COL333 Assignment-3 Report

<u>Computing Policies:</u>

Formulating taxi domain as an MDP
1_a)

State Space:

Let us suppose the grid is of size nxn.Then there are n^2 possible states if the passenger is in the taxi and if the passenger is not inside the taxi then there are (n^2-for taxi) X (n^2-for passenger) i.e n^4 possible states for this scenario and one more additional state for the end state when the passenger has been dropped at his destination.Therefore in total there are ($n^4 + n^2 +1$) possible states for a nxn Taxi-World. Here in our case n=5 so the total number of states are $5^4 + 5^2 +1=651$.

Action Space:

At each state the taxi can perform one among these 6 actions -move North, move South,move East,move West or Put Down the passenger or pick up the passenger.

Transition and Reward Model:

We defined every state as a tuple of (taxi-location,passenger-location,key) where key is 0 before passenger boards the taxi,key is 1 when passenger is inside the taxi and key is 2 at the end state and clearly when key=1 or key=2 ,the taxi-location and passenger-location are the same.

a)For action North we have

R(s,North,s')=-1
T(s,North,s')=0.85 for taxi to move north
T(s,North,s')=0.05 for taxi to move south
T(s,North,s')=0.05 for taxi to move east
T(s,North,s')=0.05 for taxi to move west

b)For action South we have

R(s,South,s')=-1
T(s,South,s')=0.85 for taxi to move south
T(s,South,s')=0.05 for taxi to move north

T(s,South,s')=0.05 for taxi to move east
T(s,South,s')=0.05 for taxi to move west

c)For action East we have

R(s,East,s')=-1
T(s,East,s')=0.85 for taxi to move east
T(s,East,s')=0.05 for taxi to move south
T(s,East,s')=0.05 for taxi to move north
T(s,East,s')=0.05 for taxi to move west

d)For action West we have

R(s,West,s')=-1
T(s,West,s')=0.85 for taxi to move west
T(s,West,s')=0.05 for taxi to move south
T(s,West,s')=0.05 for taxi to move east
T(s,West,s')=0.05 for taxi to move north

where s is any state in the Taxi-World(except the terminal state) and s’ is obtained by applying corresponding action on s

e)For action Pickup we have

Let state s=(a,b,c)
If a=b and c=0,then R(s,Pickup,s’)=-1
If c=1,then R(s,Pickup,s’)=-1
If a≠b and c=0, the passenger and taxi are not at same position and hence we have that R(s,Pickup,s’)=-10
T(s,Pickup,s')=1

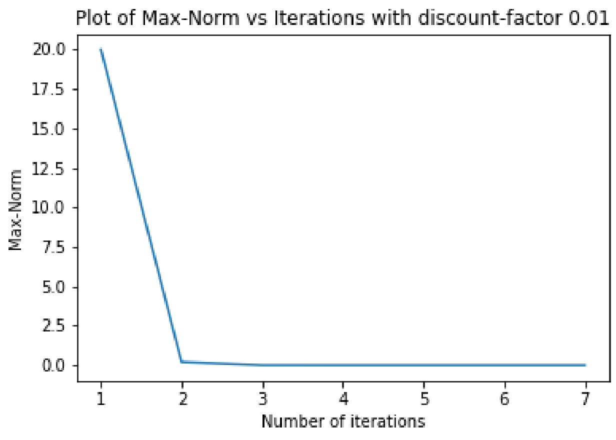
f)For action Put-down we have

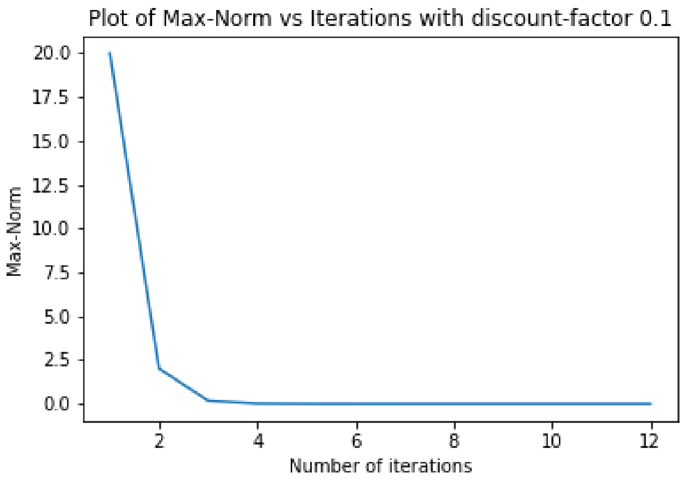
Let state s=(a,b,c)
If a≠b and c=0,then R(s,Putdown,s’)=-10
If a=b and c=0,then R(s,Putdown,s’)=-1
If c=1 and (a,b,2) is not the destination,then R(s,Putdown,s’)=-1
If c=1 and (a,b,2) is the destination state ,then R(s,Putdown,s’)=20
T(s,Putdown,s')=1

