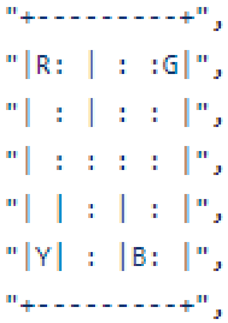
CS6700 Reinforcement Learning
Report for PA #3

Varun Gumma CS21M070
Harsha Vardhan Gudivada CS21M021

1 Introduction

In this assignment, we will be implementing different Hierarchical Reinforcement Algorithms of Options framework in a taxi domain environment. We implemented 1-step SMDP Q-Learning and intra-option Q-Learning in this environment.We used 2 sets of options while implementing the algorithms.



```
"+---------+",
"|R: | : :G|",
"| : | : : |",
"| : : : : |",
"| | : | : |",
"|Y| : |B: |",
"+---------+",
```

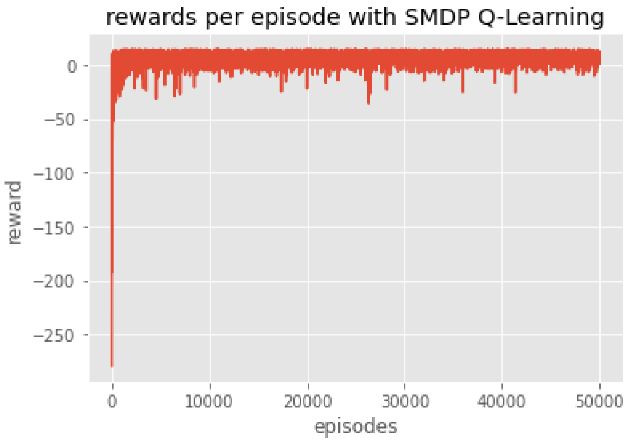
Figure 1: Taxi-v3 Environment

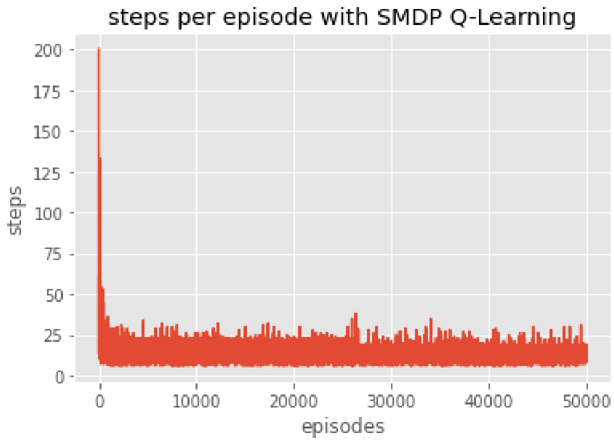
2 Primary Options

These are a set of 4 primary options. Each option uses optimal policy to reach one of the destination/pickup location.

