CS6700 - Reinforcement Learning
Programming Assignment 3

April 6, 2022

1 Hierarchical Reinforcement Learning

For this assignment, we will be referring to Sutton, Precup and Singh's 1999 paper, 'Between MDPs and semi-MDPs : A Framework for Temporal Abstraction in Reinforcement Learning'. Please read the paper upto and including Section 3, it is self explanatory and a good reference leading up to the understanding and implementation of SMDP Q-learning. Section 3 of the paper talks about SMDP planning and is necessary to build intuition to solve this assignment. We will be working with a simple taxi domain environment (explained in the next section). Your tasks are to implement 1-step **SMDP Q-Learning** and **intra-option Q-Learning** on this environment.

2 Environment Description

The environment for this task is the taxi domain, illustrated in Fig. 1. It is a 5x5 matrix, where each cell is a position your taxi can stay at. There is a single passenger who can be either picked up or dropped off, or is being transported. There are four designated locations in the grid world indicated by R(ed), G(reen), Y(ellow), and B(lue). When the episode starts, the taxi starts off at a random square and the passenger is at a random location. The taxi drives to the passenger's location, picks up the passenger, drives to the passenger's destination (another one of the four specified locations), and then drops off the passenger. Once the passenger is dropped off, the episode ends.

There are 500 discrete states since there are 25 taxi positions, 5 possible locations of the passenger (including the case when the passenger is in the taxi), and 4 destination locations. Note that there are 400 states that can actually be reached during an episode. The missing states correspond to situations in which the passenger is at the same location as their destination, as this typically signals the end of an episode. Four additional states can be observed right after a successful episodes, when both the passenger and the taxi are at the destination. This gives a total of 404 reachable discrete states.

Passenger locations: 0: R(ed); 1: G(reen); 2: Y(ellow); 3: B(lue); 4: in taxi
Destinations: 0: R(ed); 1: G(reen); 2: Y(ellow); 3: B(lue)
Rewards:

- -1 per step unless other reward is triggered.
- +20 delivering passenger.
- -10 executing "pickup" and "drop-off" actions illegally.

The discount factor is taken to be $\gamma = 0.9$.

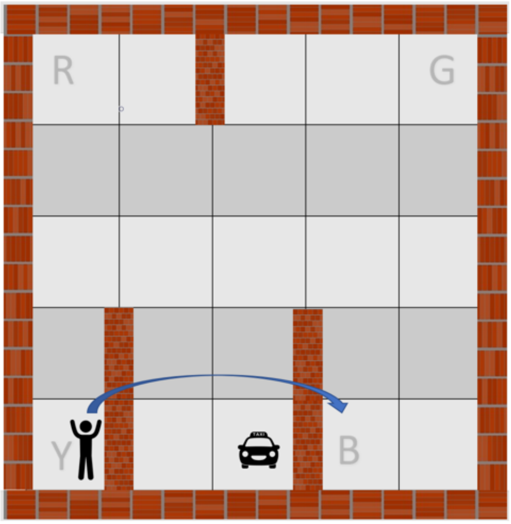


Figure 1: Taxi Domain

3 Actions and Options

Actions: There are 6 discrete deterministic actions: 0: move south; 1: move north; 2: move east; 3: move west; 4: pick passenger up; and 5: drop passenger off.
Options: Options to move the taxi to each of the four designated locations, executable when the taxi is not already there.

You will be experimenting with OpenAI Gym's Taxi-v3 environment.

4 Tasks

First, implement the single step **SMDP Q-learning** for solving the taxi problem. A rough sketch of the algorithm is as follows: Given the set of options,

- Execute the current selected option to termination (e.g. use epsilon greedy $Q(s, o)$).
- Computer $r(s, o)$.
- Update $Q(s_t, o)$.

Second, implement **intra-option Q-Learning** on the same environment.

For each algorithm, do the following (only for the configuration with the best hyperparameters):

1. Plot reward curves and visualize the learned Q-values.
2. Provide a written description of the policies learnt and your reasoning behind why the respective algorithm learns the policy.

3. Is there an alternate set of options that you can use to solve this problem, such that this set and the given options to move the taxi are mutually exclusive? If so, run both algorithms with this alternate set of options and compare performance with the algorithms run on the options to move the taxi.

Finally, provide a comparison between the SMDP Q-Learning and intra-option Q-Learning algorithms. Do you observe any improvement with intra-option Q-Learning? If so, describe why this happens as well. Please make sure that all descriptions are **brief** and to the point.

5 Submission Instructions

You are required to submit both your report and your code. Please submit in **teams of two**. One submission per team will suffice. The due date for this programming assignment is **11:59 pm on April 21st**.

Assignment 3

SMDP Learning

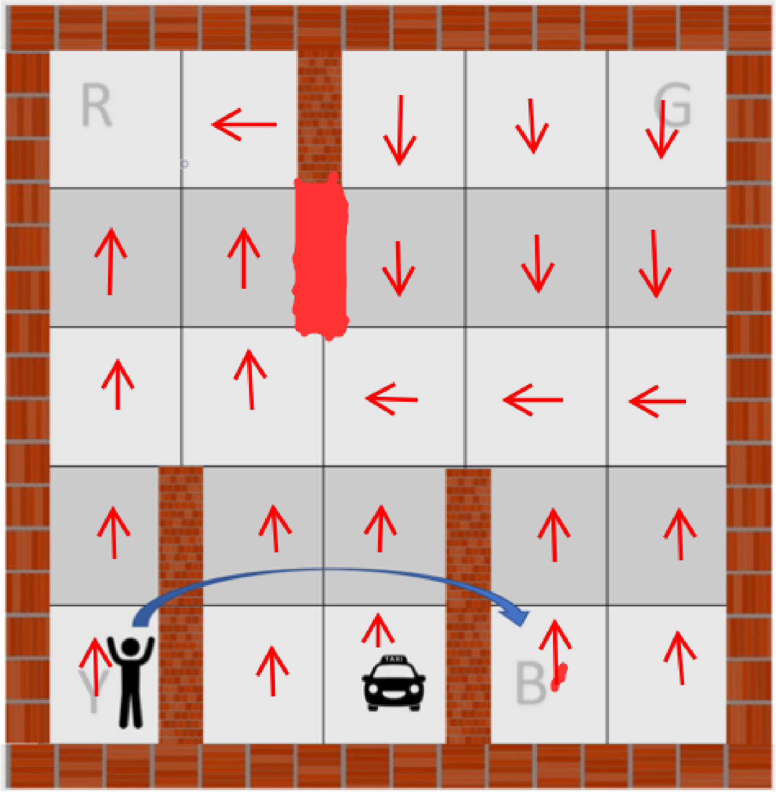
Options

We added four other options apart from the primitive six options (south, north, east, west, pick, drop).

Go to Red

1. *Initialization set* $\mathcal{I}_o \subseteq$\{all 500 states except R\}
2. *Policy*

Figure 1: Policy followed for going to Red



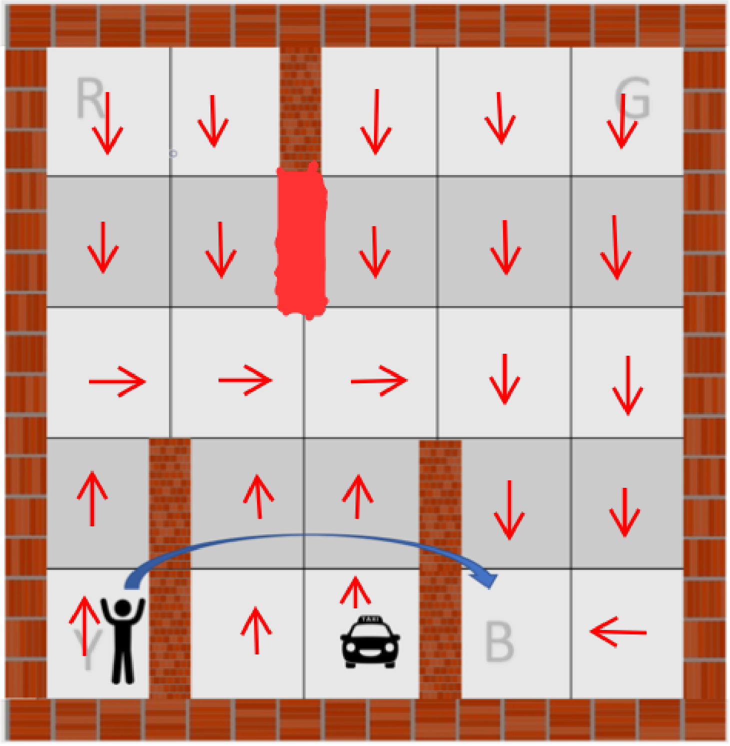
3. *Termination*

   Terminate when the taxi has reached the Red state

Go to Blue

1. *Initialization set* $\mathcal{I}_o \subseteq$\{all 500 states except B\}
2. *Policy*