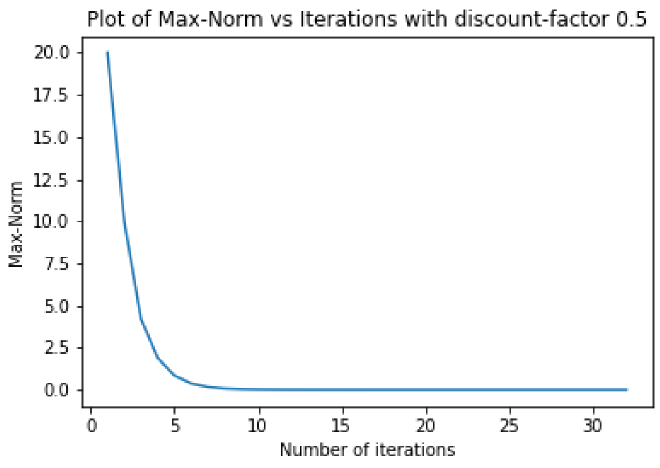
Value Iteration Implementation:

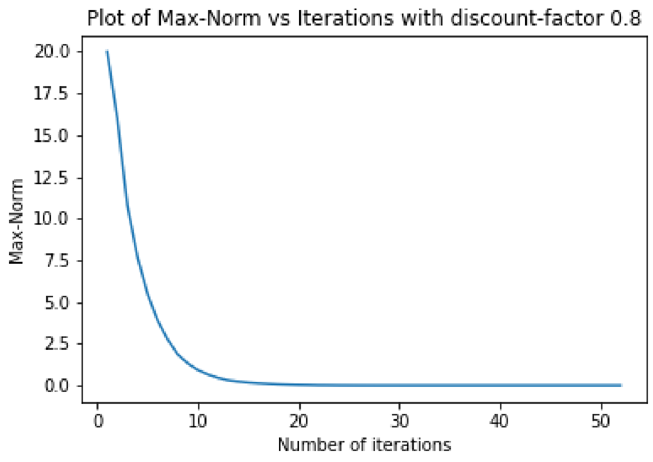
2_a)We chose the value of epsilon as 1e-10 and it took 62 iterations for convergence.

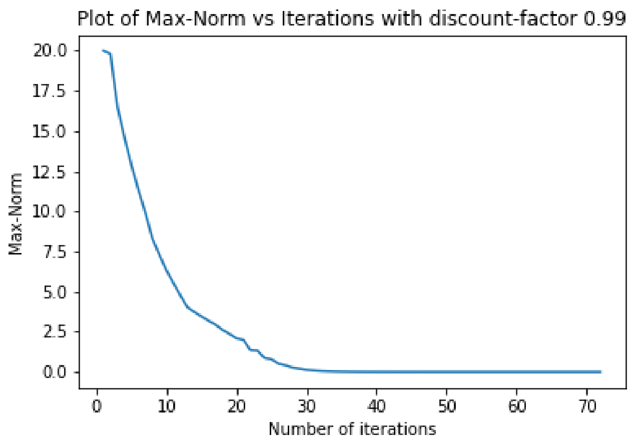
2_b)We see that the number of iterations for convergence of value iteration increases with the increase in the value of discount factor i.e.,for discount factor of 0.01,it takes only 7 iterations to converge whereas it took nearly 70 iterations for convergence when the discount factor is 0.99.







