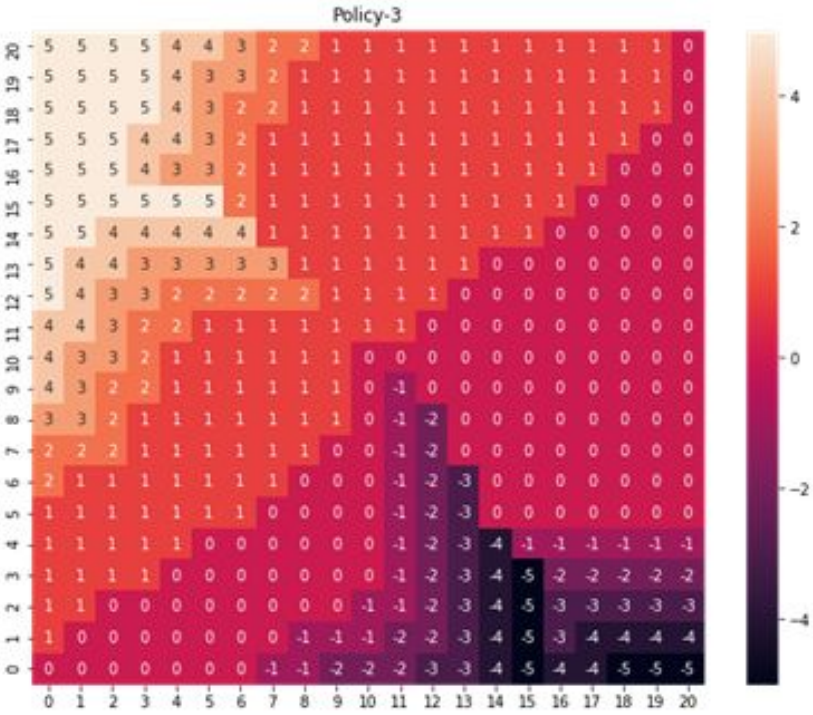
Assumption: While calculation the next state and reward we have taken a constant return of 3 at location A and 2 at location B.

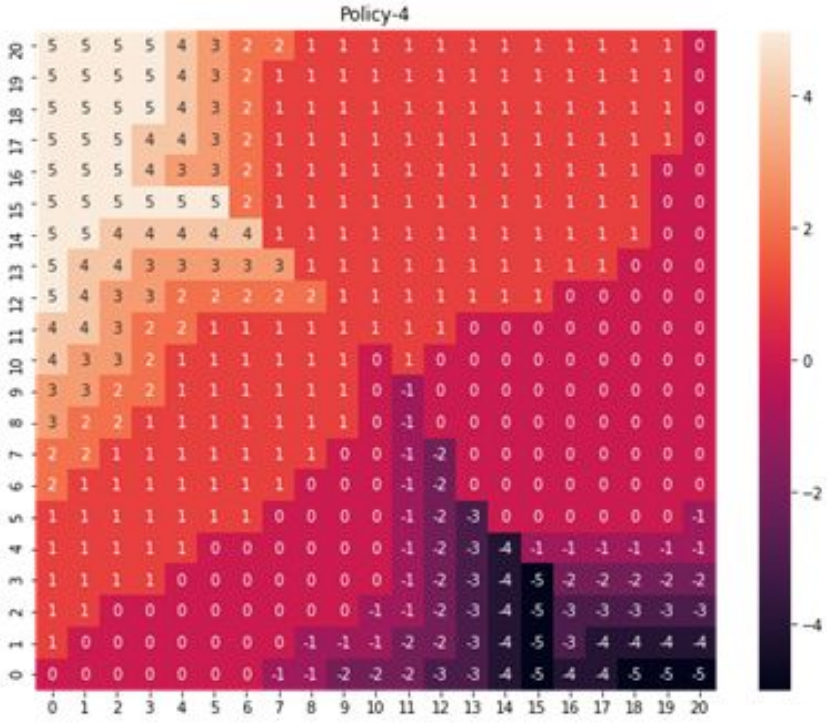
For detailed describtion see Question7_JackRental.ipynb.

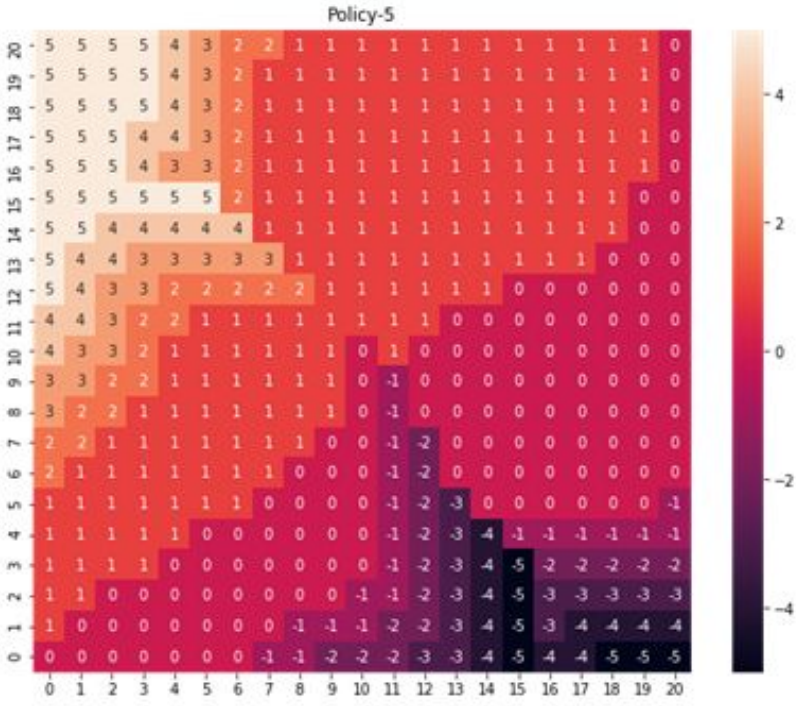
Below are the policy for every iteration of Policy Iteration