Figure 2: Policy followed for going to Blue



3. *Termination*

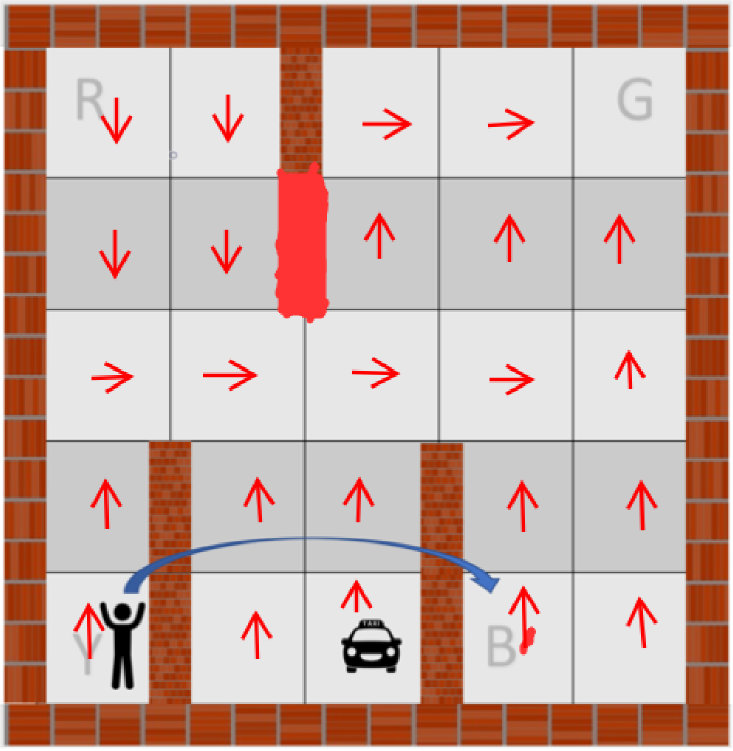
   Terminate when the taxi has reached the Blue state

Go to Green

1. *Initialization set* $\mathcal{I}_o \subseteq$\{all 500 states except G\}

2. *Policy*

Figure 3: Policy followed for going to Green



3. *Termination*

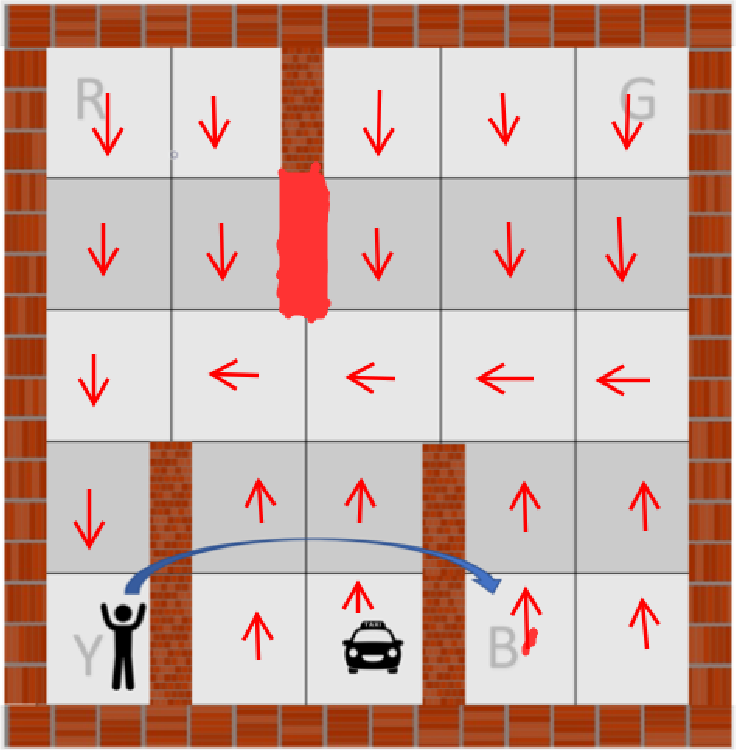
   Terminate when the taxi has reached the Green state

Go to Yellow

1. *Initialization set* $\mathcal{I}_o \subseteq$\{all 500 states except Y\}

2. *Policy*

Figure 4: Policy followed for going to Yellow



3. *Termination*

   Terminate when the taxi has reached the Yellow state

Hyper Parameter Tuning

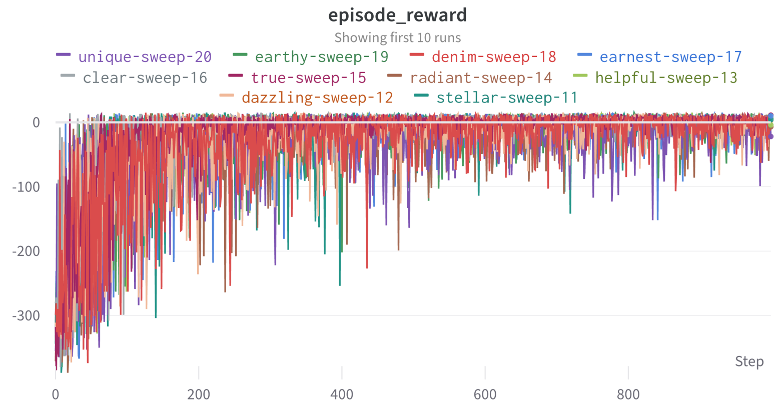
Tuning

Parameter	**max**	**min**	**Tuned value**
alpha (α)	0.1	0.4	0.28216
epsilon (ϵ)	0.1	0.3	0.13732

Tuning Graphs

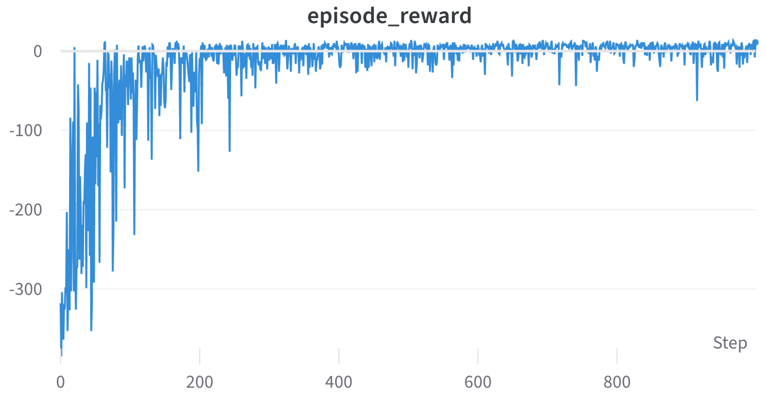
Complete Hyper parameter tuning information could be found here

Figure 5: Reward graphs of Hyper Parameter Tuning for alpha, epsilon



Reward Curve for the best Hyper parameters

Figure 6: Reward graphs for the best Hyper parameters of alpha, epsilon



Learned Policy

We have run three episodes of the learned policy, click here to play the video

Ep 1

The passenger was at red, the destination was yellow. The taxi was spanned near Red, it went to red to pickup and directly went to yellow to drop off.

Ep 2

The passenger was at yellow. The taxi was spanned in the middle. The destination is green. It followed the optimal actions to reach the destination and dropped off the passenger there.

Ep 3

The source is yellow and destination is green, The taxi spanned at Red but it directly didn't go to green rather it took a little longer path.

Overall Policy Learnt

The taxi has learned to go the source by selecting the appropriate option at the start. Then once after the pickup it has learned to select the option to go the destination. This is what is expected intuitively.

Q-value visualisation

We've plotted the heatmap of q-values of optimal actions/options in some of the sates.

