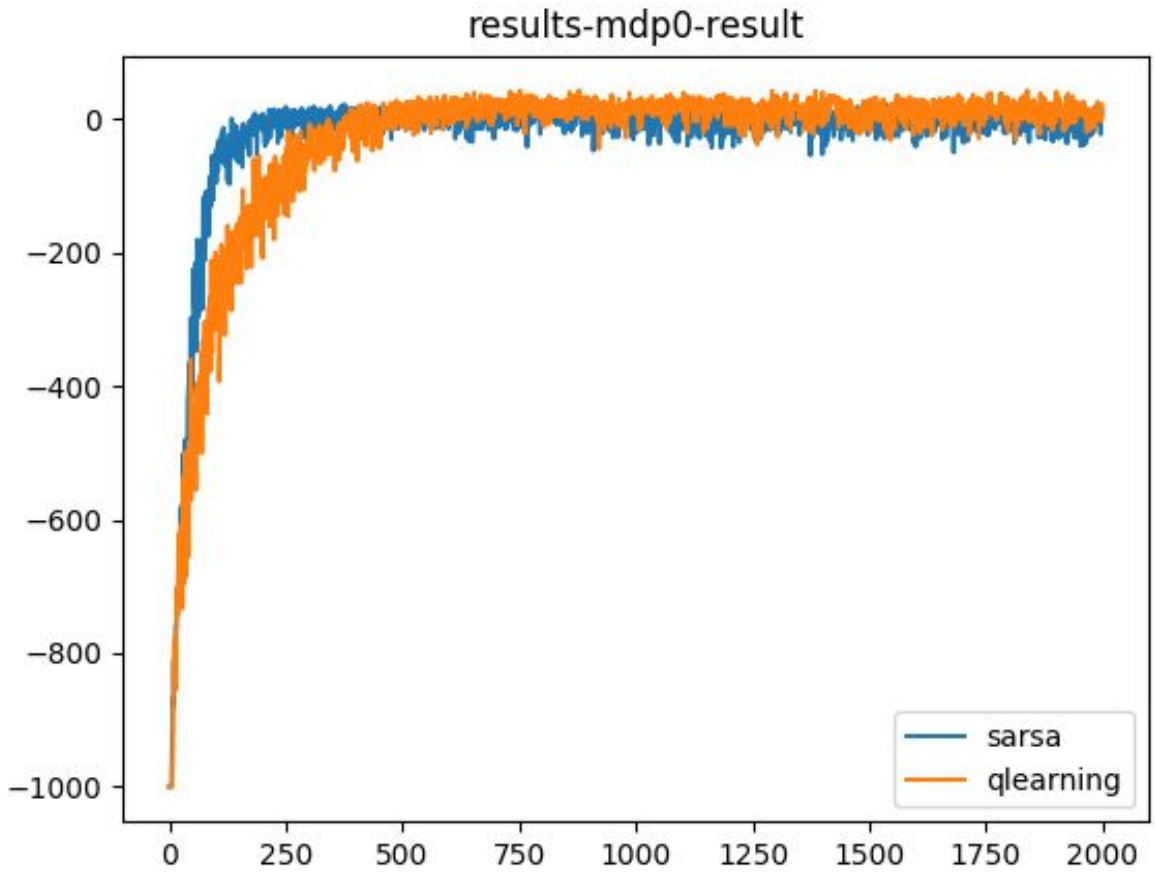
Assignment 3

Akshay Khadse
Roll Number 153079011

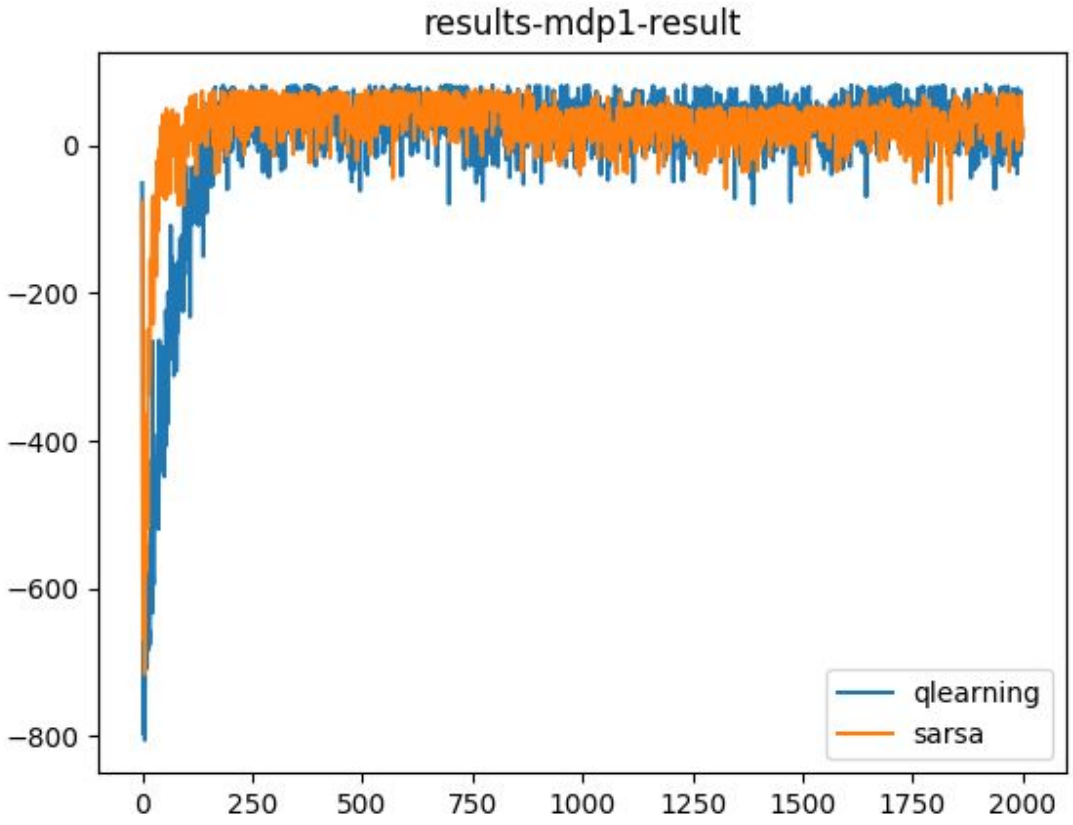
1. **A. Expected cumulative reward against episode number for Q-learning and Sarsa (λ=0.8) for MDP instance 0**

From the experiments carried out (explained further), optimal value of the lambda was found to be 0.8 for MDP instance 0. The optimal alpha was 0.8 and optimal epsilon was 0.2 for both the instances. The figure below shows the expected cumulative reward for gamma 1 vs the number of episodes for SARSA and Q Learning algorithms respectively on MDP instance 0.