Policy-0
0.08
0.04
0.00
−0.04
−0.08
Policy-1
4.5
3.0
1.5
0.0
−1.5
−3.0

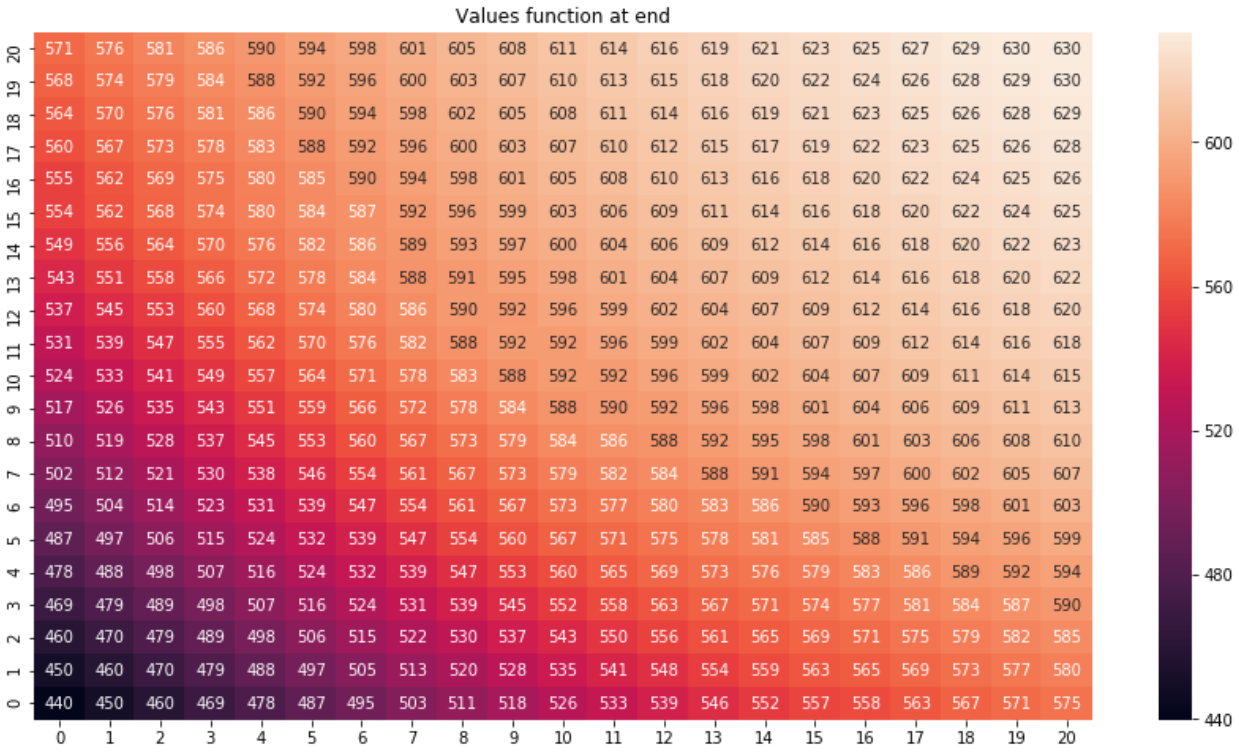

Policy-2


Policy-3

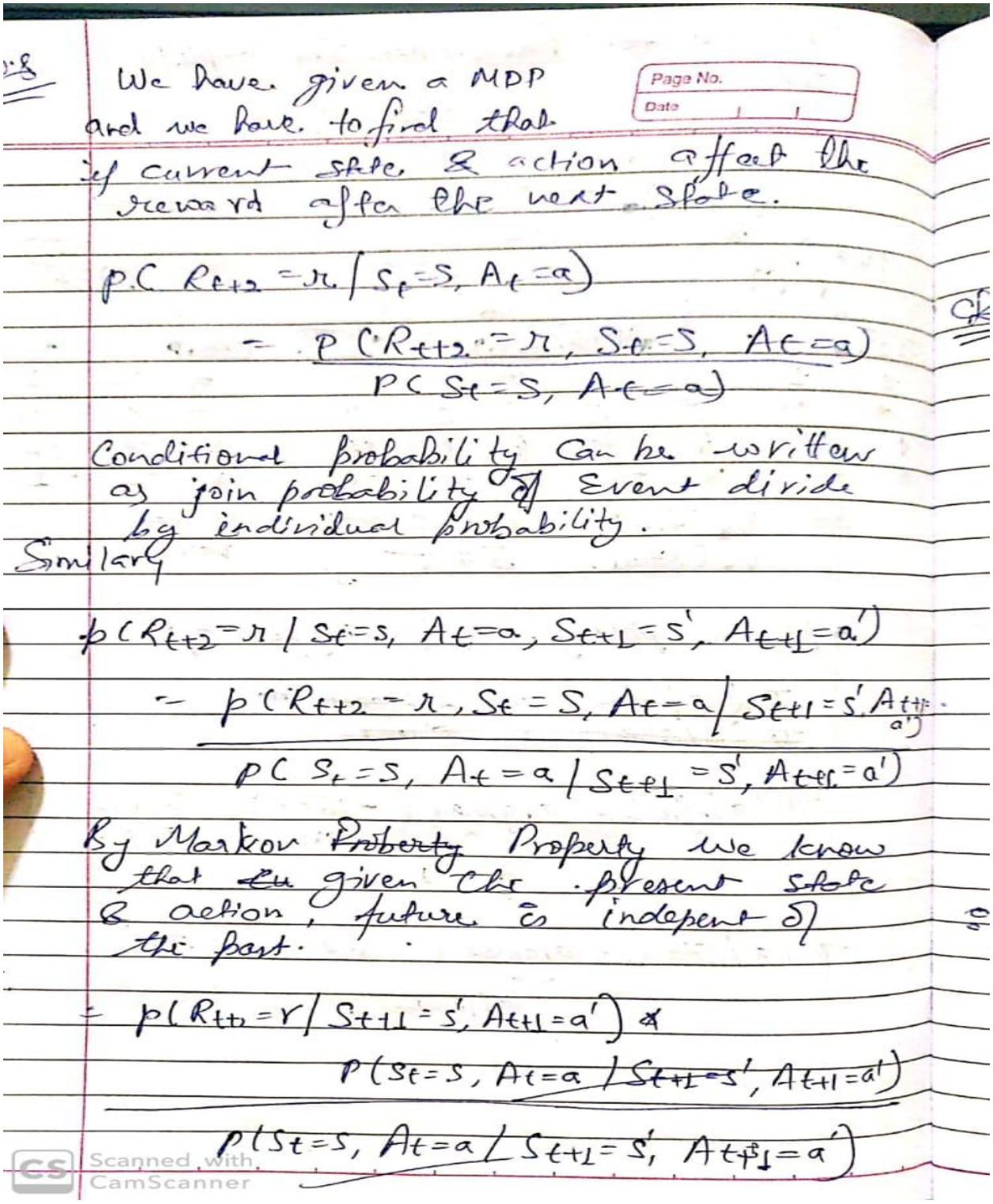




Value function after all the iteration:


Values function at end
20 571 576 581 586 590 594 598 601 605 608 611 614 616 619 621 623 625 627 629 630 630
19 568 574 579 584 588 592 596 600 603 607 610 613 615 618 620 622 624 626 628 629 630
18 564 570 576 581 586 590 594 598 602 605 608 611 614 616 619 621 623 625 626 628 629
17 560 567 573 578 583 588 592 596 600 603 607 610 612 615 617 619 622 623 625 626 628
16 555 562 569 575 580 585 590 594 598 601 605 608 610 613 616 618 620 622 624 625 626
15 554 562 568 574 580 584 587 592 596 599 603 606 609 611 614 616 618 620 622 624 625
14 549 556 564 570 576 582 586 589 593 597 600 604 606 609 612 614 616 618 620 622 623
13 543 551 558 566 572 578 584 588 591 595 598 601 604 607 609 612 614 616 618 620 622
12 537 545 553 560 568 574 580 586 590 592 596 599 602 604 607 609 612 614 616 618 620
11 531 539 547 555 562 570 576 582 588 592 592 596 599 602 604 607 609 612 614 616 618
10 524 533 541 549 557 564 571 578 583 588 592 592 596 599 602 604 607 609 611 614 615
9 517 526 535 543 551 559 566 572 578 584 588 590 592 596 598 601 604 606 609 611 613
8 510 519 528 537 545 553 560 567 573 579 584 586 588 592 595 598 601 603 606 608 610
7 502 512 521 530 538 546 554 561 567 573 579 582 584 588 591 594 597 600 602 605 607
6 495 504 514 523 531 539 547 554 561 567 573 577 580 583 586 590 593 596 598 601 603
5 487 497 506 515 524 532 539 547 554 560 567 571 575 578 581 585 588 591 594 596 599
4 478 488 498 507 516 524 532 539 547 553 560 565 569 573 576 579 583 586 589 592 594
3 469 479 489 498 507 516 524 531 539 545 552 558 563 567 571 574 577 581 584 587 590
2 460 470 479 489 498 506 515 522 530 537 543 550 556 561 565 569 571 575 579 582 585
1 450 460 470 479 488 497 505 513 520 528 535 541 548 554 559 563 565 569 573 577 580
0 440 450 460 469 478 487 495 503 511 518 526 533 539 546 552 557 558 563 567 571 575
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
600
560
520
480
440

Question 8



We have given a MDP and we have to find that if current state & action effect the reward after the next state.

$$p(R_{t+2} = r \mid S_t = S, A_t = a)$$

$$= \frac{p(R_{t+2} = r, S_t = S, A_t = a)}{p(S_t = S, A_t = a)}$$

Conditional probability can be written as join probability of event divide by individual probability.

Similary

$$p(R_{t+2} = r \mid S_t = s, A_t = a, S_{t+1} = s', A_{t+1} = a')$$

$$= \frac{p(R_{t+2} = r, S_t = S, A_t = a \mid S_{t+1} = s', A_{t+1} = a')}{p(S_t = S, A_t = a \mid S_{t+1} = s', A_{t+1} = a')}$$

By Markov ~~Property~~ Property we know that ~~in~~ given the present state & action, future is independent of the past.

$$= p(R_{t+2} = r \mid S_{t+1} = s', A_{t+1} = a') \times \frac{p(S_t = S, A_t = a \mid S_{t+1} = s', A_{t+1} = a')}{p(S_t = S, A_t = a \mid S_{t+1} = s', A_{t+1} = a')}$$

Scanned with CamScanner

$$= P(R_{t+1} = r \mid S_{t+1} = s', A_{t+1} = a)$$

Page No.
Date

So we can see given the current state (S_{t+1}) then R_{t+2} and (S_t, A_t) are conditionally independent.

Scanned with CamScanner

Question 9

Q9. Derive a Expression for $E[R_{t+2} | S_t = S, A_t = a]$ in terms of PMF

$$E[R_{t+2} | S_t = S, A_t = a]$$

$$= \sum_{r \in R} r * p[R_{t+2} = r | S_t = S, A_t = a]$$

$$= \sum_{r \in R} r * \sum_{s' \in S} p(R_{t+2} = r | S_t = S, A_t = a, S_{t+1} = s', A_{t+1} = a')$$

if we ~~argi~~ marginalize over S_{t+1} & A_{t+1} we get the same equation as above.

In the summation we have not included A_{t+1} because we assume a deterministic policy.

$$\therefore E[R_{t+2} | S_t = S, A_t = a] = \sum_{r \in R} r * \sum_{s' \in S} p(R_{t+2} = r | S_{t+1} = s', A_{t+1} = a')$$

$$= \sum_{r \in R} r * \sum_{s' \in S} \sum_{s'' \in S} p(S_{t+2} = s'', R_{t+2} = r | S_{t+1} = s', A_{t+1} = a')$$

Scanned with CamScanner

Question 10

Ques 10 $V_\pi(s) = E_\pi[G_t \mid S_t = s]$

$V_\pi(s) = E_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} \ldots \mid S_t = s]$

$= E_\pi[R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3} + \gamma^2 R_{t+4} \ldots) \mid S_t = s]$

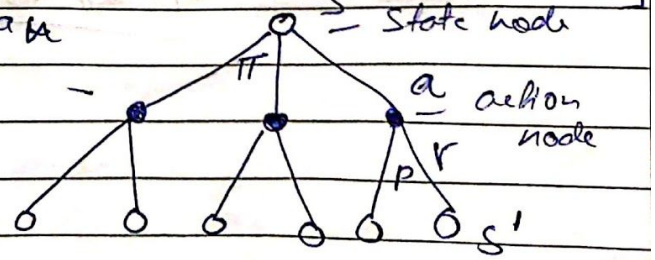
$= E_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s]$

, because $G_{t+1} = R_{t+2} + \gamma R_{t+3} + \gamma^2 R_{t+4} \ldots$

Now as expectation is over policy ~~pie~~ π we will open the expectation

$$= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r \mid s, a)\left[r + \gamma E_\pi[G_{t+1} \mid S_{t+1} = s']\right]$$

The above expansion can be understood with the help of this back up diagram



Scanned with CamScanner

From a state with respect to some particular policy we choose a action. that will take us to action node. From action node environment can blow me to any of the new value state s'. So for computing we have to consider all possibility.

$$V_\pi(s) = \sum_a \pi(a/s) \sum_{s',r} p(s',r/s,a) \left[r + \gamma E_\pi [G_{t+1} / S_{t+1} = s'] \right]$$

By definition definition of V_π we know

$$V_\pi(s') = E_\pi [G_t / S_t = s']$$

So above equation becom

$$V_\pi(s) = \sum_a \pi(a/s) \sum_{s',r} p(s'r/s,a) \left[r + \gamma V_\pi(s') \right]$$

for all $s \in S$

This is bellman equation of $V_\pi(s)$ in term of $V_\pi(s)$.

Scanned with CamScanner

Question 11

Q11 $R_1 = 2$, $R_2 = -1$, $R_3 = 10$

$R_4 = -3$

Page No.
Date

We have to calculate discounted return for each time step.

$G_t = R_{t+1} + \gamma G_{t+1}$

Return

for ~~test~~ let start from $t = 4$

$G_4 = R_5 + \gamma G_6$

$= 0 + 0 = 0$

for $t = 3$ $G_3 = R_4 + \gamma G_4$

$= -3 + 0.5 \times 0$

$= -3$

for $t = 2$ $G_2 = R_3 + \gamma G_3$

$= 10 + 0.5 \times -3$

$= 10 + -1.5 = 8.5$

for $t = 1$ $G_1 = R_2 + \gamma G_2$

$= -1 + 0.5 \times 8.5$

$= -1 + 4.25 = 3.25$

for $t = 0$ $G_0 = R_1 + \gamma G_1$

$= 2 + 0.5 \times 3.25$

$= 2 + 1.625 = 3.625$

Now suppose if we get a constant reward c then return is

Scanned with CamScanner

$C_{t=0} = R_{t+1} + V R_{t+2} + V^2 R_{t+3}$

Page No. _ _ _
Date / /

$= C + VC + V^2 C + \cdots$

$= C[1 + V + V^2 + \cdots]$

- This forms a GP (Geometric Progression.