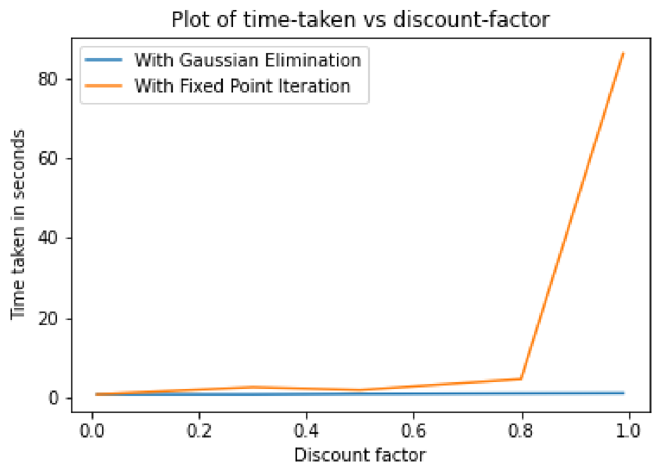




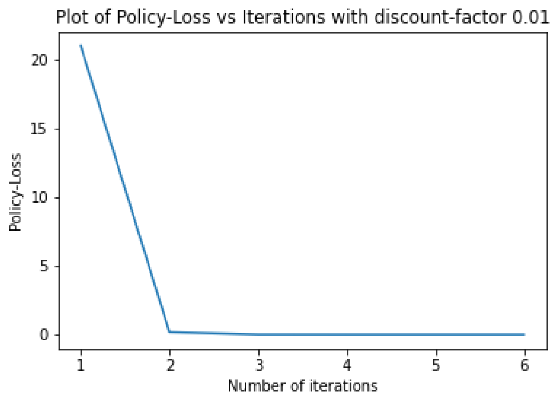
2_c)For smaller values of discount-factor(0.01),the taxi does not perform pickup of passenger in the first 20 steps and just moves in constant direction whereas for larger value of discount factor(0.99) the whole episode of picking up the passenger and dropping him at his destination gets completed within 20 steps.

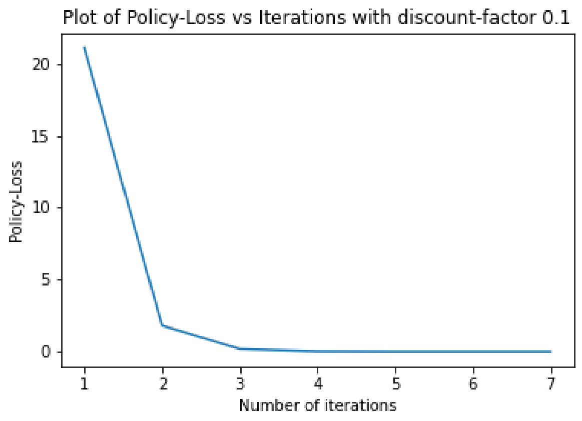
Policy Iteration Implementation:

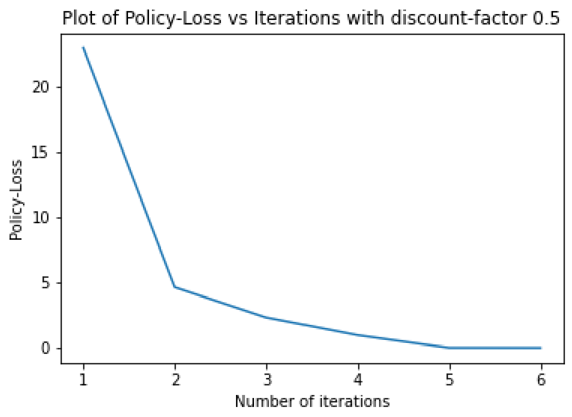
3_a)Linear algebra method will be better for larger discount factors whereas the iterative method would be better for smaller discount factors.

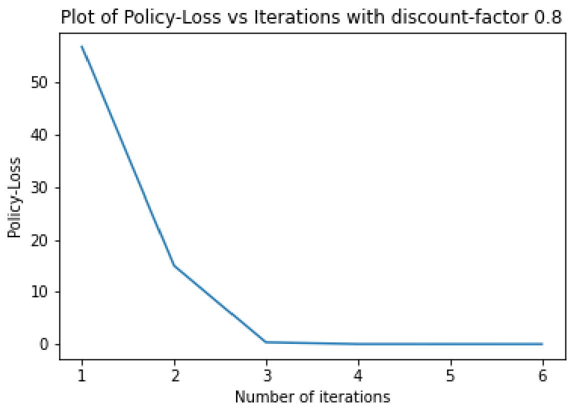


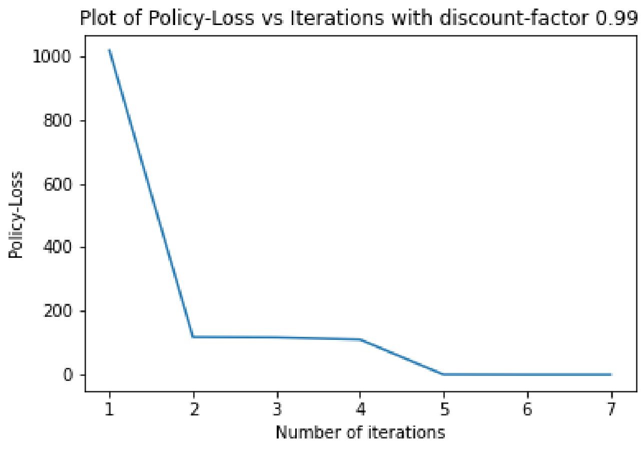
3_b)We see that the policy loss decreases with the number of iterations for all values of discount factors. But the rate of decrease of policy-loss increases sharply with the increase in the discount-factor.