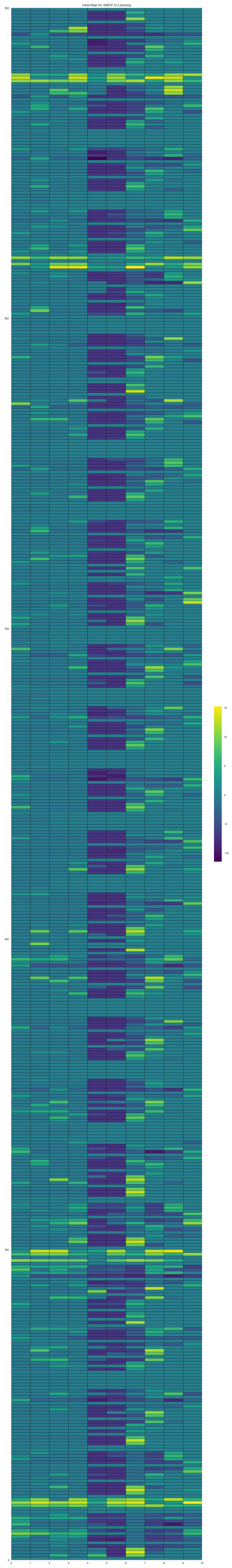
2.1 SMDP Q-Learning

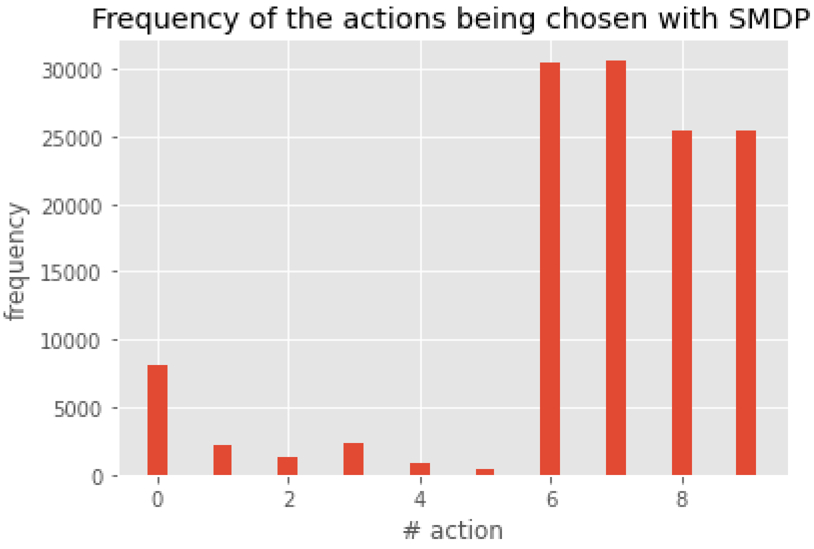
2.1.1 Reward Curve and Steps per Episode





2.1.2 Q-values and Frequency of Actions taken





2.1.3 Policy learnt and reasons

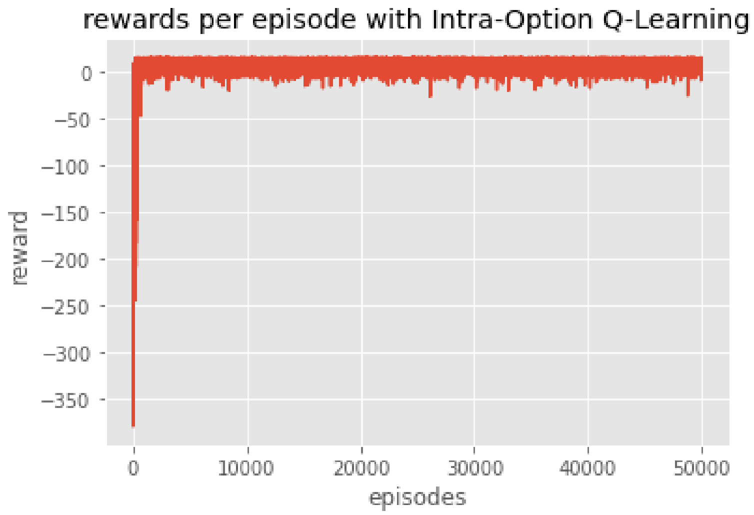
```
towards_R = [[N,W,S,S,S], [N,W,S,S,S], [N,W,W,W,W], [N,N,N,N,N], [N,N,N,N,N]]
towards_G = [[S,S,E,E,E], [S,S,E,E,N], [E,E,E,E,N], [N,N,N,N,N], [N,N,N,N,N]]
towards_Y = [[S,S,S,S,S], [S,S,S,S,S], [S,W,W,W,W], [S,N,N,N,N], [W,N,N,N,N]]
```

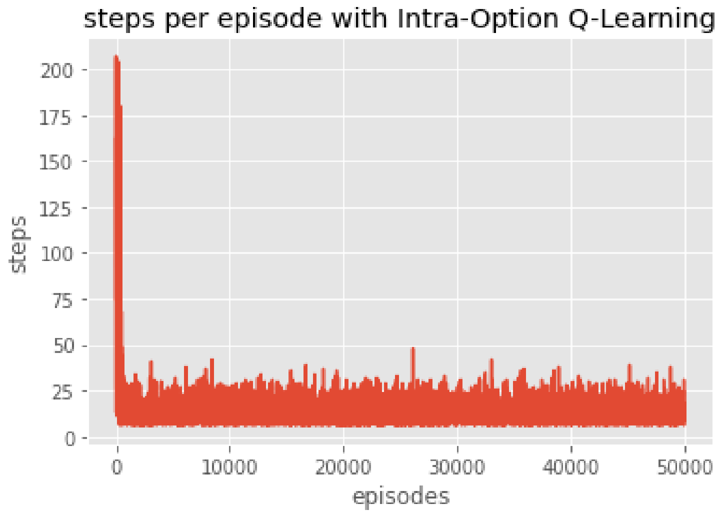
```
towards_B = [[S,S,S,S,S], [S,S,S,S,S], [E,E,E,S,W], [N,N,N,S,W], [N,N,N,S,W]]
```