Sum of GP when common ratio is less than 1 as here ($V < 1$) $= \frac{a}{1-r}$

here $a = 1 \quad r = r$

So $\frac{C \times 1}{1-r} = \boxed{\frac{C}{1-r}}$

Scanned with CamScanner

Question 12

12 ~~If~~ we have been given optimal state-value function $V_*(s)$.
& we have to find optimal policy.

Optimal policy is from a state which action we should pick so that our expected reward is maximized.

Optimal State-Value function is ~~the~~ expected value ~~our~~ one can achieve from a state following a ~~poli~~ optimal policy.
state

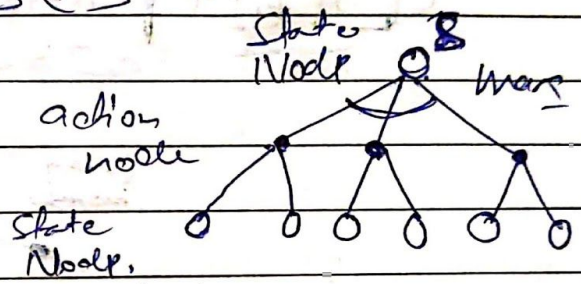
Now, when optimal value is given finding optimal policy is like a one step look ahead.

Scanned with CamScanner

So from a state take all action and see which action is giving maximum return.

$$a = \arg\max_{a} \sum_{s' r} p(s', r / s, a)\left[r + v_*(s')\right]$$

for all $s \in S$

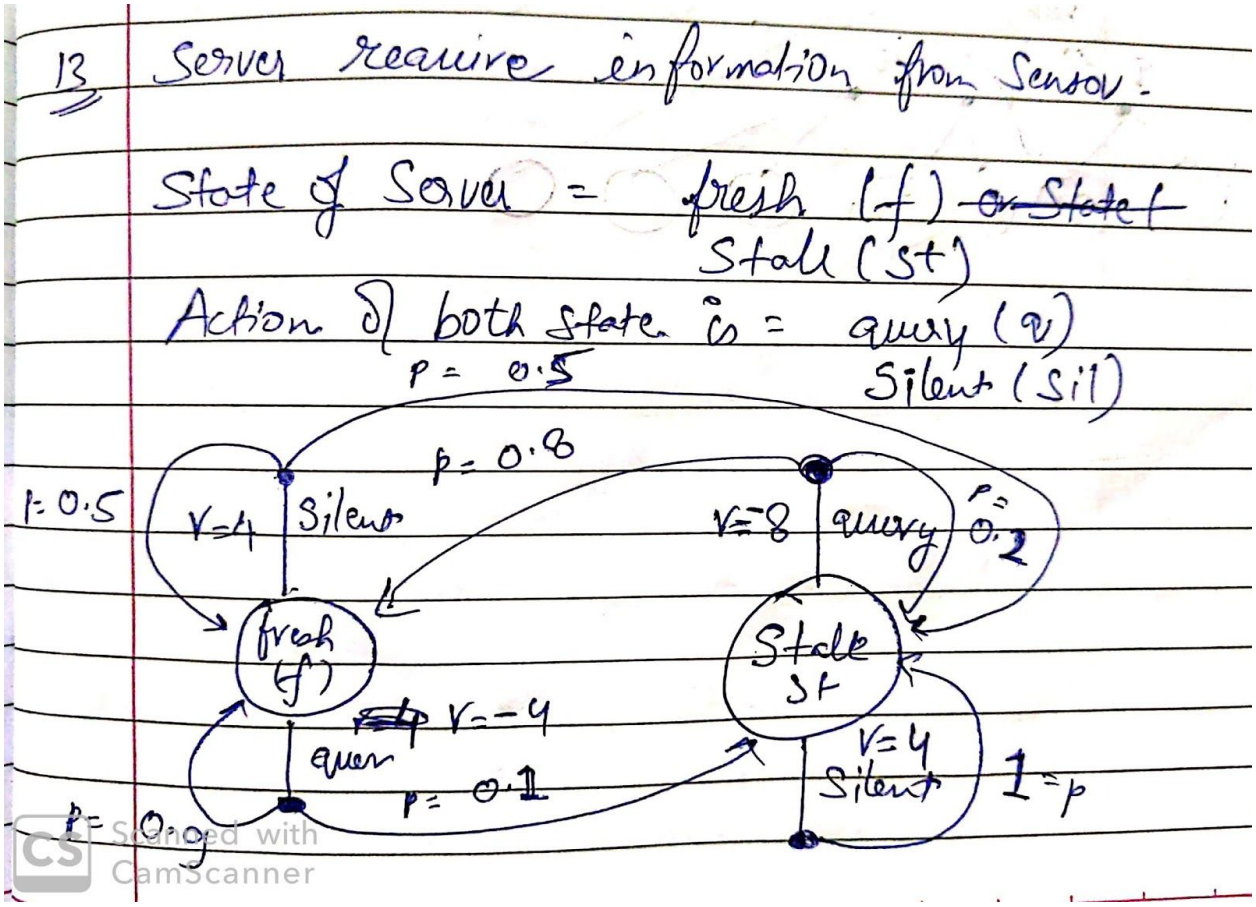


→ From node S take all action and do one step look ahead & see that reward is ar getting

→ Then optimal action is the action that yield maximum reward.

Scanned with CamScanner

Question 13 (a)



Question 13 (b)

13 b Finite horizon Problem
Optimize cost over three time step

Page No.
Date

→ At the end we get +10 reward if we are at fresh state. −10 otherwise

- Discount factor - 0.5

→ we have to find optimal cost.