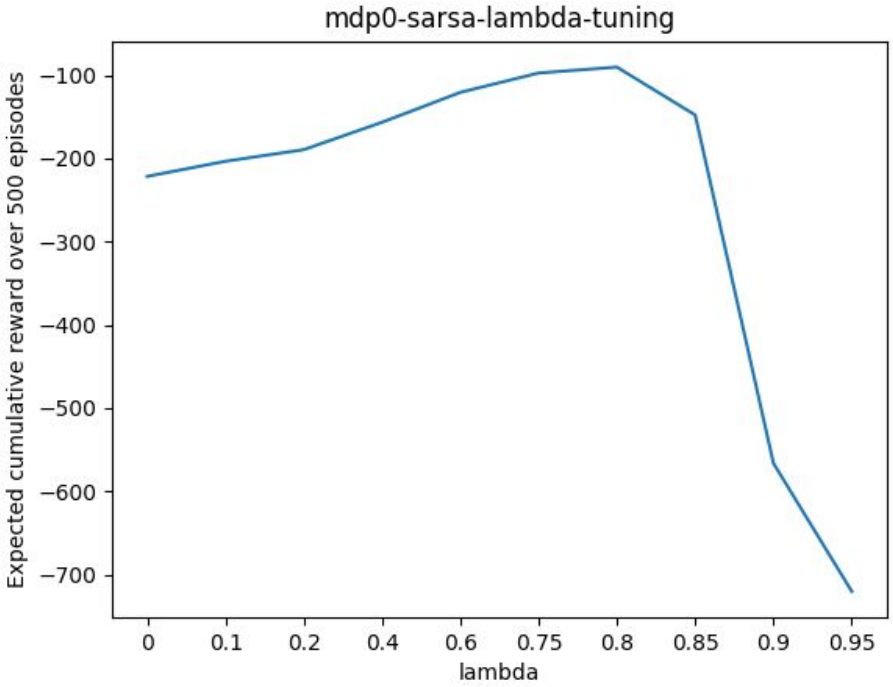
1. **B. Expected cumulative reward against episode number for Q-learning and Sarsa(λ=0.85) for MDP instance 1**

Likewise, optimal value of the lambda was found to be 0.85 for MDP instance 1. The optimal alpha was 0.8 and optimal epsilon was 0.2 for both the algorithms . The figure below shows the expected cumulative reward for gamma 1 vs the number of episodes for SARSA and Q Learning algorithms respectively on MDP instance 1.



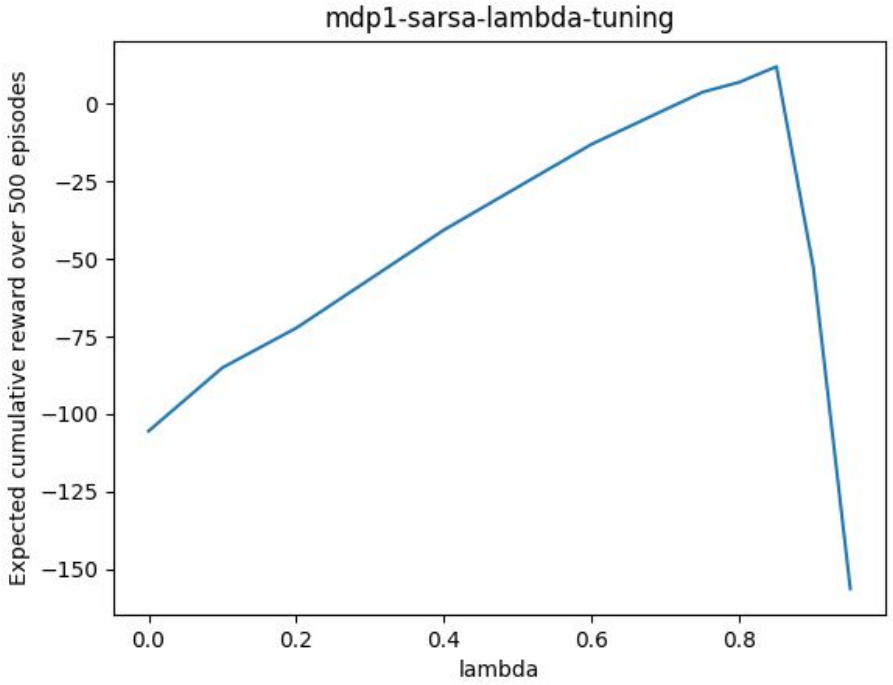
2. A. Expected cumulative reward over the first 500 episodes of training for Sarsa(λ) against λ for MDP instance 0

The procedure followed to tune the lambda value once alpha was fixed at 0.8 and epsilon was fixed at 0.2 was to plot the expected cumulative reward over first 500 episodes for SARSA on both the MDP instances. The figure below shows the variation expected cumulative reward as lambda is changed for MDP instance 0. The values of lambda used are more sparse as the likely distance from observed maxima goes increasing. The optimal value of lambda was found to be around 0.8 for MDP instance 0. This value is used in all the following experiments including the graph in section 1.



2. B. Expected cumulative reward over the first 500 episodes of training for Sarsa(λ) against λ for MDP instance 1

Similar procedure was followed for MDP 2 to find the optimal value of lambda as shown in figure below. The optimal value of lambda was found to be around 0.85 for MDP instance 1.



3. A. Tuning of epsilon for Q Learning on MDP instance 0

For tuning the epsilon, the alpha was set to 0.1. Then, the epsilon was varied from 0.1 to 0.9 in steps of 0.1. For each value of epsilon the expected cumulative reward was plotted against the episode number on the same axes for comparison.

But, due to noise, optimal epsilon is not clearly distinguishable. So, the the plots for suboptimal epsilons were removed. The plots for epsilons 0.1, 0.2 and 0.3 are nearly identical. Hence, individual plots for each epsilon (all the plots not included in the report are available on the link at the end of this report) were then checked to arrive at a moderate value of epsilon as 0.2.