Primary options are chosen such that the taxi picks the passenger and drops him in the shortest possible number of steps as seen the "guide maps" given above. The "guide maps" are in fact recursive optimal actions to be taken at each cell in the 5×5 grid to reach R/G/Y/B. For example, in state $[0, 1]$ if we take action *West* we would reach R in the shortest path. So, as the primary options are optimal, SMDP learns to pick them more frequently compared to primitive actions for achieving the highest reward, i.e. let's assume the passenger is at Y, destination is G and the taxi spawns anywhere in the world and once the training is completed and the q-values are populated, the agent will use the option to directly go to Y, perform a pickup, use the option to directly go to G and dropoff. The primitive actions in the frequency plot have very low values due as these are sometimes randomly chosen due to the epsilon-greedy strategy.

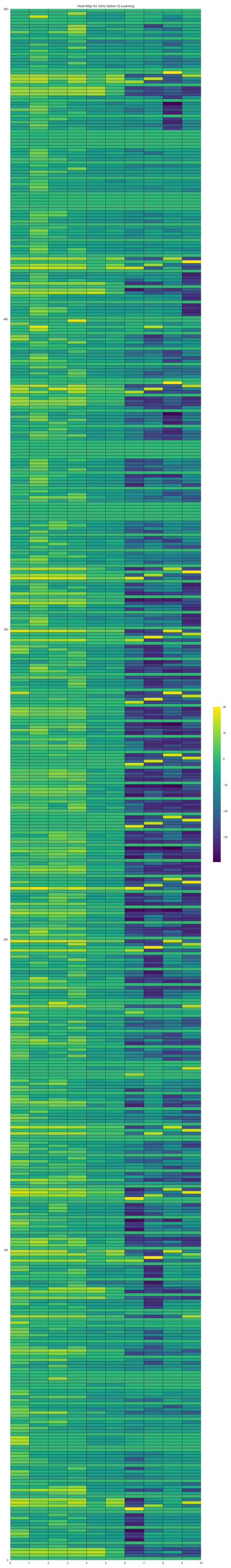
2.2 Intra-option Q-Learning

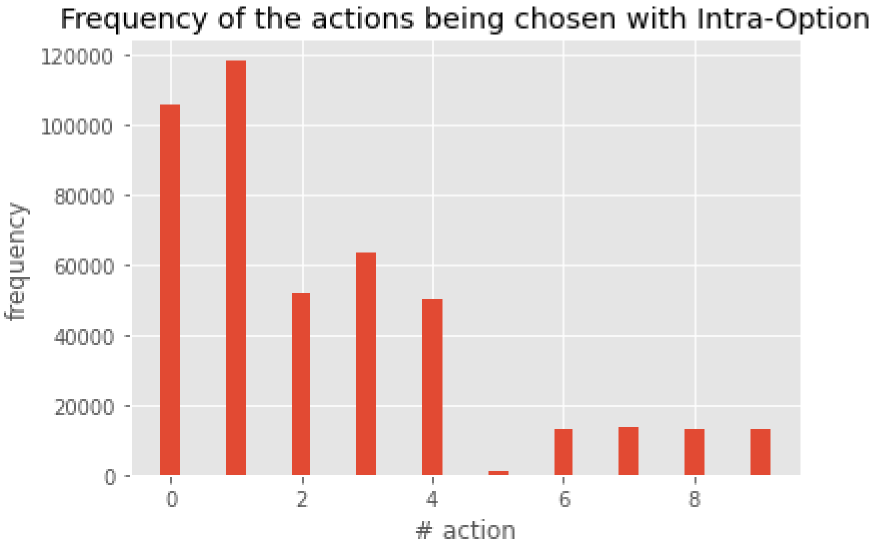
2.2.1 Reward Curve and Steps per Episode