To find optimal reward we can explore every path till 3 time step and then calculate the reward

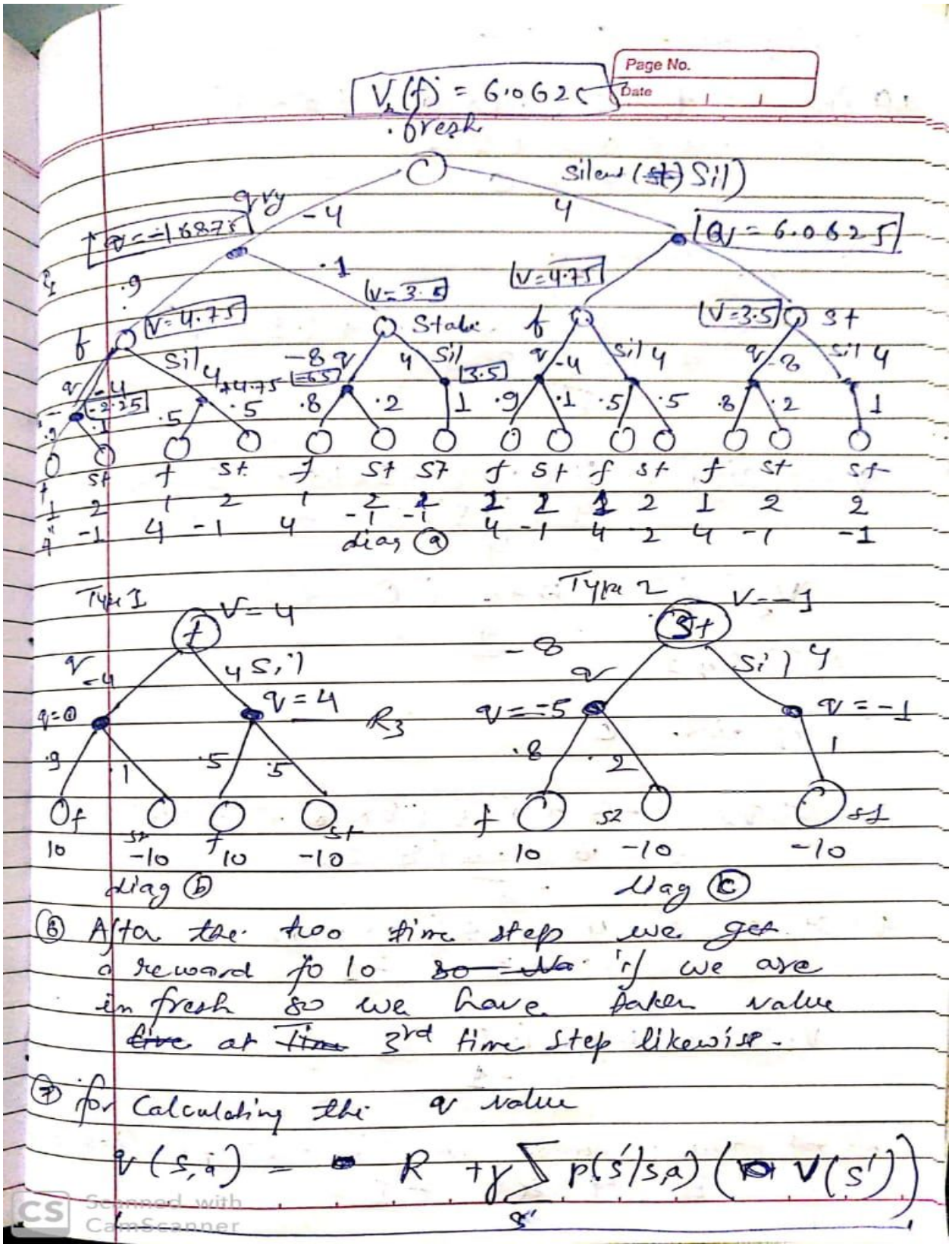
$$V_*(s) = \max_a q_*(s,a)$$

So we can take every action from state and see which gives us the best return.

Assumption

(1) ○ - represent state node ▲ - repres action node
(2) Value from state node to action node is reward
(3) Value over action node to state node represent probability.
(4) The diagram shows search space for two time step.
(5) Tree diagram for 3rd is shown seperately.

Scanned with CamScanner



(6) After the two time step we get a reward fo 10 ~~so = Na~~ if we are in fresh so we have taken value ~~five~~ at ~~Time~~ 3rd time step likewise.

(7) for Calculating the q value

$$q(s,a) = R + \gamma \sum_{s'} p(s'|s,a)\left(V(s')\right)$$

Scanned with CamScanner

Solving diag b first

$q(\theta, q) = -4 + 0.5(.9 \times 10 + .1 \times -10)$

$= -4 + 0.5(9 - 1)$

$-4 + 4 = 0$

Solving diagram c

$= -8 + .5((.8 \times 10) + (.2 \times 10))$

$= -8 + .5(8 - 2)$

$= -5$

$q(f, sil) = 4 + .5(.5 \times 10 + .5 \times -10)$

$4 + .5(5 - 5)$

$= 4$

$V_s(f) = \max(0, 4)$

$= 4$

At TimStep 3

Solving diag c

$q(St, q) = -8 + 0.5((.8 \times 10) + (.2 \times 10))$

$= -8 + .5(8 - 2)$

$= -5$

$q(St, sil) = 4 + .5(1 \times -10)$

$= 4 - 5 = -1$

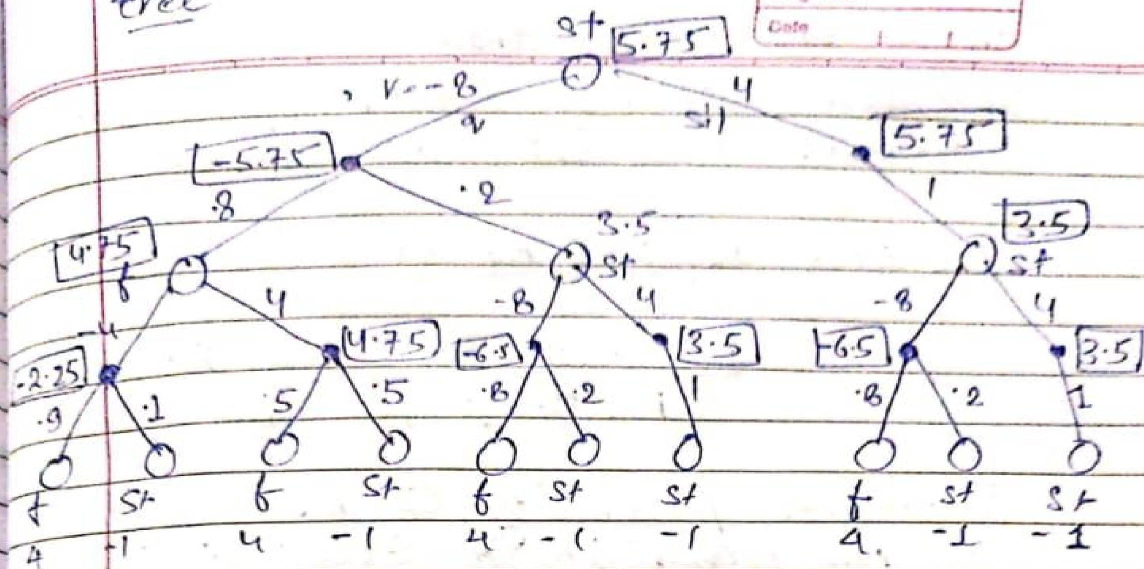
$V_s(s) = \max(-5, -1)$

$= -1$

So Value of nodes at the end for diag a is 4 & -1

Scanned with CamScanner

Like wise ~~to~~. I have filled the complete tree



So $V_*(fresh) = 6.0625$ } optimal Value
$V_*(Stale) = 5.75$

Optimal policy: $\pi_*(s) = \arg\max_a q^*(s,a)$

~~$\pi(a/fresh) = $ Silent~~
~~$\pi(a/stale) = $ Silent~~

$\pi(a/fresh) = \arg\max_a \left(\underset{cry}{-16875}, \underset{silent}{8.0625} \right)$