Symbols
R : Go to Red option
B : Go the Blue option
Y : Go to Yellow option
G : Go to Green option

Figure 7: Optimal Q values are visualized for the states where the pickup location is Red(Top Left)

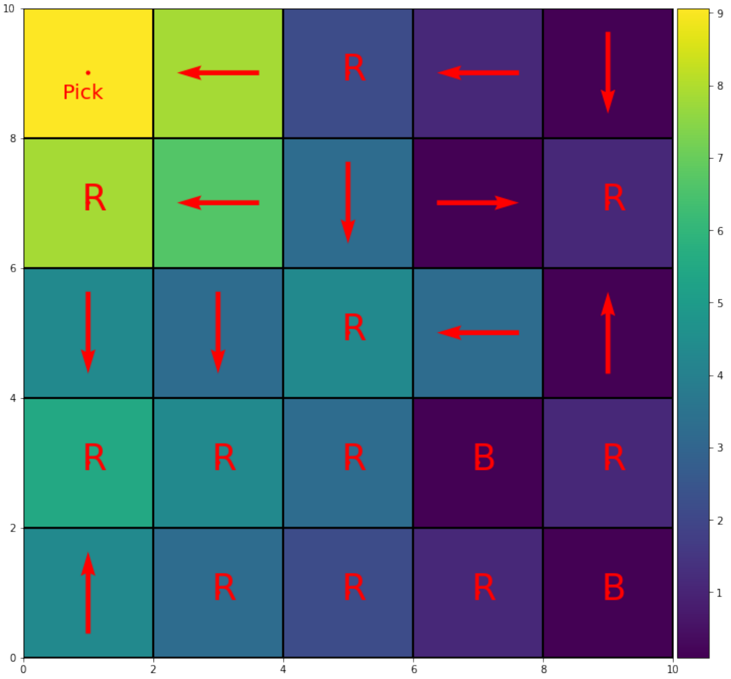
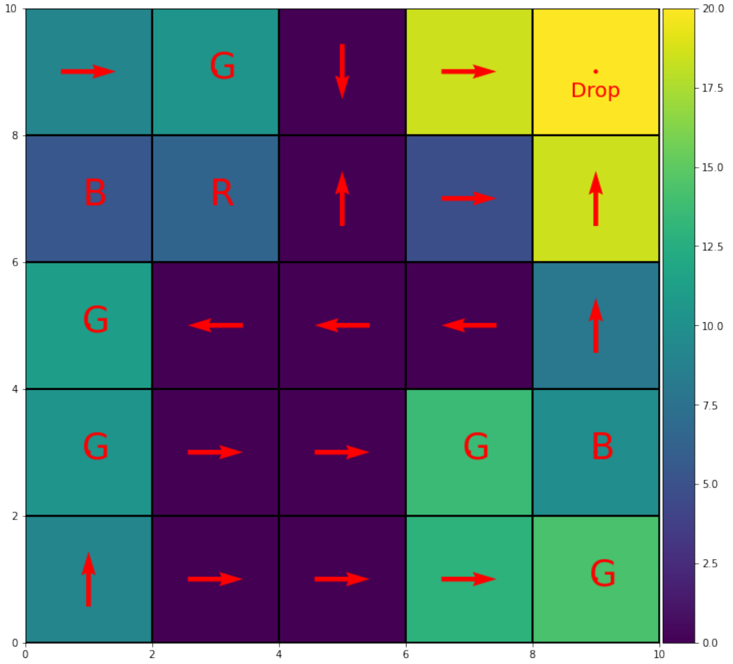


Figure 8: Optimal Q values are visualized for the states where the drop location is Green(Top Right)



Intra Q Learning

Options

We used the same options as above

1. Go to Red
2. Go to Blue
3. Go to Green
4. Go to Yellow

Hyper Parameter Tuning

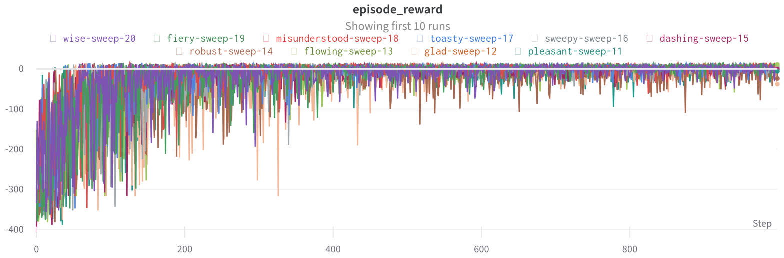
Tuning

Parameter	**max**	**min**	**Tuned value**
alpha (α)	0.1	0.4	0.1975
epsilon (ϵ)	0.1	0.3	0.122

Tuning Graphs

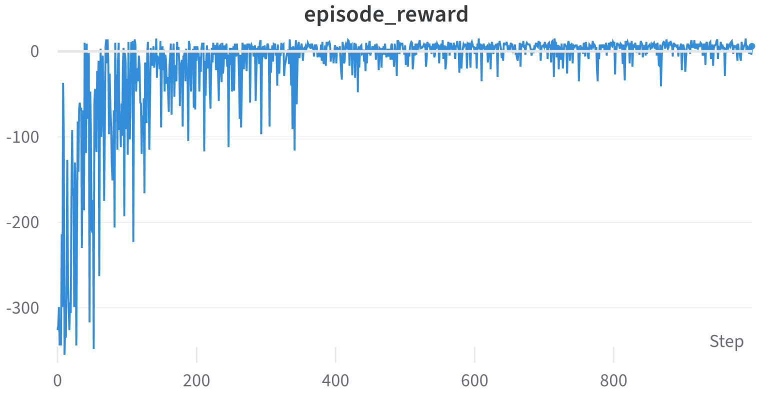
Complete Hyper parameter tuning information could be found here

Figure 9: Reward graphs of Hyper Parameter Tuning for alpha, epsilon



Reward Curve for the best Hyper parameters

Figure 10: Reward graphs for the best Hyper parameters of alpha, epsilon



Learned Policy

We have run three episodes of the learned policy, click here to play the video

Ep 1

The passenger was at Blue, the destination was Green. The taxi was spanned near Blue, it went to blue to pickup and directly went to Green to drop off.

Ep 2

The passenger was at yellow. The taxi was spanned in the Middle. The destination is red. It followed the optimal actions to reach the destination and dropped off the passenger there.

Ep 3

The source is green and destination is blue, The taxi spanned in the middle side it directly went to Blue after pickup.

Q-value visualisation

Figure 11: Optimal Q values are visualized for the states where the pickup location is Red(Top Left)

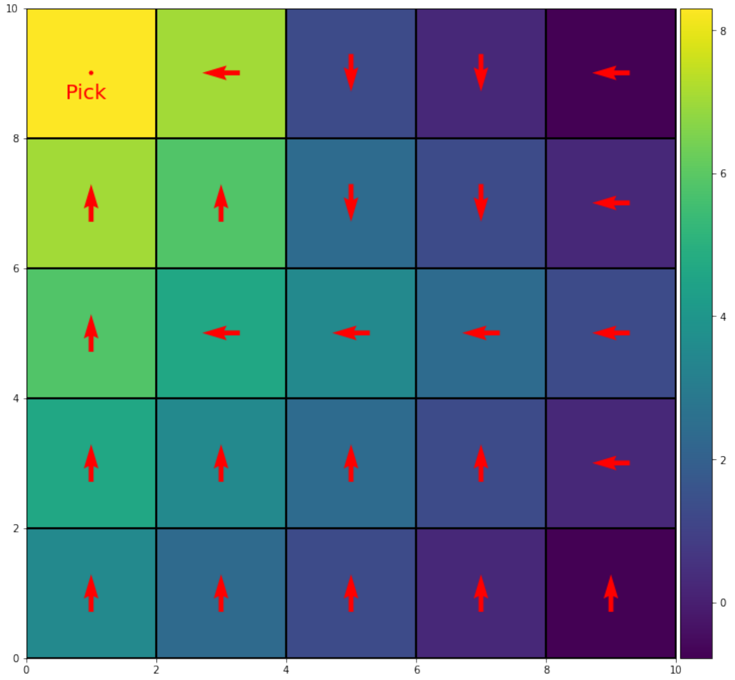
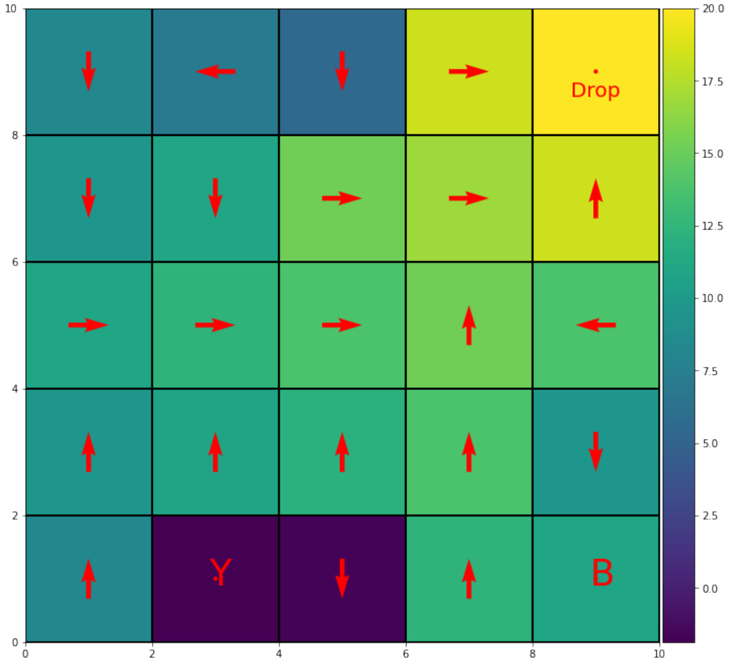


Figure 12: Optimal Q values are visualized for the states where the drop location is Green(Top Right)



Overall Policy Learnt

The taxi has learned to go the source by selecting the appropriate option at the start. Then once after the pickup it has learned to select the option to go the destination.This is almost the same for SMDP.