2.2.2 Q-values and Frequency of Actions taken





2.2.3 Policy learnt and reasons

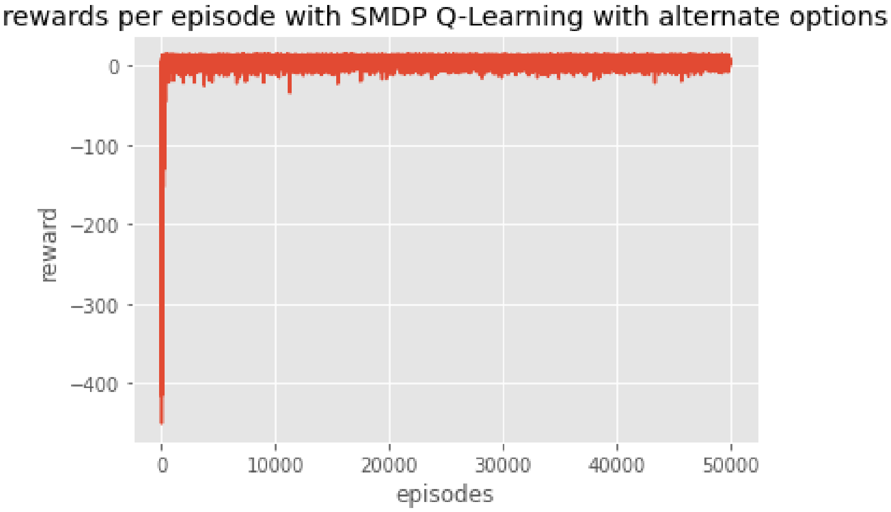
Intra-Option Q-Learning learns to pick primitive actions more frequently compared to designed options. In SMDP, choosing the optimal option updates the starting state to a large Q-value by accumulating all the rewards till the option completion. But, in Intra-Option Q-Learning, updates occur within small steps inside option, even when that option is not selected. So, this may have resulted in the final policy having more frequent primitive actions.

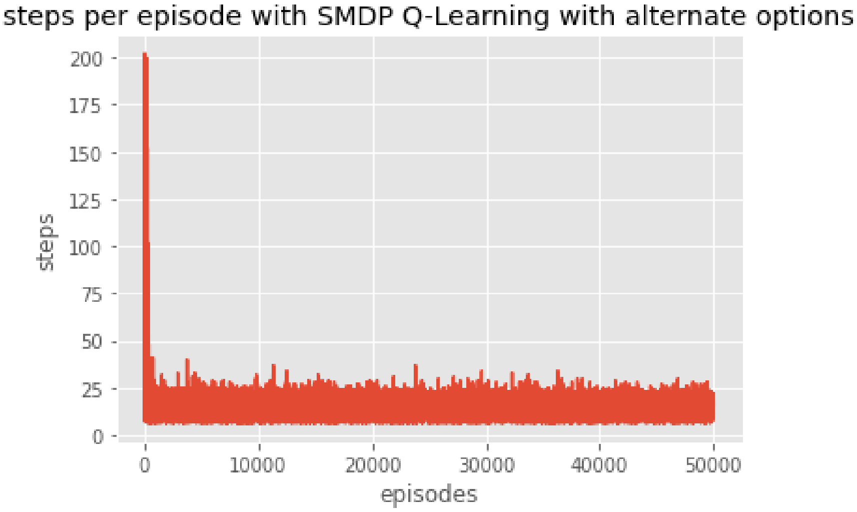
3 Alternate options

There are a set of 4 alternate options. Each option has a policy of moving continuously in one of the 4 directions till it reaches either a wall/boundary or executes a pickup/dropoff.