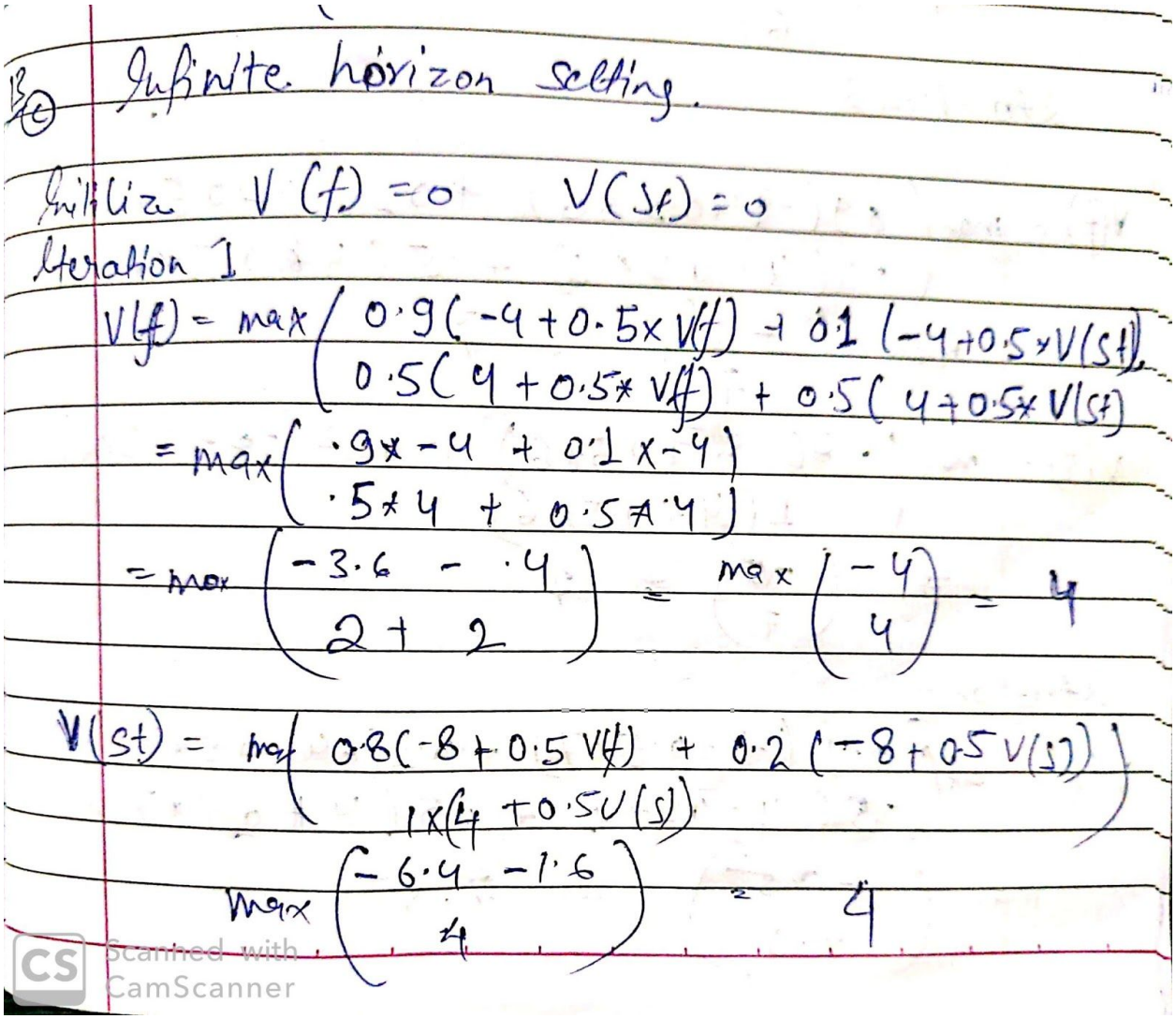
= Silent

$\pi(a/Stale) = \arg\max_a (-5.75, 5.75)$

= Silent.

Scanned with CamScanner

Question 13 (c)



Iteration 2

$$V(f) = \max\begin{pmatrix} 0.9(-4+0.5\,V(f)) + 0.1(-4+0.5\,V(s)) \\ .5(4+0.5\,V(f)) + .5(4+0.5\,V(s)) \end{pmatrix}$$

$$\max\begin{pmatrix} 0.9(-4+0.5\times4) + 0.1(-4+0.5*4) \\ .5(4+0.5\times4) + .5(4+0.5\times4 \end{pmatrix}$$

$$\max = \begin{pmatrix} 0.9*-2 + 0.1*-2 \\ 0.5*6 + .5*6 \end{pmatrix}$$

$$= \begin{pmatrix} -2 \\ 6 \end{pmatrix} = 6$$

$$V(s) = \max\begin{pmatrix} .8(-8+0.5*4) + 0.2(-8+0.5\times4) \\ 1\times(4+0.5*4) \end{pmatrix}$$

$$= \begin{pmatrix} .8(-6) + 0.2\times-6 \\ 4+2 \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} -6 \\ 6 \end{pmatrix} = 6$$

Iteration 3

$$V(f) = \max\begin{pmatrix} 0.9(-4+0.5*6) + 0.1(-4+0.5*6) \\ .5(4+0.5*6) + .5(4*0.5*6) \end{pmatrix}$$

$$\max\begin{pmatrix} -1 \\ 7 \end{pmatrix} = 7$$

$$V(s) = \max\begin{pmatrix} .8(-8+0.5*6) + 0.2(-8+0.5*6) \\ 1(4+0.5*6) \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} -5 \\ 7 \end{pmatrix} = 7$$

Iteration 4

$$V(f) = \max\begin{pmatrix} 0.9(-4+0.5*7) + 0.1(-4+0.5*7) \\ .5(4+0.5*7) + .5(4*0.5*7 \end{pmatrix}$$

$$\max\begin{pmatrix} -0.5 \\ 7.5 \end{pmatrix} = 7.5$$

Scanned with CamScanner

$$V(S) = \max \begin{pmatrix} .8(-8+0.5\times 7) + 0.2(-8+0.5*7) \\ 1(4+0.5*7) \end{pmatrix}$$