Plot of Policy-Loss vs Iterations with discount-factor 0.1
Policy-Loss
Number of iterations


Plot of Policy-Loss vs Iterations with discount-factor 0.5
Policy-Loss
Number of iterations

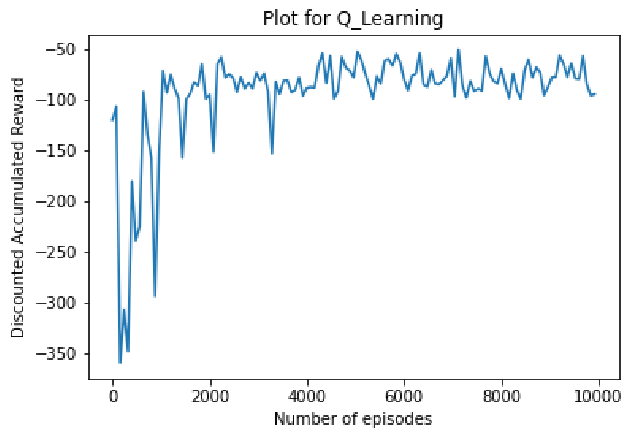

Plot of Policy-Loss vs Iterations with discount-factor 0.8
Policy-Loss
Number of iterations

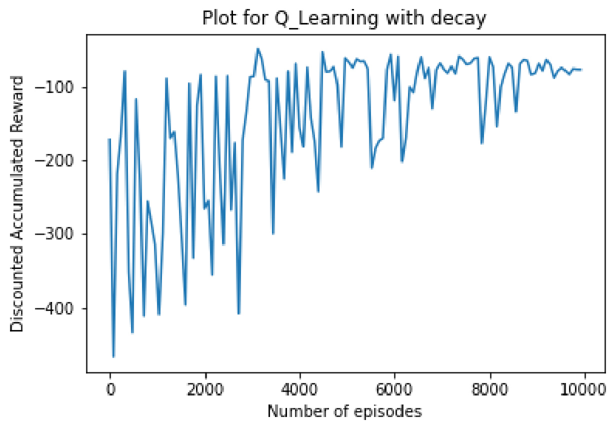

Plot of Policy-Loss vs Iterations with discount-factor 0.99
Policy-Loss
Number of iterations

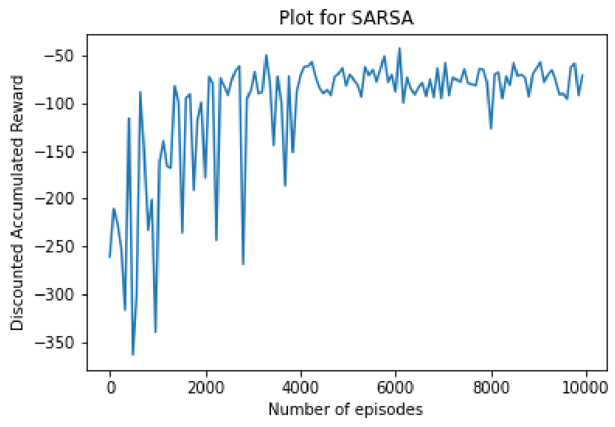
Incorporating Learning:

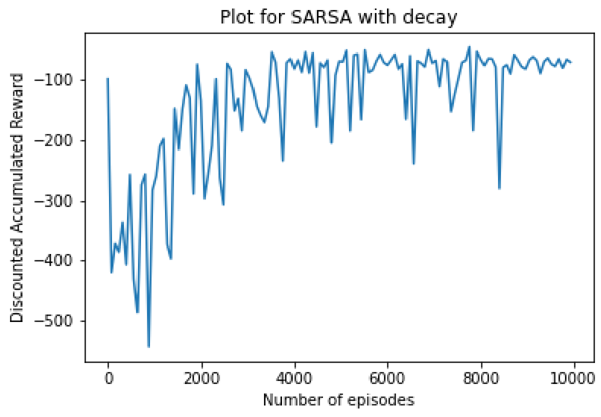
Model free approaches to learn the optimal policy:

2)We executed the algorithms SARSA,Q-Learning,SARSA with decay and Q-Learning with decay for 10000 randomly generated episodes and evaluated the algorithms on another episode to compute discounted accumulated reward for every 80 episodes of the training algorithm.
Like from the plots,we can see that the algorithms with decay,initially tries to visit all possible states and tendency to visit other states decreases with increase with iterations and hence their reward values converge eventually (after 8000-9000 episodes)