3.1 SMDP Q-Learning

3.1.1 Reward Curve and Steps per Episode





3.1.2 Q-values and Frequency of Actions taken

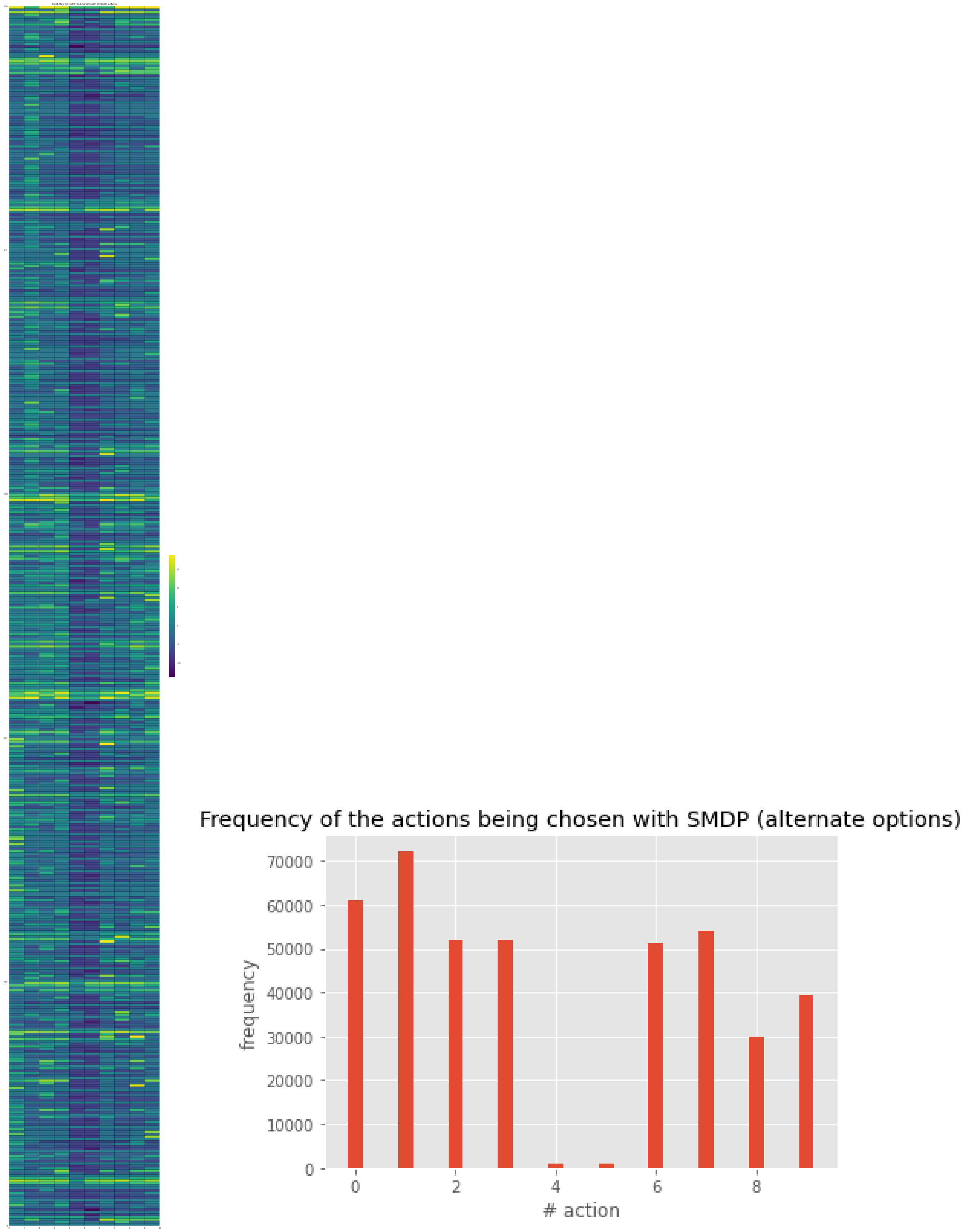


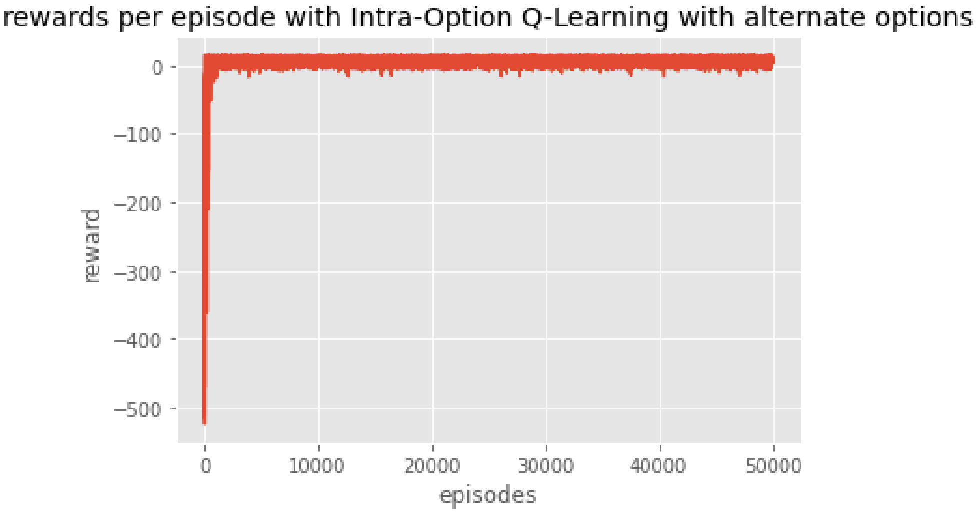
Figure 2: Learned Q values

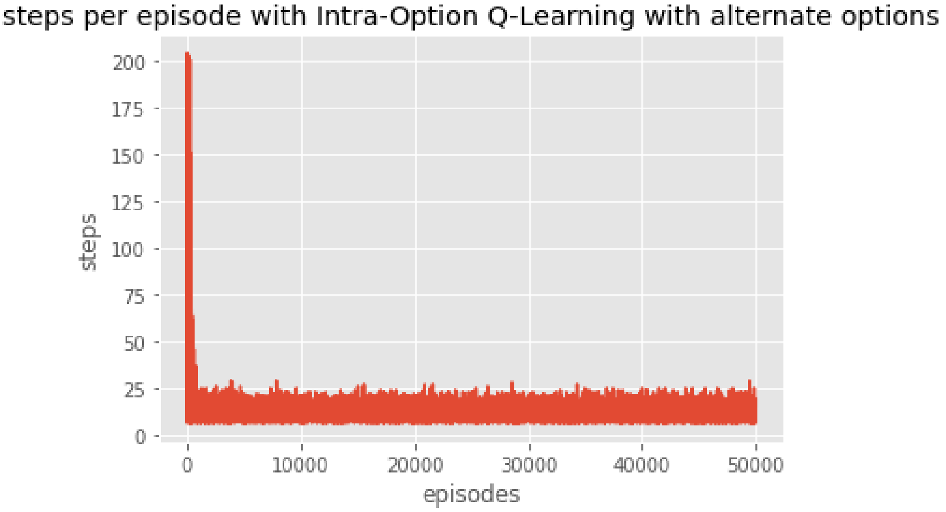
3.1.3 Policy learnt and reasons

Here we observe that both primitive actions and options are chosen almost equally. This is because the agent uses an