Inferences

1. One important difference we can observe between the optimal actions from q-values in SMDP and intra q-learning is that in the latter most of the optimal actions are primitive actions whereas in SMDP sizeable number are options. This is apparently because in intra q-learning while performing the options we also update the q-values of the primitive actions which are part of the option.

2. From the reward curves we haven't seen a significant difference between the SMDP and intra q-learning. However, in the next section we do extensive plotting of q-value tables which might point out some differences.

3. Intra q-learning converges more quickly as compared to SMDP.

Alternate Options

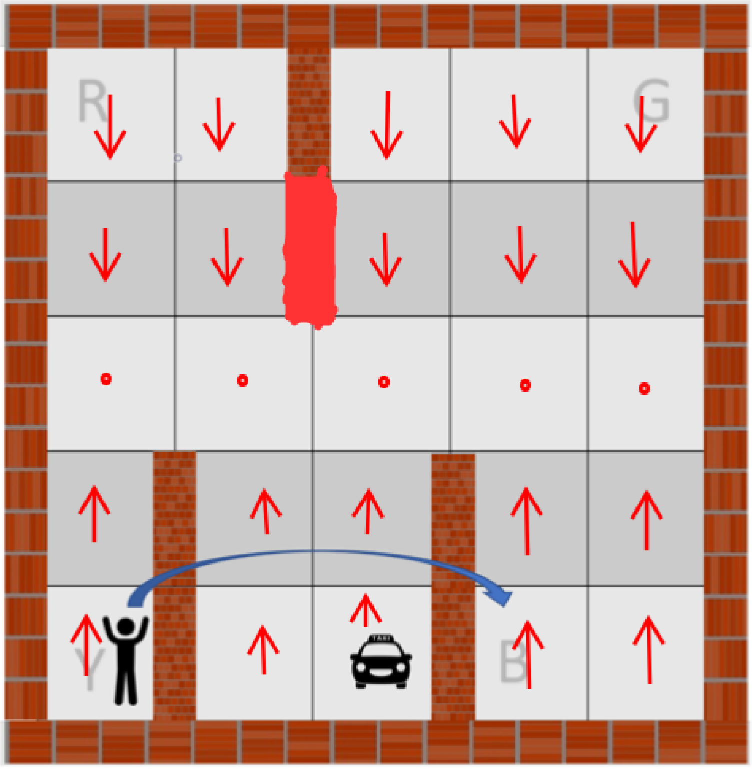
Now we take a completely new set of options(3) disjoint from the previous ones.

We could think of the entire environment as 5 compartments separated by the walls. We call the middle row as the highway, and the new options are derived from this idea.

Go to Highway

1. *Initialization set* $\mathcal{I}_o \subseteq$\{all states except the highway\}
2. *Policy*

Figure 13: Policy followed for getting on highway



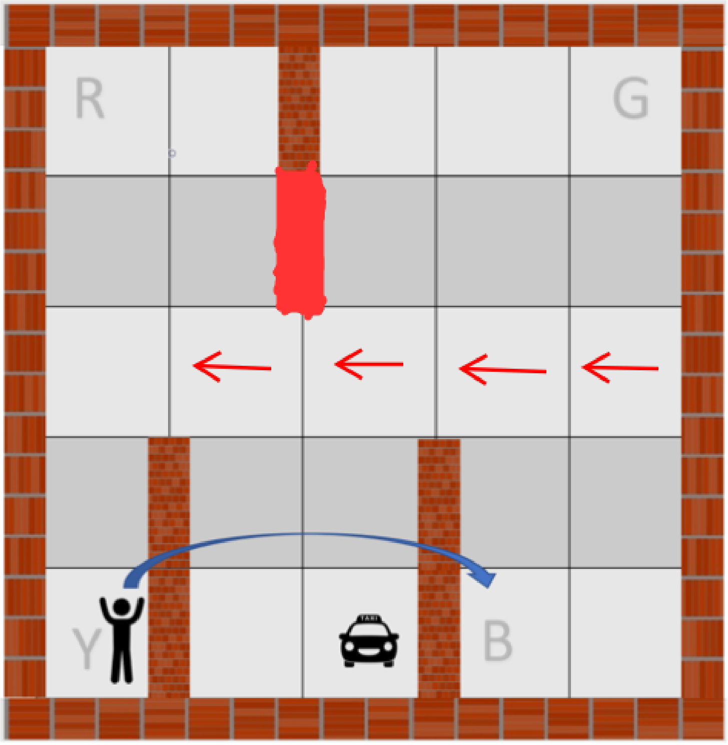
3. *Termination*

   Terminate when the taxi has reached the highway(i.e. middle row).

Go left on Highway

1. *Initialization set* $\mathcal{I}_o$ $\subseteq$\{4 states on the highway marked on the below figure\}
2. *Policy*

Figure 14: Policy followed for going left on highway



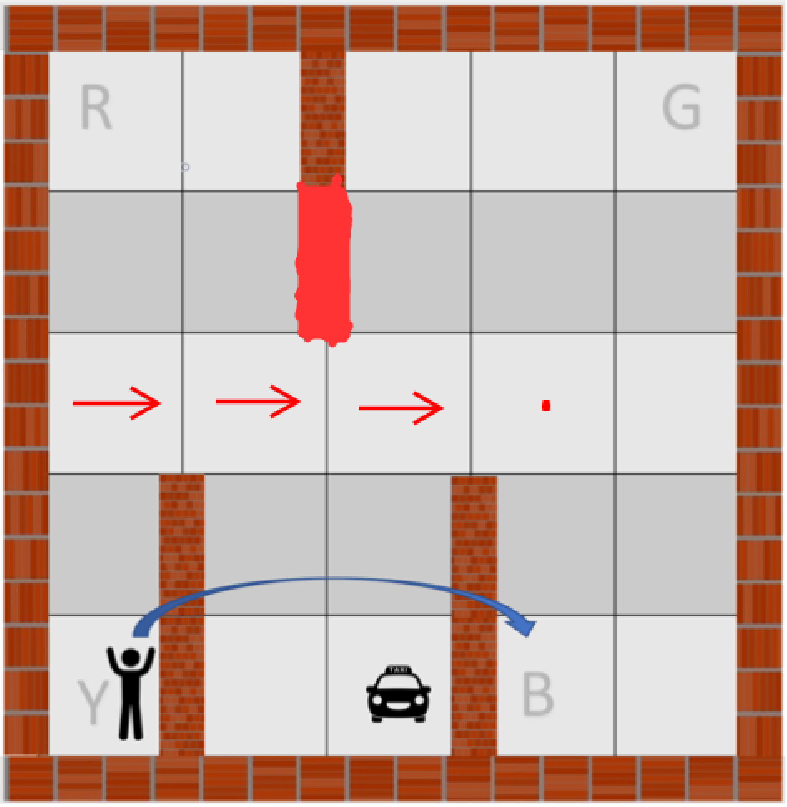
3. *Termination*

   Terminate when the taxi has reached the state-[2,0]

Go right on Highway

1. *Initialization set* $\mathcal{I}_o$ $\subseteq$\{3 states on the highway marked on the below figure\}
2. *Policy*

Figure 15: Policy followed for going right on highway



3. *Termination*

   Terminate when the taxi has reached the state-[2,3]. Here the option was terminated before reaching the right-most end to avoid a longer route to 'B'.