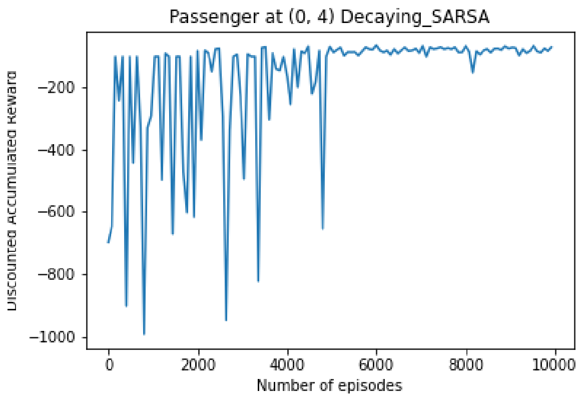


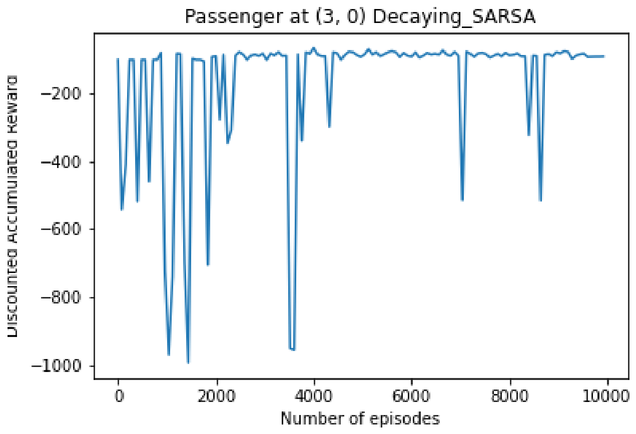


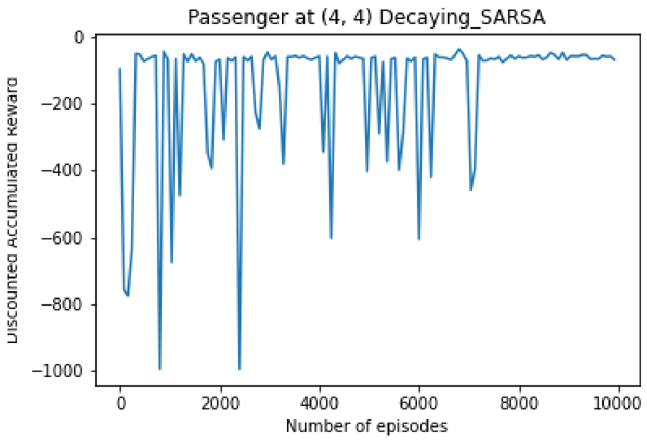


3)The best algorithm obtained from experiments was SARSA with decay. This was because decaying the exploration rate with increase in number of iterations eventually decreases the tendency of choosing random actions by agent.Also SARSA is faster than Q-Learning as we only need to choose a random a’ in case of SARSA in contrast to Q-Learning where we choose A’=argmax Q(s’,a’).

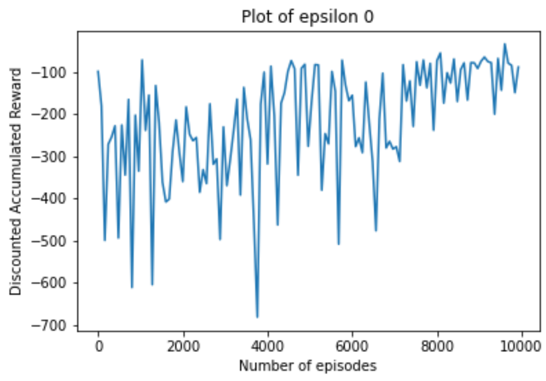
We see that the discounted accumulated rewards for every taxi instance remains the same irrespective of the initial positions of taxi and the passenger.In our case the expected reward value converges to (-1)/(1-0.99)=-100 in all cases.