Page No.
Date

$$= \max \begin{pmatrix} -4.5 \\ 7.5 \end{pmatrix} = 7.5$$

Policy After Converge is

State	Action query	Action silent
1	0.0	1.0
2	0.0	1.0

Scanned with CamScanner

Q. 13 3 Policy Iteration

Initilizing

$V(f) = 0 \quad V(St) = 0$

Policy ~~After Initi~~ : Equi probable policy

$\pi(\text{quem}/f) = 0.5 \quad \pi(\text{quem}/St) = 0.5$

$\pi(\text{silent}/f) = 0.5 \quad \pi(\text{silent}/St) = 0.5$

Iteration 1

Policy Evaluation $= V(s) = \sum_{s' r} p(s', r | s, \pi(s)) [r + V(s')]$

Policy Evaluation Iteration 1

$V(f) = 0.5 [0.9(-4 + 0.5 * V(f)) + 0.1(-4 + 0.5 V(St))]$

$+ 0.5 [0.5(4 + 0.5 * V(f)) + 0.5(4 + 0.5 V(St))]$

$= 0.5 [0.9(-4) + 0.1(-4)] + 0.5 [0.5(4) + 0.5(4)]$

$\Rightarrow -2 + 2 = 0$

$V(St) = 0.5 [0.8(-8 + 0.5 V(f)) + 0.2(-8 + 0.5 V(St))]$

$+ 0.5 [4 + 0.5 V(St)]$

Scanned with CamScanner

$V(St) = 0.5[0.8(-8) + 0.2(-8)]$

$+ 0.5[4]$

$\Rightarrow 0.5(-8) + 0.5(4)$

$-4 + 2 = -2$

Policy Evaluation Iteration 2

$V(f) = 0.5[0.9(-4+0) + 0.1(-4+0.5(-2)]$

$+ 0.5[0.5(4+0.5*0) + 0.5(4+0.5(-2)]$

$0.5[0.9(-4) + 0.1(-5)]$

$+ 0.5[2 + 1.5]$

$\rightarrow 0.5[-3.6 - .5] + 0.5(3.5)$

$\Rightarrow -2.05 + 1.75$

$= -0.3$

$V(Sf) = 0.5[0.8(-8 + 0.5(-0.3)) + 0.2(-8 + 0.5(-2)]$

$+ 0.5(4 + 0.5(-2))$

$\Rightarrow 0.5[0.8(-8 - 0.15) + 0.2(-8-1)] +$

$0.5(4-1)$

$\Rightarrow 0.5[-6.52 - 1.8] + 1.5$

$\Rightarrow -4.16 + 1.5$

$= -2.66$

Policy Evaluation Iteration 3

$V(f) = 0.5[0.9(-4 + 0.5(-.3) + 0.1(-4 + 0.5(-2.66)]$

$+ 0.5[0.5[4 + 0.5(-.3)] + 0.5(4 + 0.5[-2.66)]$

Scanned with CamScanner

$= 0.5[-3.735 + -.533] +$

$= 0.5[1.925 + 1.335]$

$\Rightarrow -2.134 + 1.63$

$= -.504$

$V(s_t) = 0.5[.8(-8 + 0.5(-.504)) + 0.2(-8 + .5(-2.66)$

$+ 0.5[4 + 0.5(-2.66)]$

$= 0.5[-6.6016 + -1.866]$

$+ 0.5[4 - 1.33]$

$= -4.2338 + 1.335$

$= -2.8988.$

Policy Improvement.

$\pi(a/s_t) = \underset{a}{\max}\ avg \begin{pmatrix} 0.9(-4 + .5\times(-.504)) + 0.1(-4 + .5(-.289) \\ 0.5(4 + 0.5\times(-.504)) + 0.5(4 + .5(-2.89 \end{pmatrix}$

$= \underset{a}{\max}\ avg \begin{pmatrix} 0.9(-4 + -0.252) + 0.1(-4 - .1445) \\ 0.5(4 + -0.252) + 0.5(4 - .1445) \end{pmatrix}$

= action 2 ~~is maxtac~~ i'e, Silent.

$\pi(a/s_t) = \underset{a}{\max}\ avg \begin{pmatrix} .8(-8 + 0.5 * (-0.504) + 0.2(-8 + .5(-.289) \\ 4 + 0.5(-.289) \end{pmatrix}$

- action 2 ie silent is better

$\pi(a/s=s_t) = 2$ (Silent)

$\pi(a/s=s_t) = 2$ (Silent)

Scanned with CamScanner

Question 14

14

Infinite horizon discounted problem

To show policy improvement step improves the policy or the current policy is optimal policy (in terms of cost)

$$V_\pi(s) = E[G_t / S_t = s]$$

Goal of the agent in reinforcement learning is to maximize the reward or minimize the cost

$$V_\pi(s) = E[-C_{t+1} + \gamma(-C_{t+2}) + \gamma^2(-C_{t+3}) \ldots / S_t = s]$$

$$= E[-C_{t+1} - \gamma C_{t+2} - \gamma^2 C_{t+3} \ldots / S_t = s]$$

to infinity

$$= -E[C_{t+1} + \gamma C_{t+2} + \gamma^2 C_{t+3} \ldots / S_t = s]$$ imp

$$= -E[C_{t+1} - (+\gamma)[(C_{t+2} + \gamma^2 C_{t+3} \ldots / S_t = s]$$

$$= -E[C_{t+1} - \gamma V_\pi(S_{t+1})] / S_t = s]$$

$$= E[-C_{t+1} + \gamma V_\pi(S_{t+1}) / S_t = s]$$

When we want to improve our policy we do one step look ahead and then be greedy in what we want to choose.

$$q_\pi(s,a) = E[-C_{t+1} + \gamma V_\pi(S_{t+1}) / S_t = s, A_t = a]$$

Scanned with CamScanner

$$= \sum_{s',r} p(s', r | s, a) \left[-c + \gamma V_\pi(s') \right]$$

Page No.
Date

So when choosing the action we will choose the maximum $q_\pi(s, a)$
It means that it would be better to ~~How~~ select a then to follow π all the time.
Let π & π' be any pair of deterministic policy such that, for all $s \in S$

$$q_\pi(s, \pi'(s)) \geq V_\pi(s) \quad \text{— (1)}$$

Then we have to ~~prof~~ prove that policy π is as good as or better than π.
i.e. $V_{\pi'}(s) \geq V_\pi(s)$ — (2)

Lets try to prove equation (2). If it is true then we can say that policy Improvement ~~iteration~~ step improves the policy.

$$V_\pi \leq q_\pi(s, \pi'(s))$$
$$= E[-C_{t+1} + \gamma V_\pi(S_{t+1}) | S_t = s, A_t = \pi'(s)]$$
$$= E_{\pi'}[-C_{t+1} + \gamma V_\pi(S_{t+1})]$$
$$\leq E_{\pi'}[-C_{t+1} + \gamma q_\pi(S_{t+1}, \pi'(S_{t+1})) | S_t]$$
$$= E_{\pi'}[-C_{t+1} + \gamma E[-C_{t+2} + \gamma V_\pi(S_{t+2} | S_{t+1}, A_{t+1} = \pi'(S_{t+1}))] | S_t = s]$$
$$= E_{\pi'}[-C_{t+1} + \gamma C_{t+2} + \gamma^2 V_\pi(S_{t+2}) | S_t = s]$$
$$\leq E_{\pi'}[-C_{t+1} - \gamma C_{t+2} - \gamma^2 C_{t+3} - \gamma^3 V_\pi(S_{t+3}) | S_t = s]$$

CS Scanned with CamScanner

So] we will continue

$\leq E_{\pi'}[\,= C_{t+1} - \gamma(C_{t+2} -$

$\gamma^2(C_{t+3} - \gamma^2(C_{t+4} + \cdots \mid S_t = s]$

$= V_{\pi'}(s)$

Now Suppose that new policy is as good as, but not better than the old policy π.