Hyper Parameter Tuning

SMDP Tuning

Parameter	max	min	Tuned value
alpha (α)	0.1	0.4	0. 2
epsilon (ϵ)	0	0.3	0.01
gamma (γ)	0.9	0.99	0.9

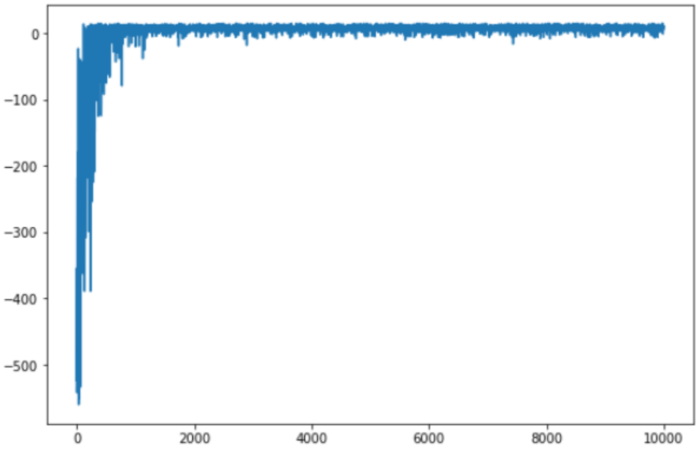
Intra Tuning

Parameter	**max**	**min**	**Tuned value**
alpha (α)	0.1	0.4	0.2
epsilon (ϵ)	0	0.3	0.001
gamma (γ)	0.9	0.99	0.93

Reward Curves

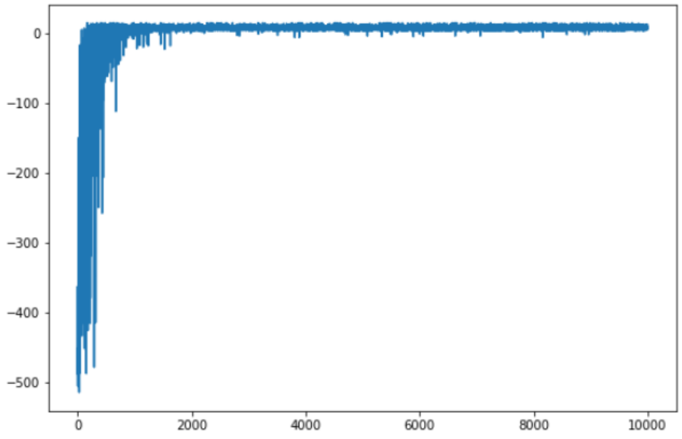
Best hyper-parameters for SMDP:

Figure 16: SMDP



The final average reward in case of SMDP is close to 7.
Best hyper-parameters for Intra:

Figure 17: Intra



The final average reward in case of Q-learning is close to 7.9(greater than the previous set of options). Also, intra Q-learning seems marginally better than SMDP, We shall plot Q-tables for all states to see further differences between the two.

Learned Policy

We have run four episodes of the learned policy, click here to play the video

Inference from the Episodes

The policy learned by the agent is very intuitive. If it is spanned at any random point, it choose to move to the highway taking the *goto hw* option. Then it chooses to take one of the two options of moving right or left on highway depending on the passenger location. It then goes to pickup the passenger. It again chooses to *goto hw*. Depending on the drop-off location, it chooses to move left or right on the highway by selecting the corresponding option. Then drops off the passenger.

Visualising Q-values

States are segregated based on passenger location(when passenger is not in taxi) or the destination(when passenger is in taxi). We plotted the heat-map of the q-values along with the optimal action/option at that location.

Symbols
H : Go to High way option
L : Go the left on Highway option
R : Go to right on Highway option

SMDP Q Value Vizualization

Figure 18: Optimal Q values are visualized for the states where the pickup location is Red(Top Left)

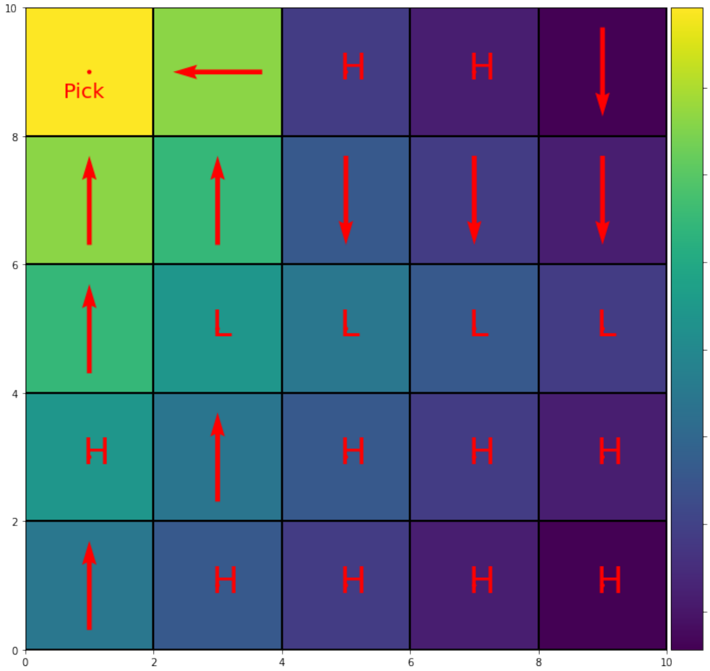
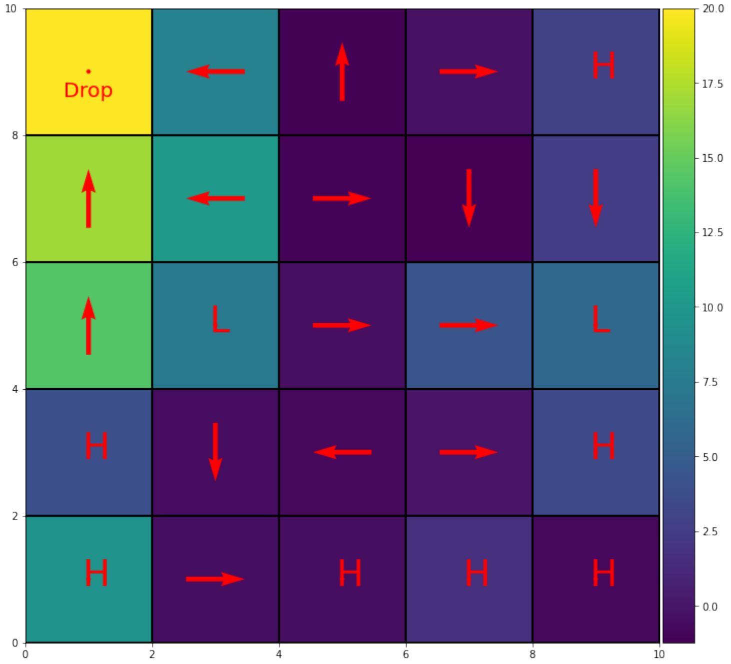
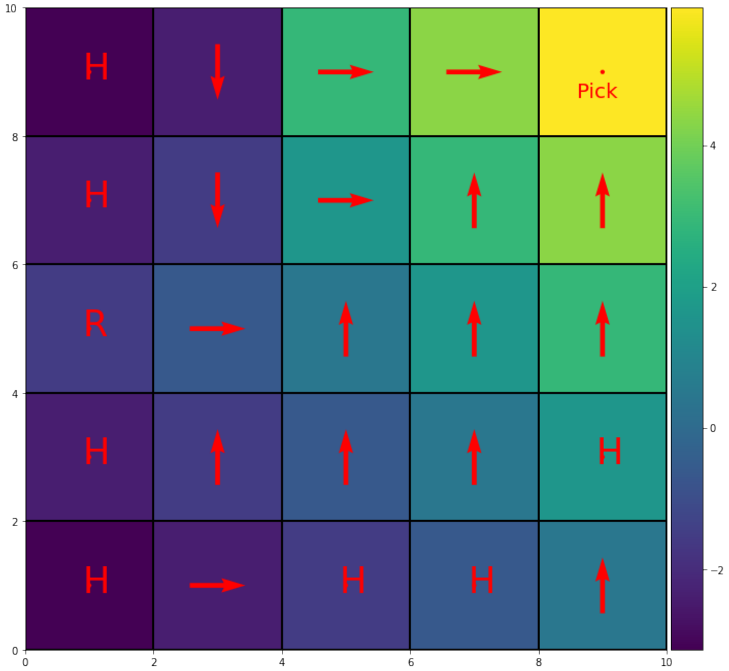


Figure 19: Optimal Q values are visualised for the states where the drop off location is Red(Top left location).