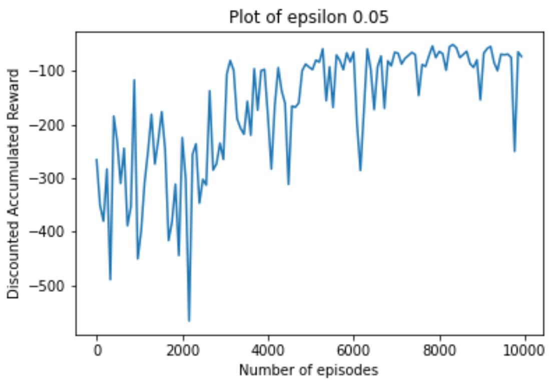




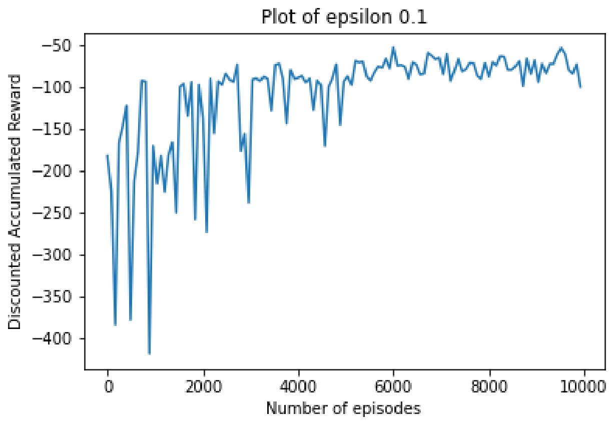


4_i)We see that when the exploration rate is too low(0),the Q-Learning algorithm does not converge as it follows a single fixed policy with low exploration rate and in case of high exploration rate since it covers all possible states,it converges earlier in that case.

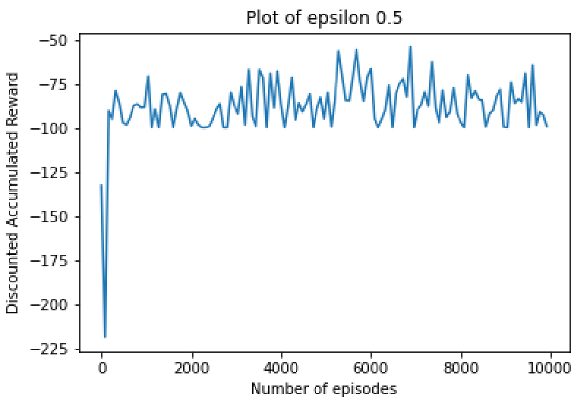




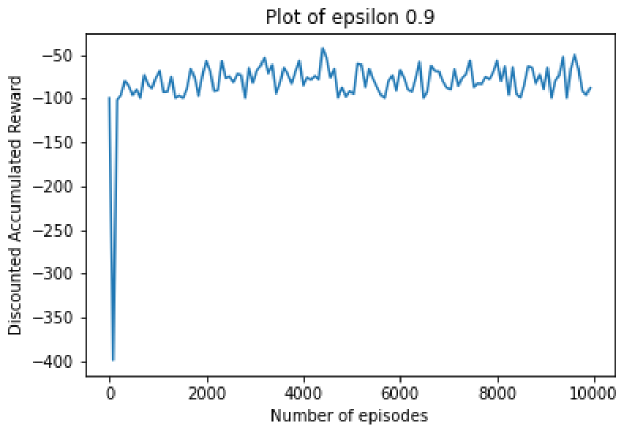
Plot of epsilon 0.05
Discounted Accumulated Reward
Number of episodes



Plot of epsilon 0.1
Discounted Accumulated Reward
Number of episodes

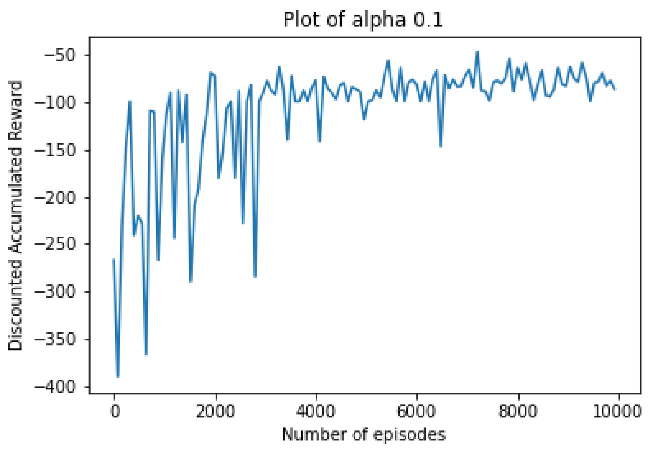


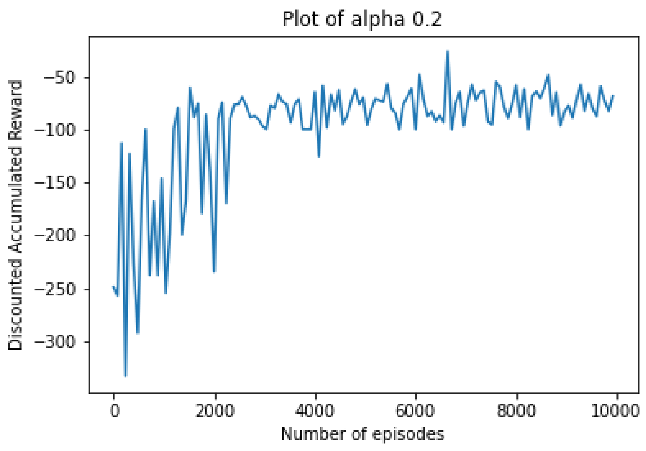
Plot of epsilon 0.5
Discounted Accumulated Reward
Number of episodes