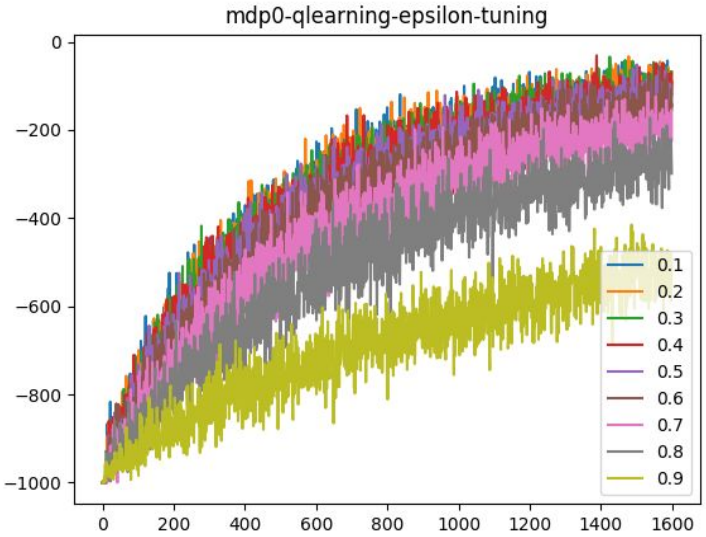


Figure 1

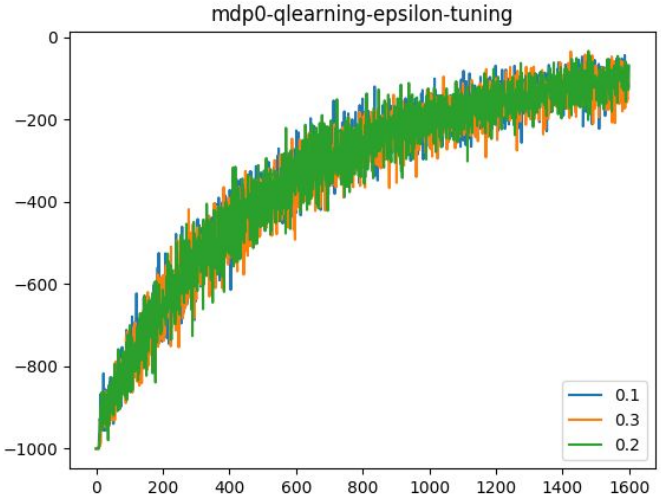


Figure 2

3. B. Tuning of epsilon for Q Learning on MDP instance 1

Same procedure as above was followed for rest of the experiments for tuning the epsilon across MDP instances 0 and 1. A suitable epsilon was found to be 0.2.

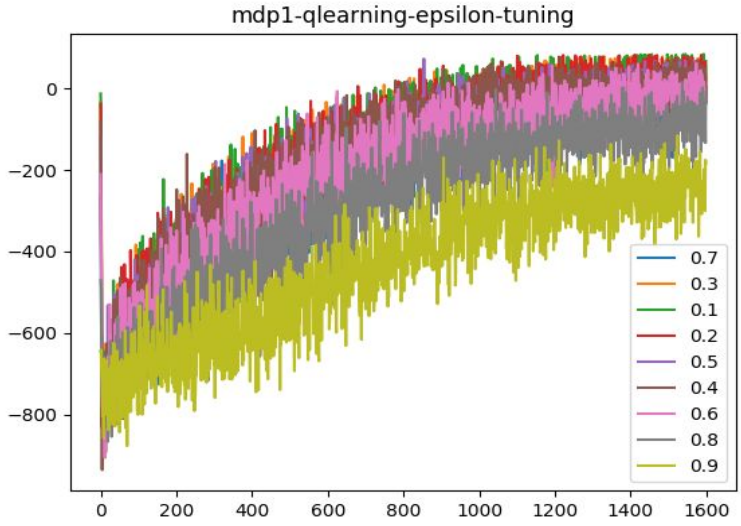


Figure 3

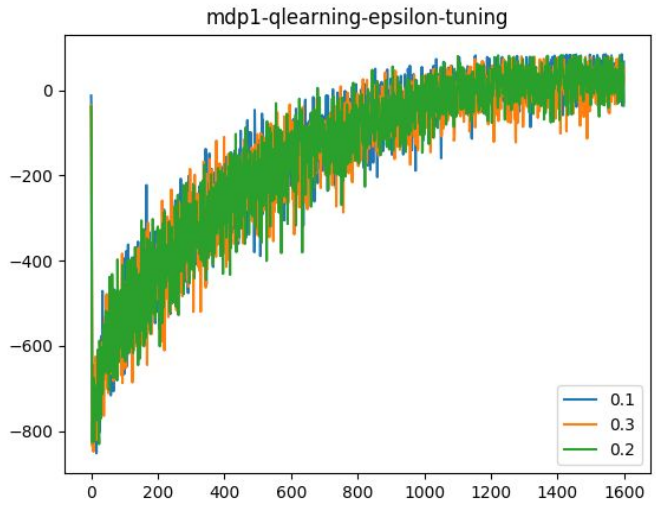


Figure 4

3. C. Tuning of epsilon for SARSA on MDP instance 0

By applying the same hypothesis, while keeping alpha to be 0.1 and lambda 0, 0.2 was found out to be the suitable epsilon. The trace was kept as replace. This is compared with accumulating trace at the end.