option to move completely North/South/East/West till it hits a wall/boundary and then it uses the primitive actions to perform a direction change before moving continuously in that direction again. So, in a way, every option is mostly followed by primitive action.

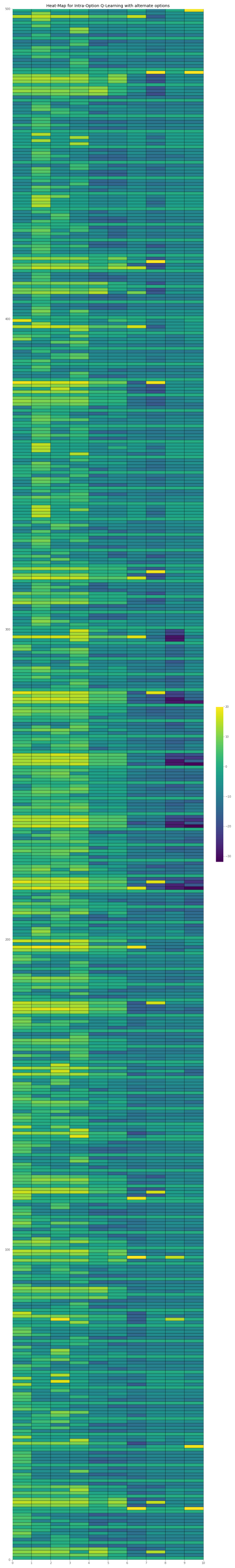
3.2 Intra-option Q-Learning

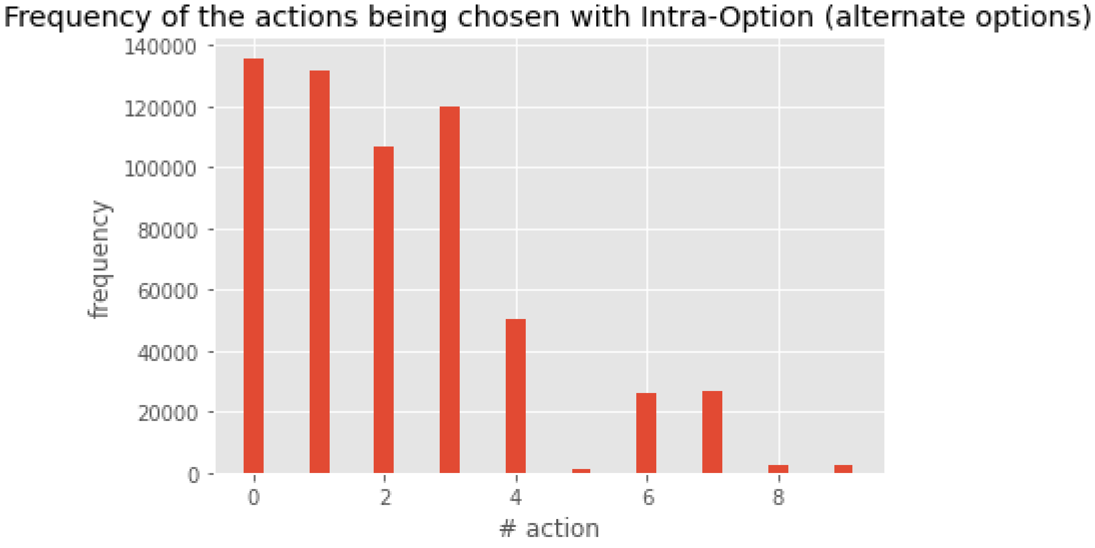
3.2.1 Reward Curve and Steps per Episode