Then $V_\pi = V_{\pi'}$

$\pi'(s) = \arg\max_a q_\pi(s, a)$

$= \arg\max_a E[R_{t+1} + \gamma V_\pi(S_{t+1}) \mid S_t = s, A_t = a]$

$= \arg\max_a \sum_{s', r} p(s', r \mid s, a)[r + \gamma V_\pi(s')]$

$V_{\pi'}(s) = \max_a E[R_{t+1} + \gamma V_{\pi'}(S_{t+1}) \mid S_t = s, A_t = a]$

$= \max_a \sum_{s', r} p(s', r \mid s, a)[r + \gamma V_{\pi'}(s')]$

So ~~it~~ value is not ~~chaing~~ Changing

So it is the maximum value

So Optimal State value function:

Scanned with CamScanner

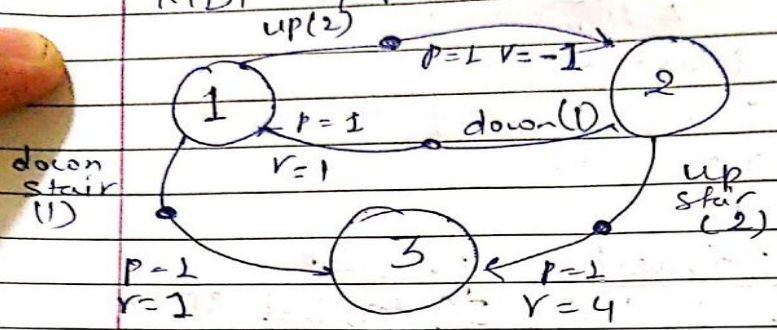
Question 15

Question 15 Stairwell

States = 1, 2, 3 (terminal)

Action from each State except terminal = { down-stair (1), up-stair (2) }

MDP of Problem



Use Policy Iteration to find optimal policy for n=2

Policy Iteration has two step policy evaluation & policy Improvement

Initializing

V = [0, 0]

V(1) = 0 V(2) = 0 V(3) = 0

Policy is a equiprobable policy (P(a/s))

p(1/1) = 0.5 p(1/2) = 0.5

p(2/1) = 0.5 p(2/2) 0.5

Iteration 1

Policy Evaluation $V(s) = \sum_{s'v} p(s', v | s, \pi(s)) [v + V(s')]$

PE1

V(1) = 0.5 (1(1 + 1 * V(3))) + 0.5 (1(-1 + 1 * V(2))

reward, gamma, Value of next

Scanned with CamScanner

$V(1) = 0.5(1) + 0.5(-1)$

$= 0$

Page No.
Date

$V(2) = 0.5(1+0) + 0.5(4+0)$

$= 0.5 + 2 = 2.5$

PE 2 - Policy Evaluation Step 2

$V(1) = 0.5(1+0) + 0.5(-1+2.5)$

$= 0.5 + 0.5 + 0.75 = 1.25$

$V(2) = 0.5(1+1.25) + 0.5(4)$

$\Rightarrow .5(2.25) + 2$

$= 3.125$

Policy Evaluation Step 3

$V(1) = 0.5(1+0) + 0.5(-1+3.125)$

$\Rightarrow 0.5 + 1.0625$

$= 1.5625$

$V(2) = 0.5(1+1.5625) + 0.5(4)$

$= 1.28125 + 2$

$= 3.28125$

Policy Evaluation Step 4

$V(1) = 0.5(1) + 0.5(-1+3.28125)$

$= 0.5 + 1.140625$

1.640625

$V(2) = 0.5(1+1.640625) + 0.5(4)$

$= 1.3203125 + 2$

3.3203125

Scanned with CamScanner

We are doing Policy Evaluation till 4 step only.

Page No.
Date

Policy Improvement

$$a = \arg\max_a \sum_{s'r} p(s', r | s, a)[r + \gamma V(s')]$$

$V(1) = \arg\max \begin{pmatrix} -1 + V(3) \\ -1 + V(2) \end{pmatrix}$

$= \begin{pmatrix} 1 \\ -1 + 3.3203125 \end{pmatrix} = 2$

$V(2) = \arg\max \begin{pmatrix} 1 + 1.640625 \\ 4 \end{pmatrix} = 2$

Poly$\pi(a|1) = 2$ $\quad \pi(a|2) = 2$

Policy Evaluation & new Policy

PEL

$V(1) = -1 + 3.3023125$ → Now we are considering only option 2 as pe policy

$= 2.3023125$

$V(2) = 4$

Policy Evaluation Iteration 2

$V(1) = -1 + 4 = 3$

$V(2) = 4$

Policy Evaluation Iteration 3

$V(1) = -1 + 4 = 3$

$V(2) = 4$

Scanned with CamScanner

As There is no improvement in policy we can do the policy Improvement Step

Page No.
Date

$V(1) = \underset{a}{\text{argmax}} \begin{pmatrix} 1+0 \\ -1+4 \end{pmatrix} \xrightarrow{\text{argmax}} \begin{pmatrix} 1 \\ 3 \end{pmatrix} =$ action 2

$V(2) = \underset{a}{\text{argmax}} \begin{pmatrix} 1+3 \\ 4 \end{pmatrix} = \underset{a}{\text{argmax}} \begin{pmatrix} 4 \\ 4 \end{pmatrix} =$ 0

both action with .5 prob

$\pi(2|1) = 1 \quad \pi(1|2) = 0.5 \quad \pi(2|2) = 0.5$

Policy Evaluation with New Policy

$V(1) = -1 + 4 = 3$

$V(2) = 0.5(1+3) + 0.5(4)$

$= 2 + 2 = 4$

As There is no improvement in State value. So state we again Improve policy

Policy Improvement Step

$V(1) = \underset{a}{\text{argmax}} \begin{pmatrix} 1+0 \\ -1+4 \end{pmatrix} \underset{a}{\text{argmax}} \begin{pmatrix} 1 \\ 3 \end{pmatrix} =$ action 2

$V(2) = \underset{a}{\text{argmax}} \begin{pmatrix} 1+3 \\ 4 \end{pmatrix} = \underset{a}{\text{argmax}} \begin{pmatrix} 4 \\ 4 \end{pmatrix} =$ both action with .5 prob

So final Prob

State	Act. 1	2
1	0.0	1.0
2	0.5	0.5

Scanned with CamScanner