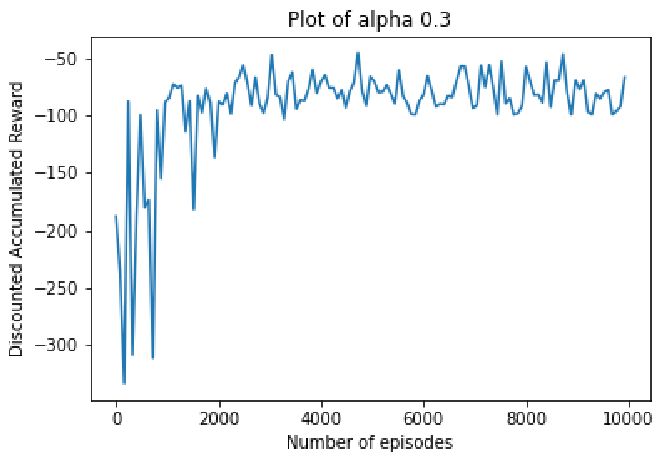


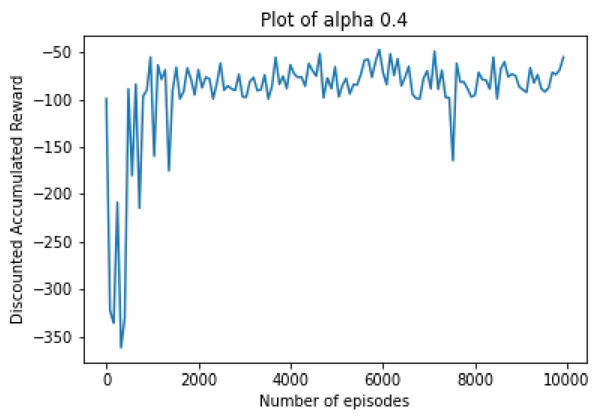
Plot of epsilon 0.9
Discounted Accumulated Reward
Number of episodes

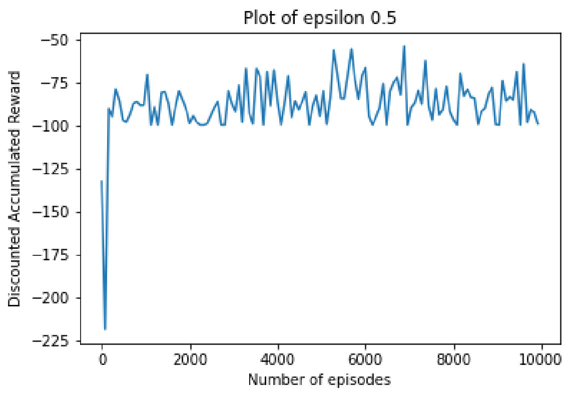
4_ii)We see that with the increase in learning rate(alpha) initially ,the value of accumulated reward convergences faster (from 0.1 to 0.2),it was converging faster but if the increase is very large,then it leads to larger perturbations of Q(s,a) leading to a slower convergence(or even divergence if alpha is close to 1)