3.2.2 Q-values and Frequency of Actions taken

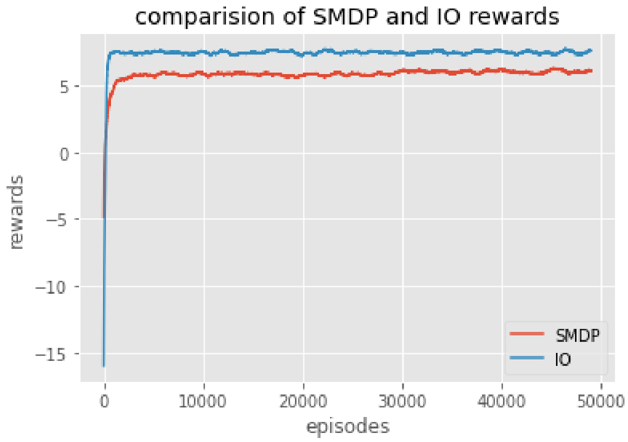


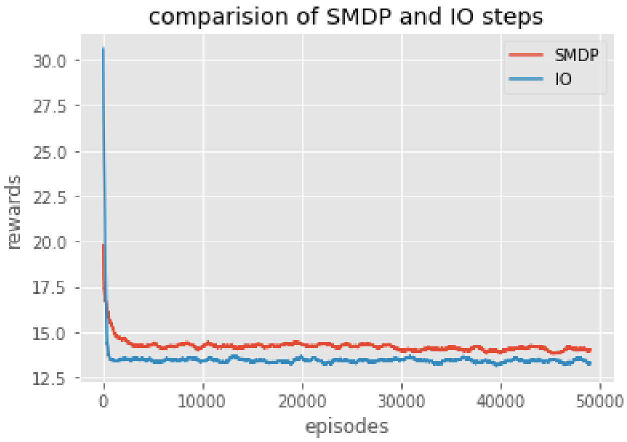


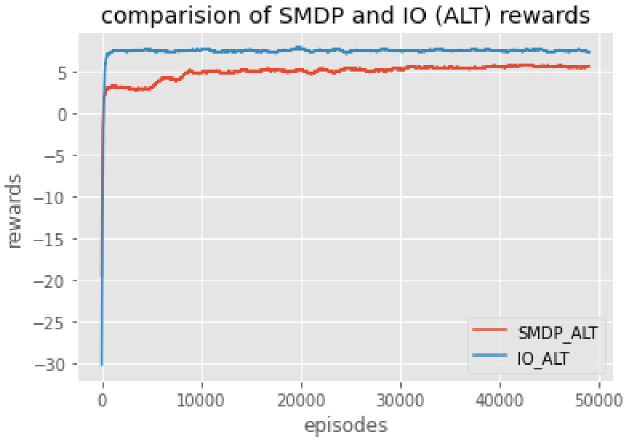
3.2.3 Policy learnt and reasons

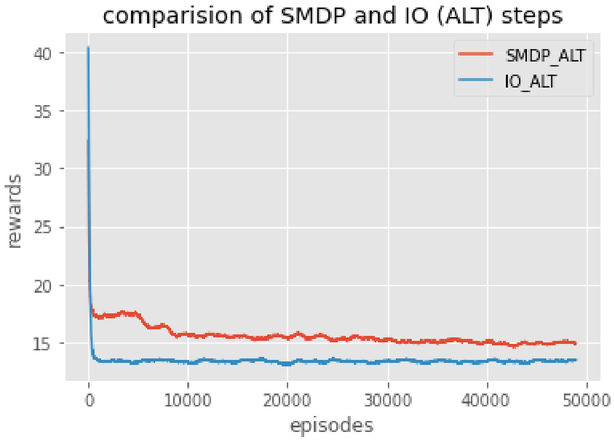
Intra option Q learning learns a policy of choosing primitive actions more frequently. In it, small steps inside the option are also used for updating those Q values, even when that option is not selected. This may have resulted in the policy of choosing primitive actions more frequently.