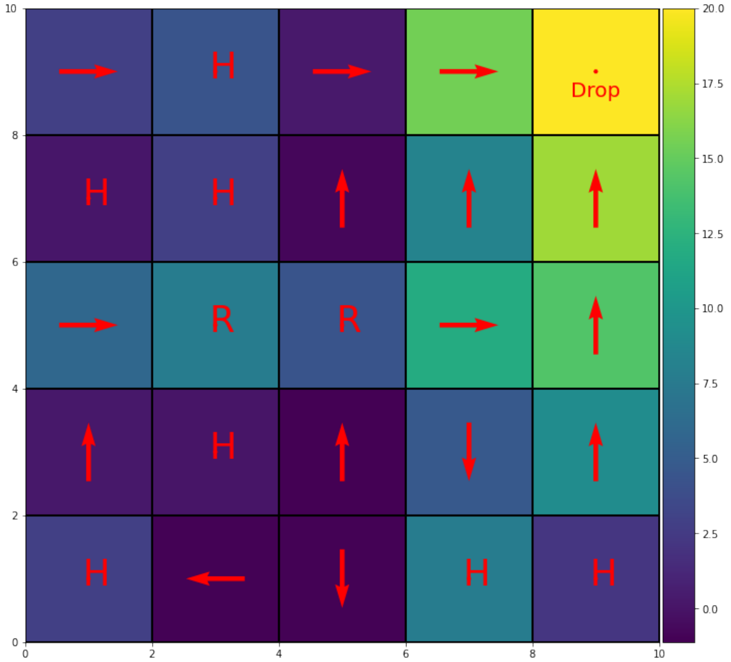
Note that we observe some discrepancies(i.e. it hasn't really learnt the best policy) in 3 states between the two walls in 4th and 5th rows. This recurs in the next couple of tables in SMDP particularly while dropping. This is in a way reasonable/ allowed because while dropping, the taxi always starts from R,G,Y,B. And from there it has to go to one among the four again. The four squares between the walls are isolated from R,G,Y,B and the chance of going through them is extremely low(that too in training phase). So as long as there is an optimal path from R,G,Y,B to destination which is learnt, we are successful, optimal actions in states not in this optimal path doesn't really matter.

Figure 20: Optimal Q values are visualized for the states where the pickup location is Green(Top right)



All optimal actions point towards G location

Figure 21: Optimal Q values are visualised for the states where the drop off location is Green(Top right location)



Here too we have optimal paths starting from R,Y,B. Some locations' optimal actions do have discrepancies(but it doesn't really matter) as discussed previously.

Figure 22: Optimal Q values are visualised for the states where the pickup location is yellow(bottom left location)

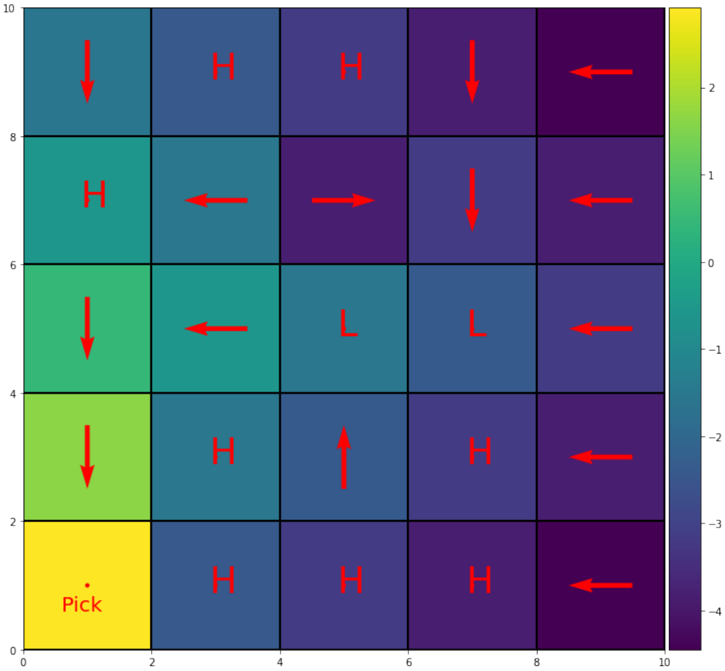


Figure 23: Optimal Q values are visualised for the states where the drop location is Yellow, (bottom left location)

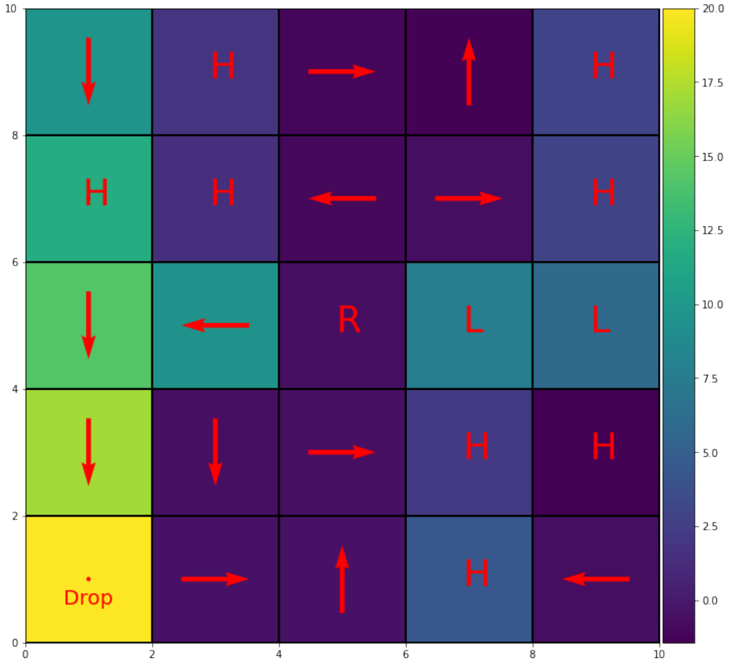


Figure 24: Optimal Q values are visualised for the states where the pickup location is Blue. It can be seen that this location was never updated at all. This is due to the fact that the system never allows for the passenger pickup location to be Blue, we have verified this across many episodes. We are not sure if this was a error in source code or this is an intentional subtitlity in gym environment

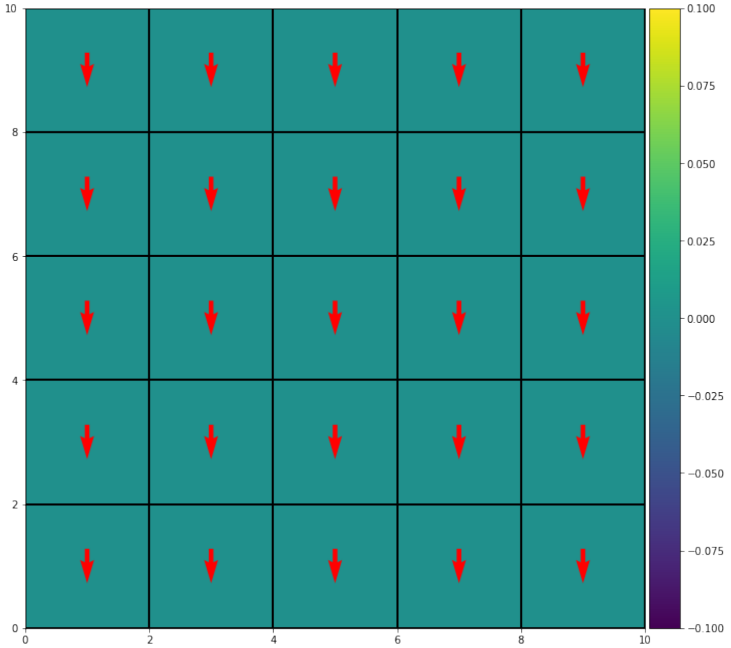


Figure 25: Here we visualized the update frequency table of the above situation to verify, it confirms our hypothesis