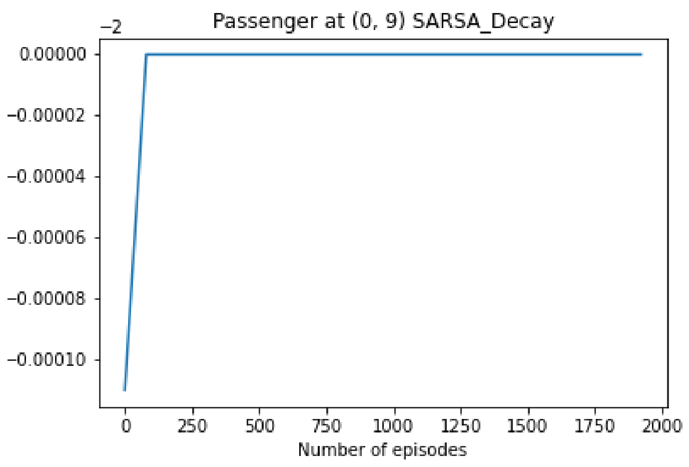


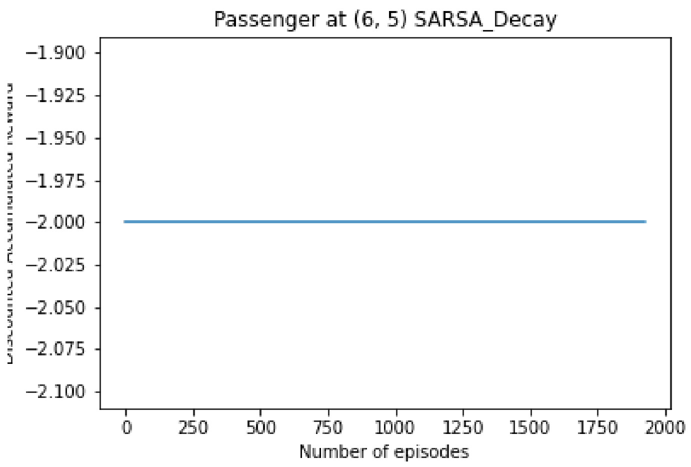


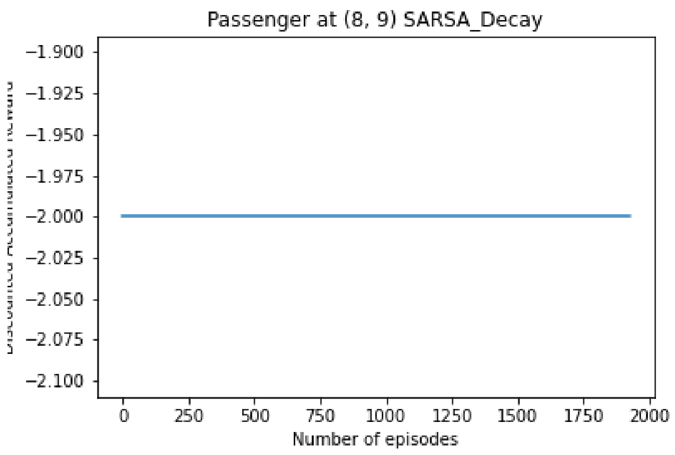


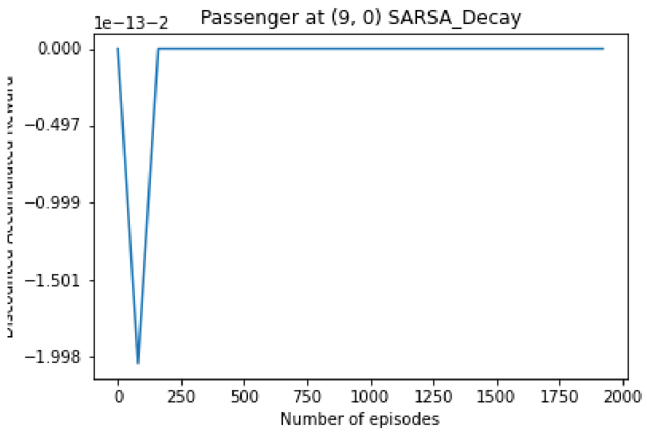


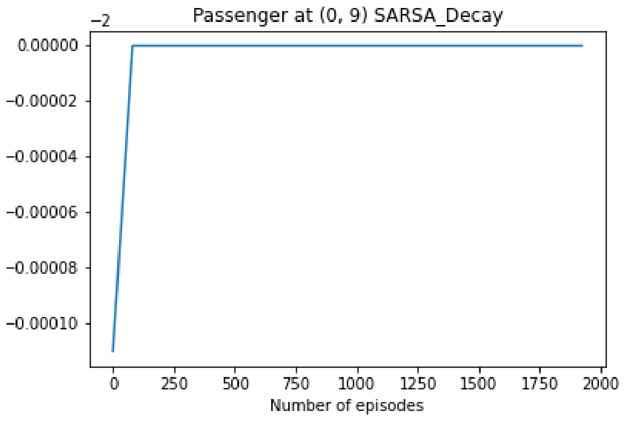
5)The best algorithm in above experiments was SARSA with decay.We implemented each episode to a maximum of 2000 iterations and chose a discount factor of 0.5.All the experiments with different positions of passenger converged to a cumulative discounted reward of (-1)/(1-0.5)=-2.




Passenger at (6, 5) SARSA_Decay
−1.900
−1.925
−1.950
−1.975
−2.000
−2.025
−2.050
−2.075
−2.100
0
250
500
750
1000
1250
1500
1750
2000
Number of episodes


Passenger at (8, 9) SARSA_Decay
−1.900
−1.925
−1.950
−1.975
−2.000
−2.025
−2.050
−2.075
−2.100
0
250
500
750
1000
1250
1500
1750
2000
Number of episodes


Passenger at (9, 0) SARSA_Decay
1e−13−2
0.000
−0.497
−0.999
−1.501
−1.998
0
250
500
750
1000
1250
1500
1750
2000
Number of episodes


Passenger at (0, 9) SARSA_Decay
−2
0.00000
−0.00002
−0.00004
−0.00006
−0.00008
−0.00010
0
250
500
750
1000
1250
1500
1750
2000
Number of episodes