4 Comparison between SMDP and Intra-option Q-learning







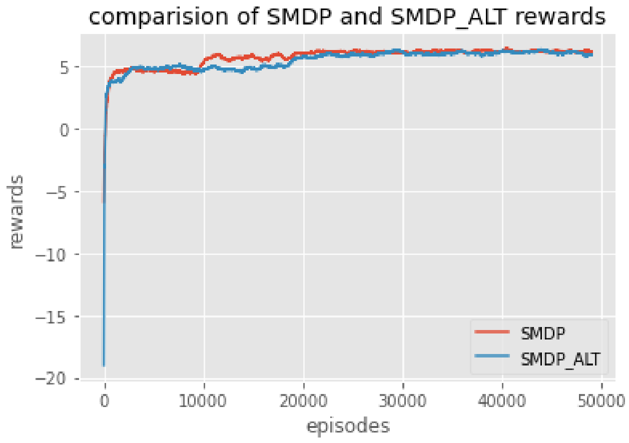


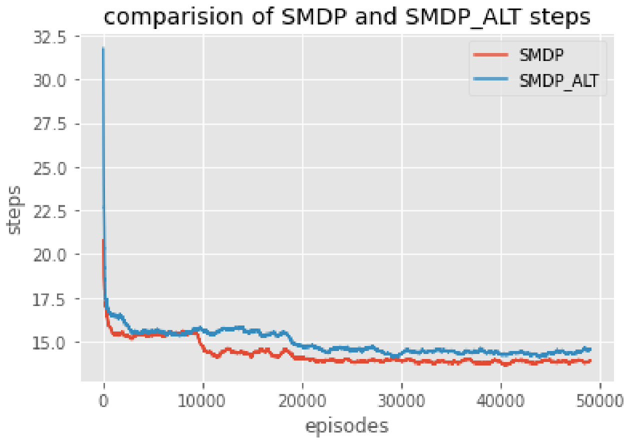
l

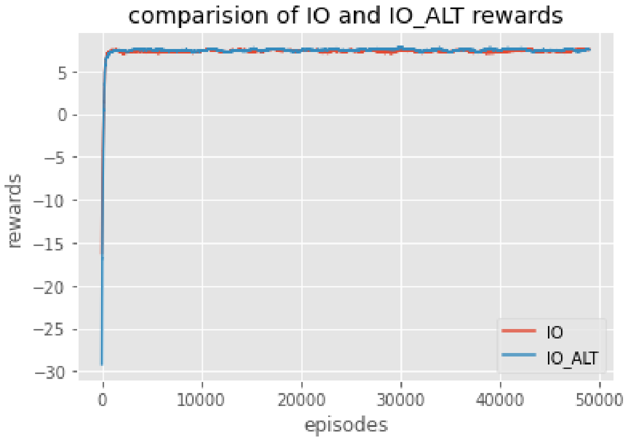
1. For both primary options and alternate options, rewards of Intra-Option Q-Learning algorithm are more stable across the episodes after converging.

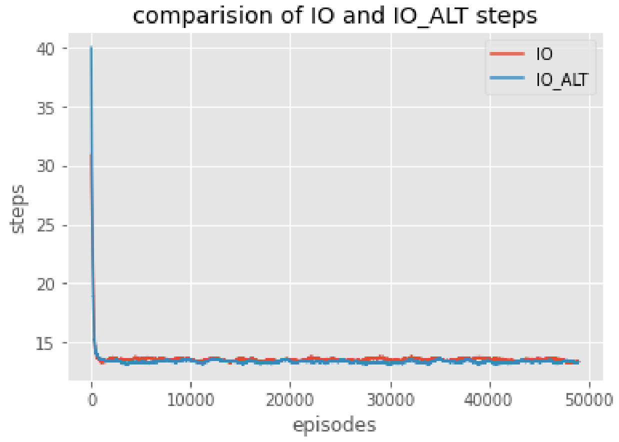
2. Intra-Option Q-Learning is performing better than SMDP. In SMDP, option is required to be run till termination for the update. Whereas in Intra-Option Q-Learning, we learn about the option even from small fragments of experience. This learning of option even before it terminates may be the reason for the better performance of Intra-Option Q-Learning.

5 Comparsion between Primary and Alternate Options









1. For both options, as we run for large number of episodes, we find that they converge similarly.

2. We can observe that the primary options can be simulated using the alternate options and primitive actions. For example, some pieces of the option `towards_R` can be replaced with the alternate option `Opt_North` (as seen in the code). Therefore, we observe the performance of both options is similar.