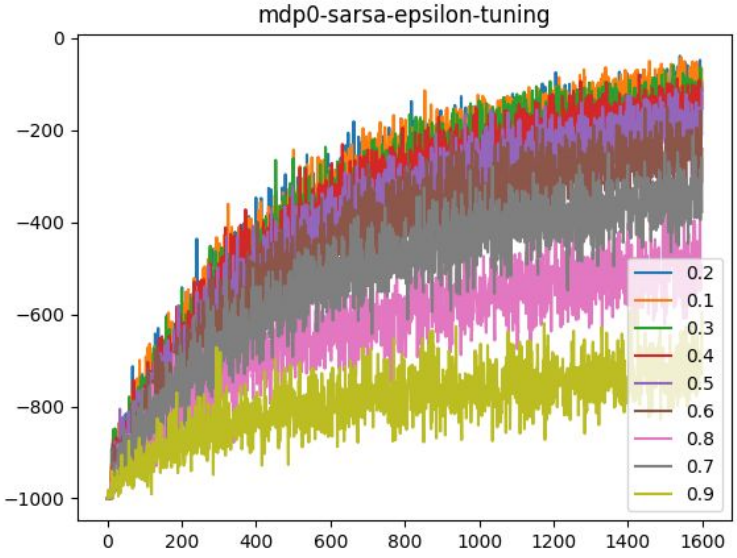


Figure 5

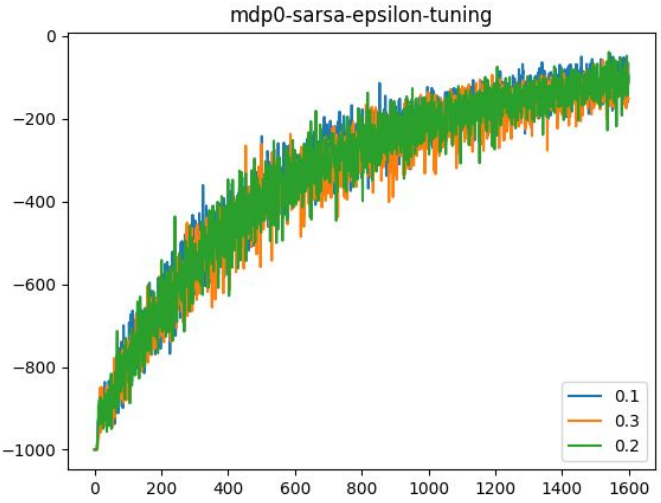


Figure 6

3. D. Tuning of epsilon for SARSA on MDP instance 1

Alpha was kept at 0.2, lambda was kept at 0 and trace set to replace, to get epsilon as 0.2 using the above described procedure.

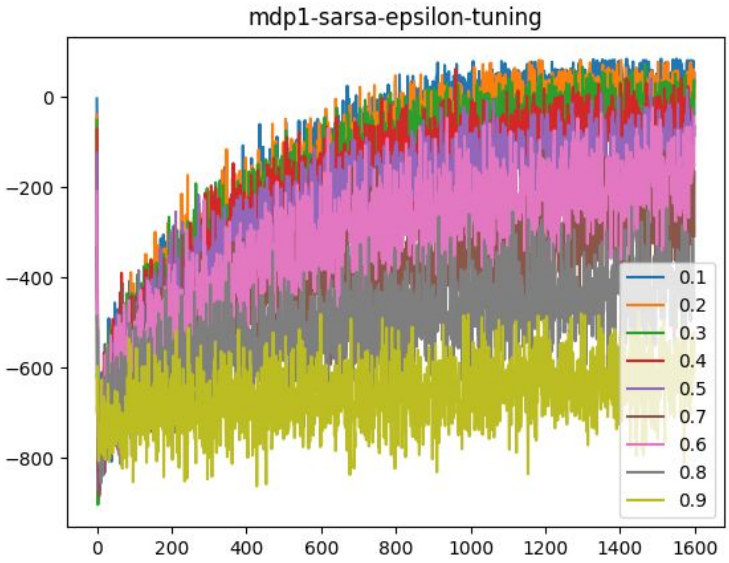


Figure 7

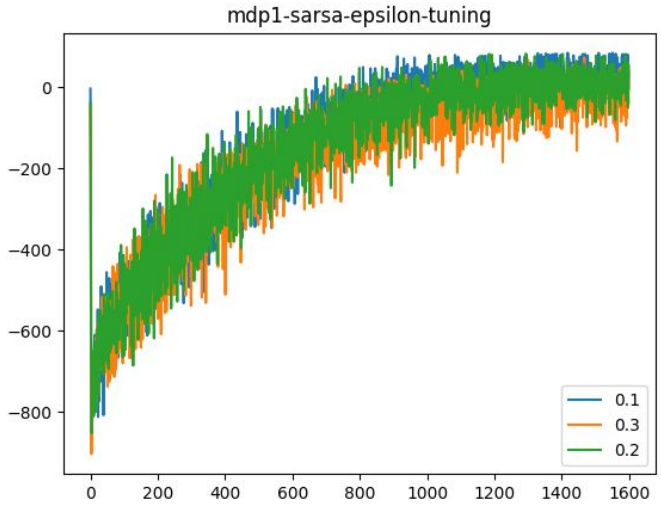
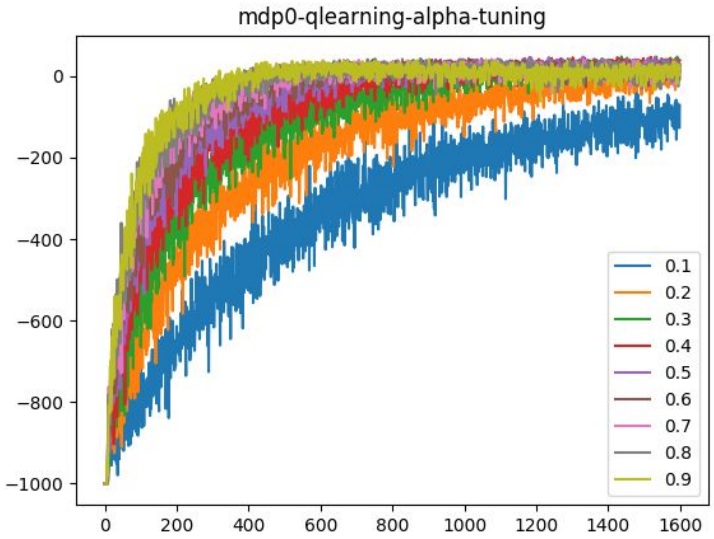


Figure 8

Since, the optimal value of epsilon is not clearly distinguishable and lies around 0.2 in each case, this was set to be the default value of epsilon in client.py. All the tuned options hereafter are set as default in client.py.