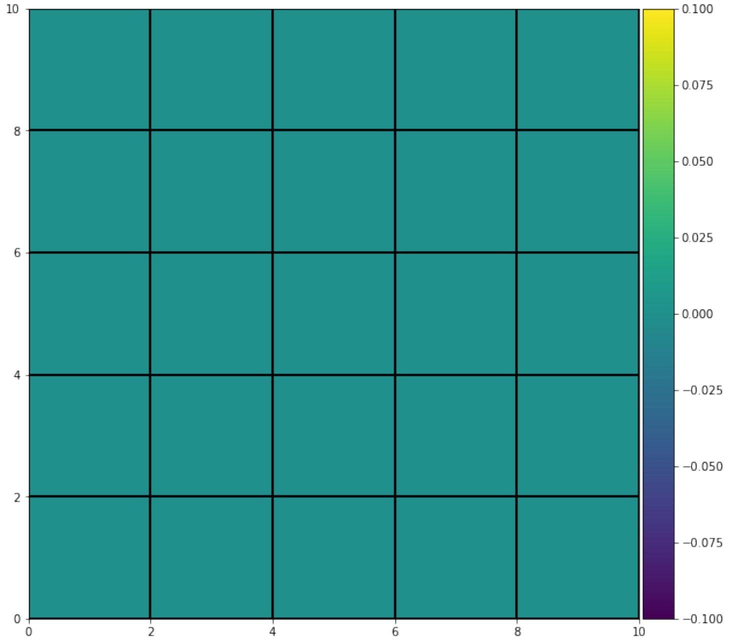
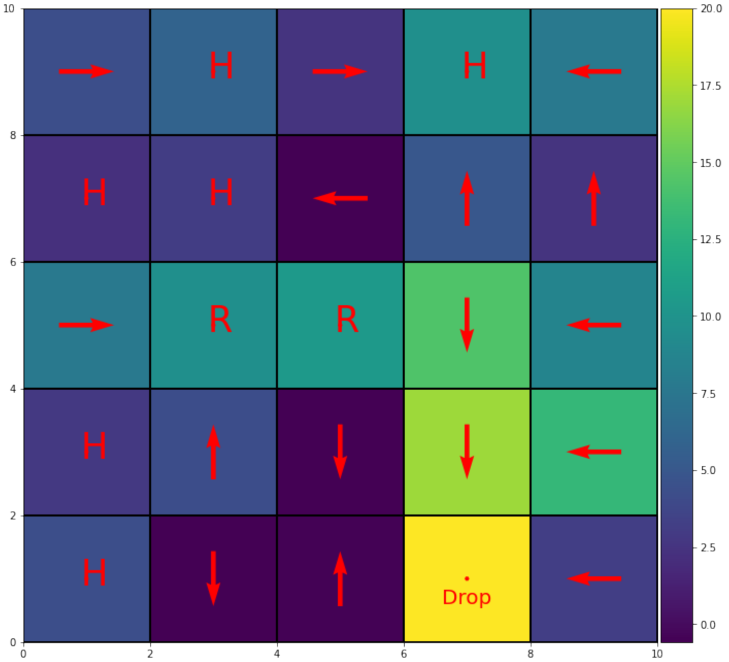


Figure 26: Optimal Q values are visualised for the states where the Dropoff location is Blue



0.0.1 Q value visualization for Intra Q learning

Figure 27: Optimal Q values are visualised for the states where the pickup location is blue, all the optimal actions definitely point to the red(top left location)

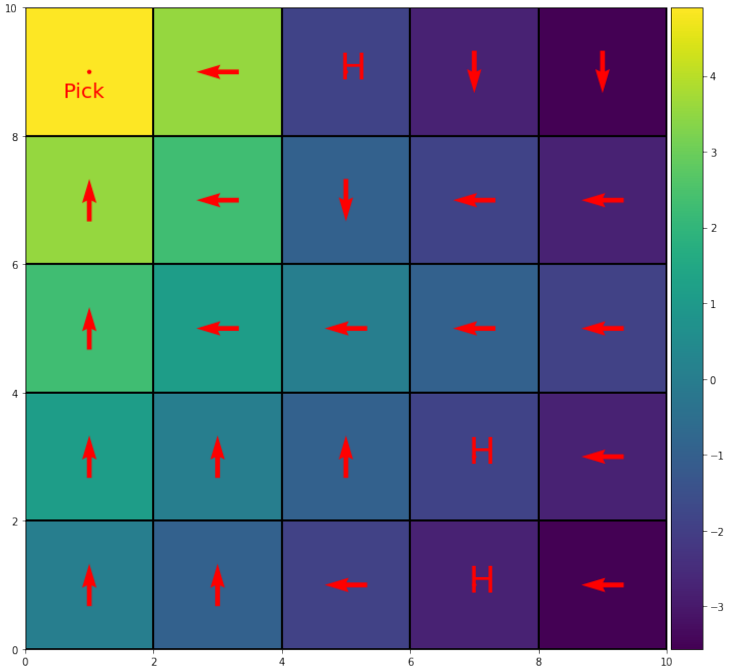


Figure 28: Optimal Q values are visualised for the states where the drop location is blue, all the optimal actions definitely point to the red(top left location)

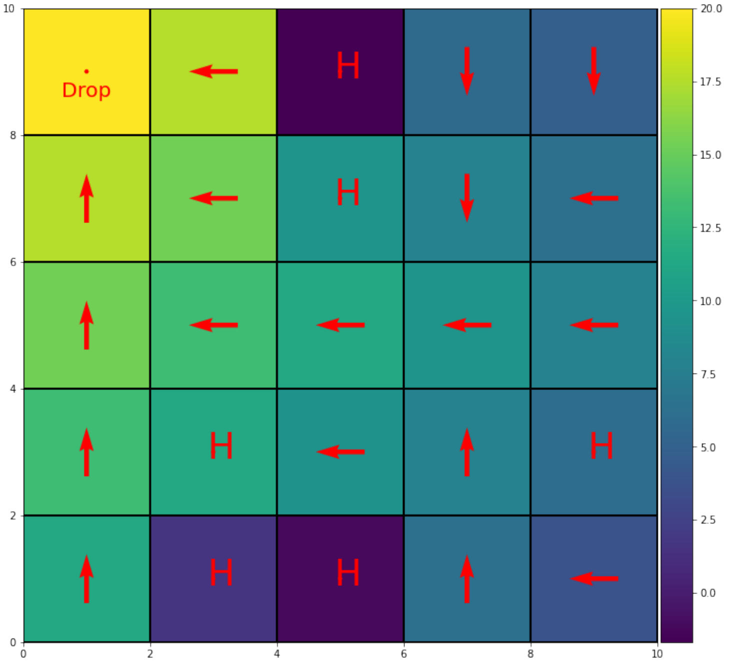


Figure 29: Optimal Q values are visualised for the states where the pickup location is green(top right) location

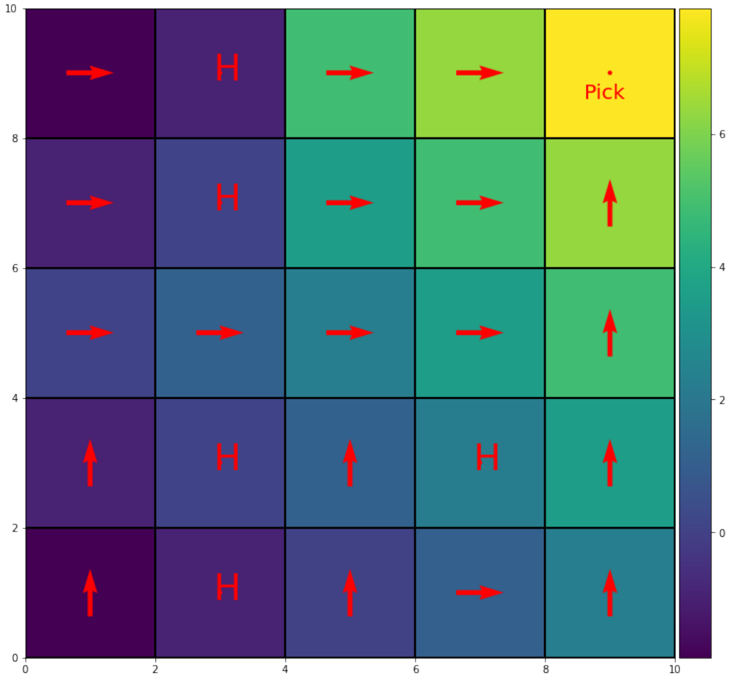


Figure 30: Optimal Q values are visualised for the states where the drop location is green(top right) location