4. A. Tuning of alpha for Q Learning on MDP instance 0



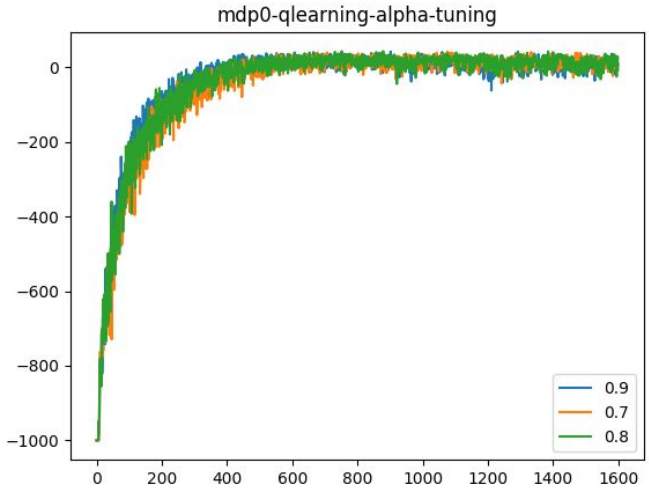


Figure 9 **Figure 10**

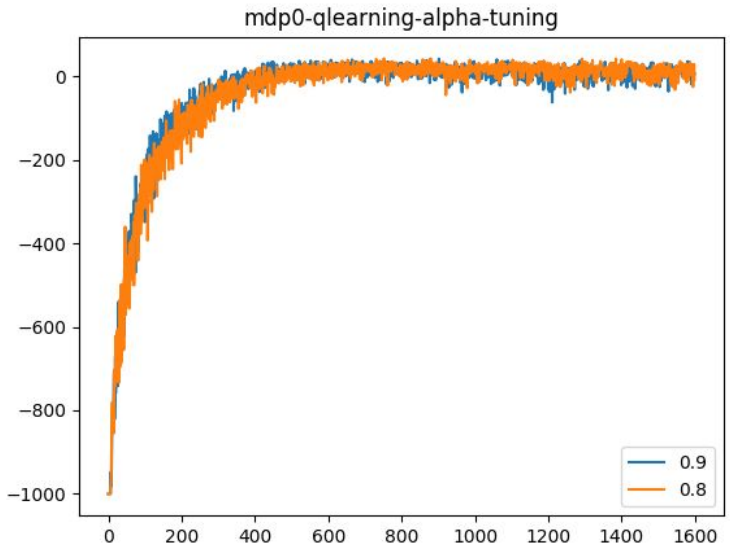


Figure 11

For tuning the alpha, epsilon was set to the value found in previous section i.e. 0.2. The alpha was varied from 0.1 to 0.9 in steps of 0.1 and at each step the expected cumulative reward was plotted against the episode number on the same axes for the sake of comparison. To get a better idea of the trend of the alpha, plots for suboptimal values of alpha were removed and the remaining plots nearly coincided. The individual plots for each alpha (all the plots not included in the report are available on the link at the end of this report) were then checked to arrive at a suitable value of alpha as 0.8.

4. B. Tuning of alpha for Q Learning on MDP instance 1

The same procedure was followed for MDP instance 1 with epsilon 0.2 and alpha was chosen to be 0.8.

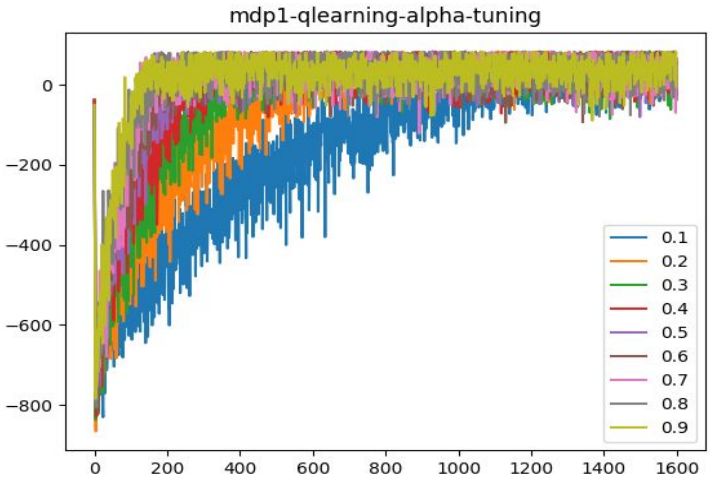


Figure 12

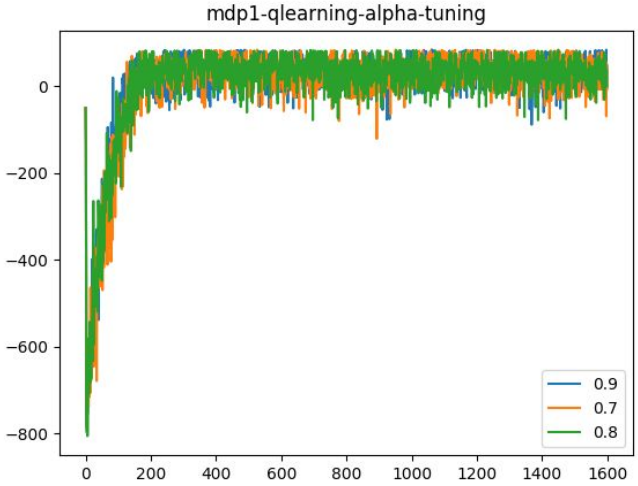


Figure 13

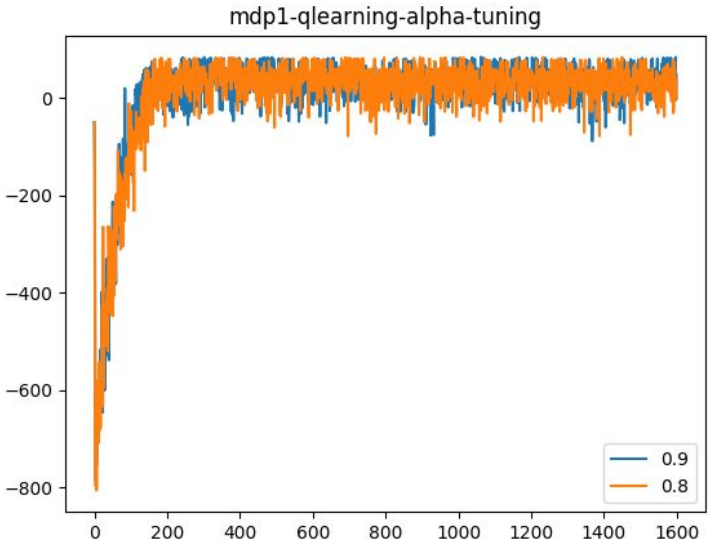


Figure 14

4. C. Tuning of alpha for SARSA on MDP instance 0

Similar procedure was applied for SARSA instances. Epsilon was kept to be 0.2, lambda to be 0 and the trace was defaulted to replace. The suitable value of alpha turned out to be 0.8

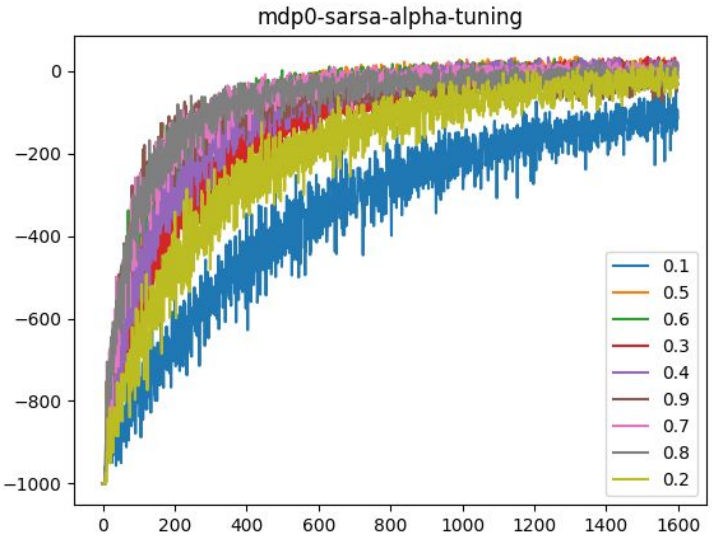


Figure 15

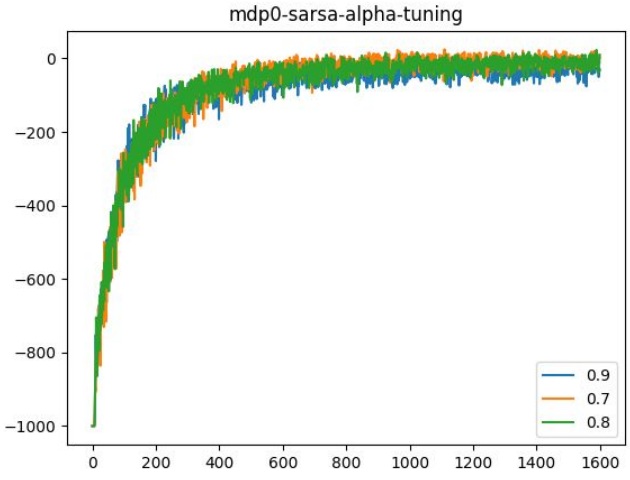


Figure 16

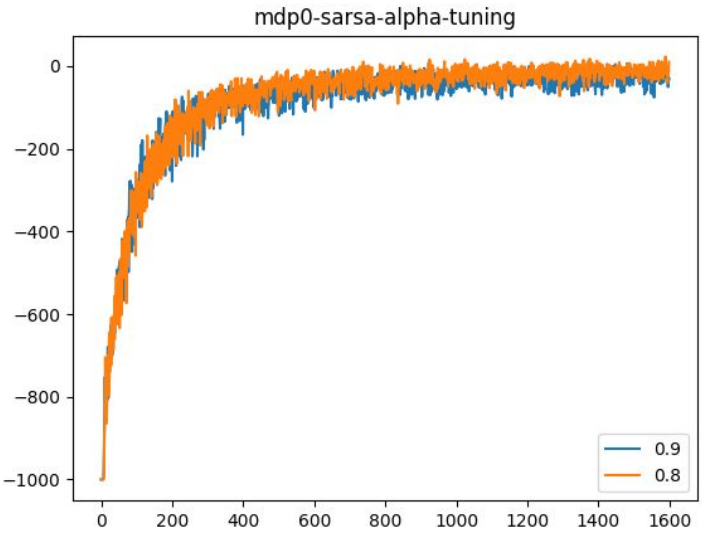


Figure 17

4. D. Tuning of alpha for SARSA on MDP instance 1

The same procedure with same parameters as above was applied here also to get alpha as 0.8.