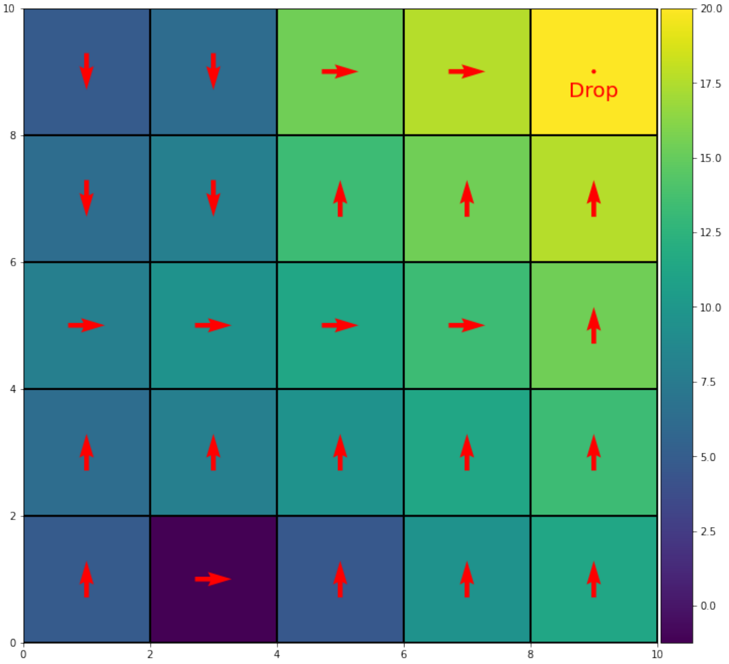


Figure 31: Optimal Q values are visualised for the states where the pickup location is yellow(bottom left) location

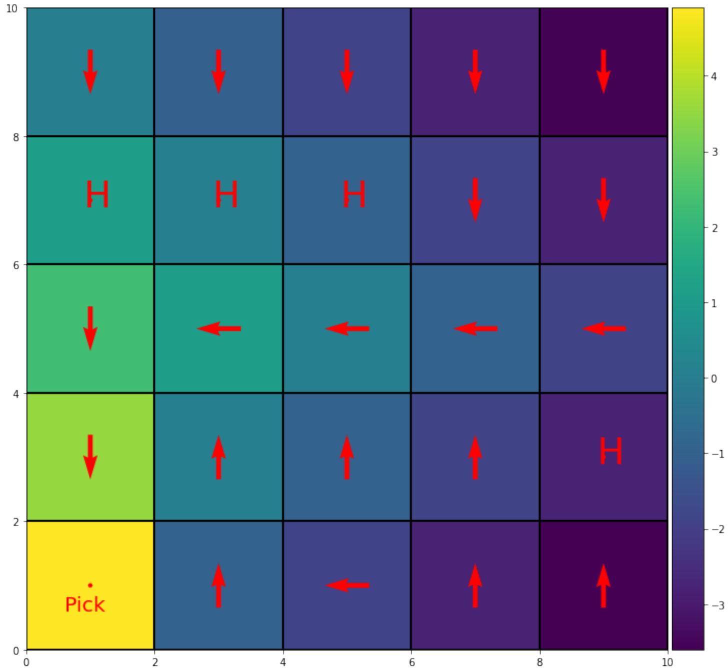


Figure 32: Optimal Q values are visualised for the states where the drop location is Yellow(bottom left) location

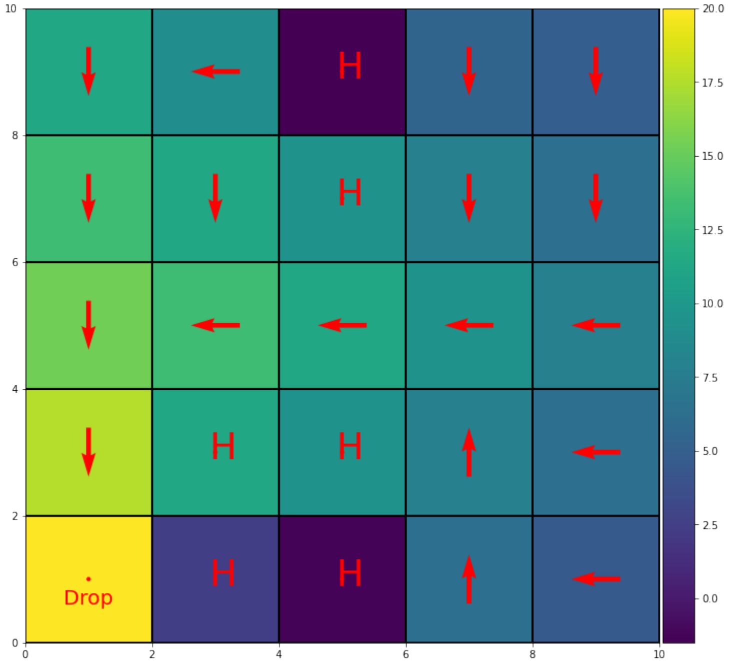


Figure 33: Optimal Q values are visualised for the states where the pickup location is blue. It can be seen that this location was never updated at all. This is due to the fact that the system never allows for the passenger pickup location to be Blue, we have verified this across many episodes. We are not sure if this was a error in source code or this is an intentional subtlety in gym environment

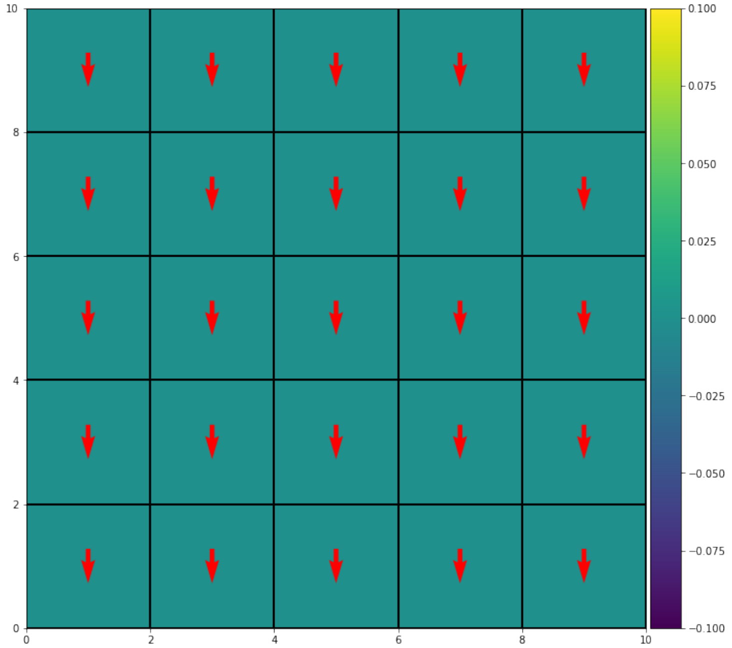
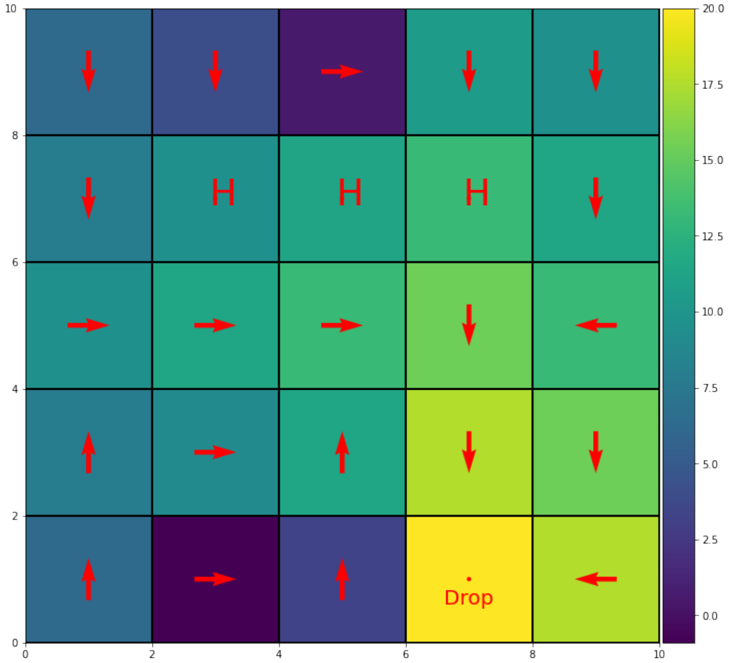


Figure 34: optimal Q values are visualised for the states where the drop location is blue



We've seen the optimal actions for both SMDP and intra q-learning in different scenarios(i.e. destination and passenger location), intra q-learning operates on more primitive actions as compared to SMDP, and also we get much better q-tables with almost no discrepancies in case of intra q-learning. Even final average reward for intra q-learning is higher.