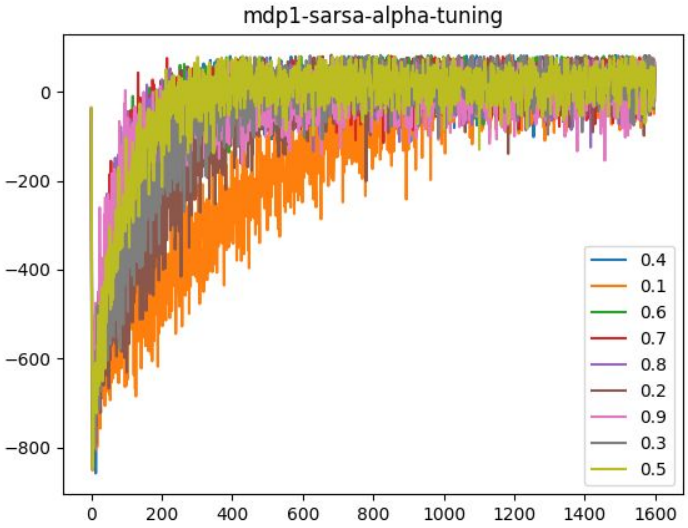


Figure 18

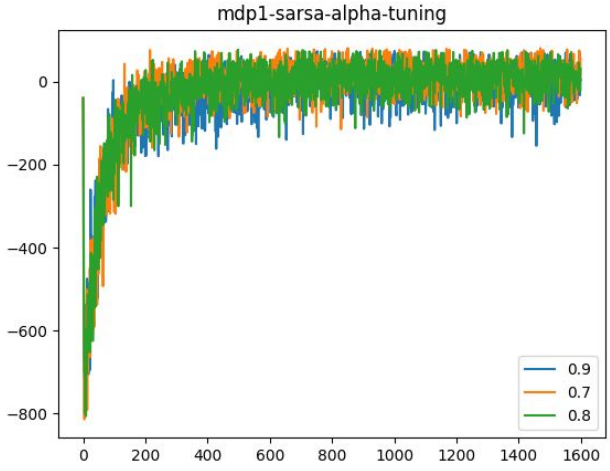


Figure 19

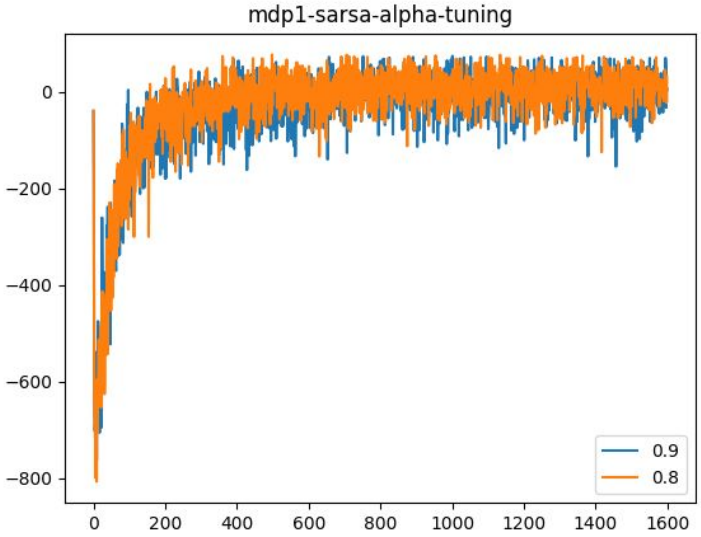


Figure 20

5. A. Lambda tuning of SARSA on MDP instance 0

The lambda was tuned based on the requirement of the assignment to plot expected cumulative reward for first 500 episodes against lambda. To get more insight however, the trend of expected cumulative reward for the same 500 episodes was plotted against the episode number for each lambda and is shown in figure 22. The maximum expected cumulative reward for MDP instance 0 was found to be at lambda 0.8. This was selected as the optimal value of lambda.

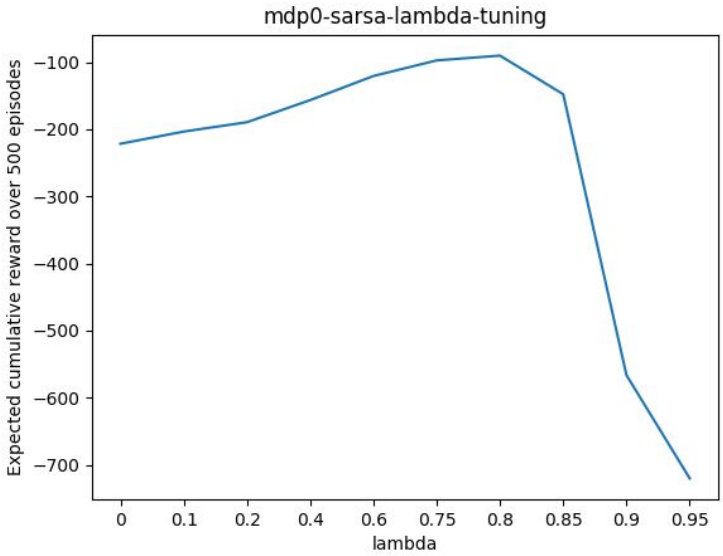


Figure 21

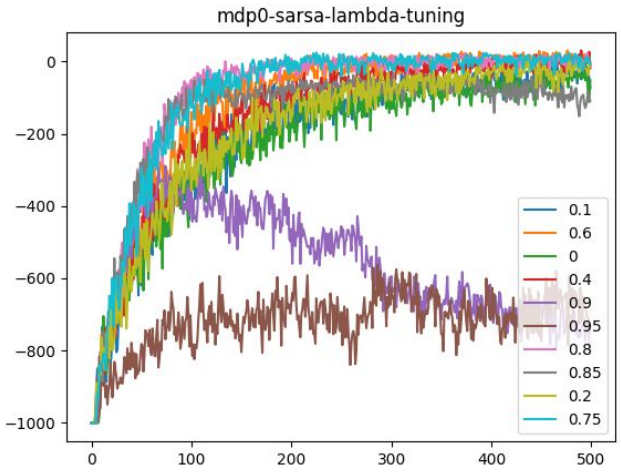


Figure 22

5. B. Lambda tuning of SARSA on MDP instance 1

As in the section 5. A., here optimal value of lambda was found to be 0.85 from figure 23.

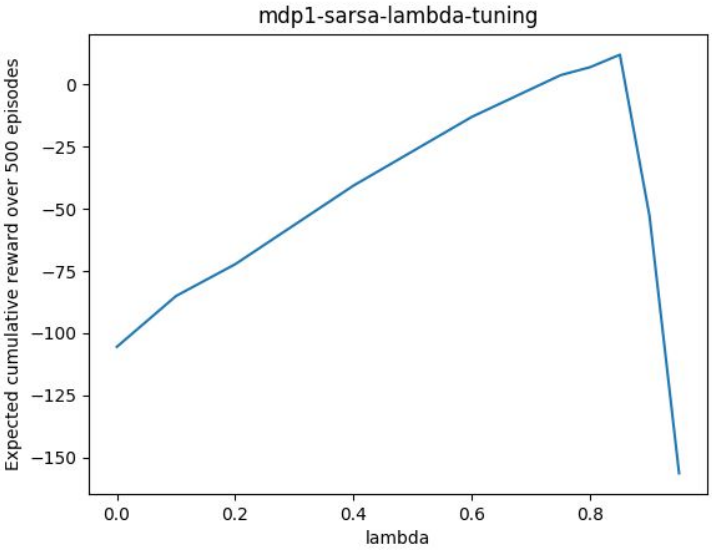


Figure 23

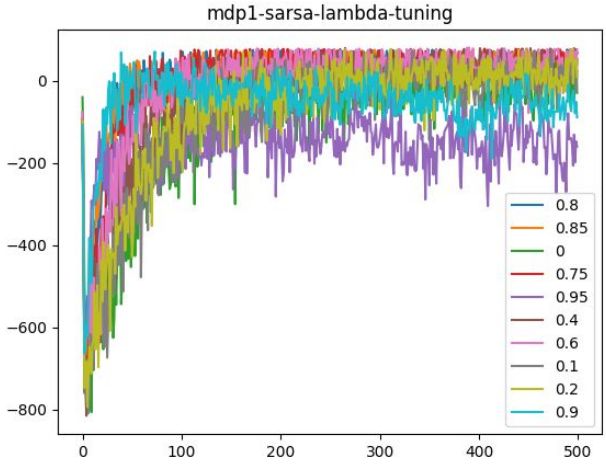


Figure 24

6. Comparison between variants of Q Learning on basis of annealing

An experiment was carried out to spot the effect of annealing of epsilon on the Q Learning algorithm. Figures 25 and 26 show the expected cumulative reward vs the episode number under annealing and in its absence for MDP instances 0 and 1 respectively. In MDP instance 0,

Annealing improves the performance slightly but no such conclusion can be drawn from the MDP instance 1. Hence, annealing was not considered.

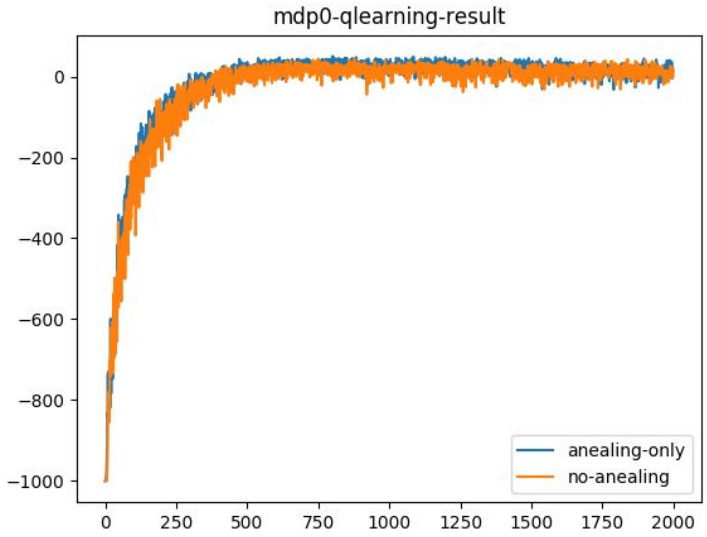


Figure 25

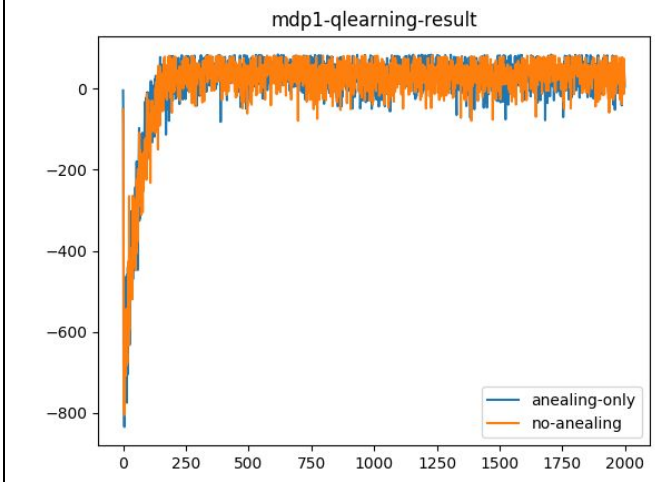
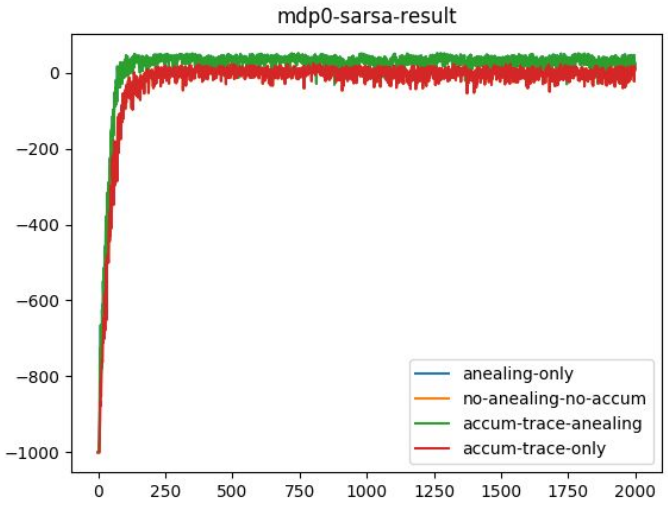


Figure 26

7. Comparison between variants of Comparison between variants of SARSA on basis of annealing and accumulating trace

Unlike above case, SARSA shows significant improvement when both annealing and accumulating trace are used, but the improvement in the MDP instance 1 is not significant. Also, the no annealing no accumulating trace is in between the both the extremes. Since, this agent is to be tested on random instances of this MDP, the annealing and accumulating traces were not used.



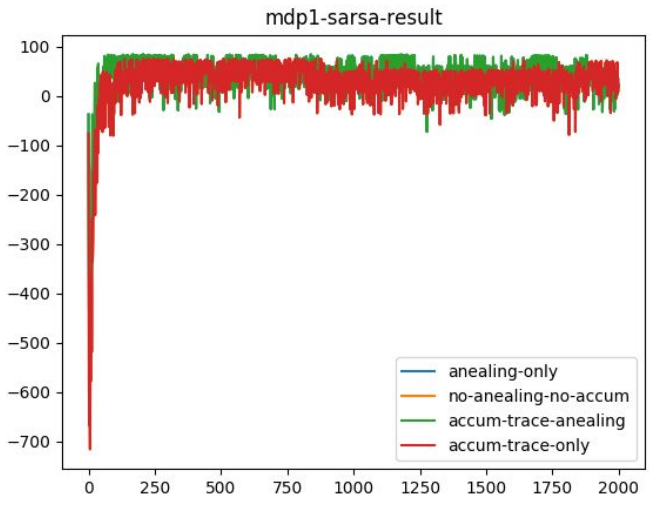


Figure 27 **Figure 28**

8. Observations

From section 1 it was observed that with proper tuning, SARSA(lambda) performs better than Q Learning for the same MDP. SARSA converges quickly than Q learning but leaves some margin for improvement w.r.t Q Learning in MDP instance 0.
From section 2, it was observed that as lambda is increased, the expected cumulative reward increases upto a certain lambda, then it decreases even more as lambda approaches 1. This optimal lambda changes if the learning rate alpha changes.
The noise in plots after 50 trials is more pronounced in MDP instance 1 which makes the tuning difficult. The number of trials for MDP 1, therefore should be more than 50.

Changes in client.py:
Command line options to change alpha i.e -al or --alpha, and epsilon i.e -ep or --epsilon were added.
Link to complete result:
https://drive.google.com/drive/folders/0B_M9Q9M4VTFtTU5ScjhRbTEwbWc?usp=sharing