CS747 Assignment 3

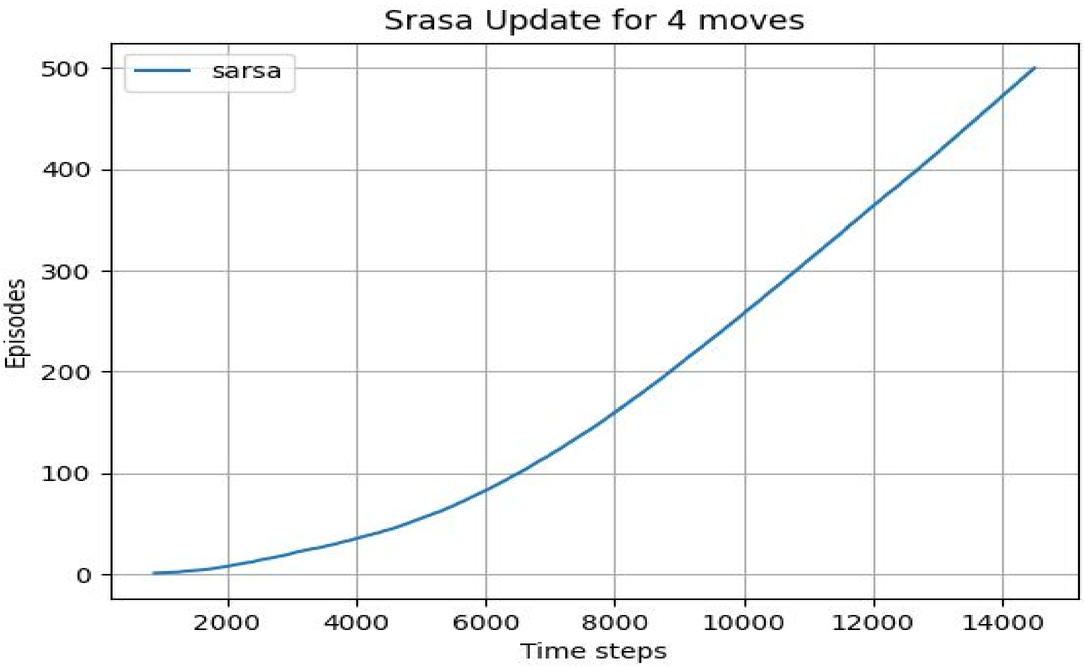
Manoj Bhadu(170010036)

Approaches:

1. I used a step function which gives the next states taking state, action, and stochasticity as input. I have taken all corner cases by taking min and max with 0 and height or width
2. Implement episode function for all three algorithms named Sarsa, Expected Sarsa, Q learning update. This function takes q values(initialized to zero) and gives time steps in one episode for reaching the goal state. Taking -1 as reward for all states except Goal state.
3. Plots function takes input of update used and stochastic for 4 moves and for kings moves it takes only stochasticity as input(either 0 or 1) this returns the time steps after every episode and plot of them. I have used the 500 episode limit and 10 seeds for every episode and taken the average of them.
4. Epsilon and alpha values are 0.1 and 0.5 respectively for all tasks.

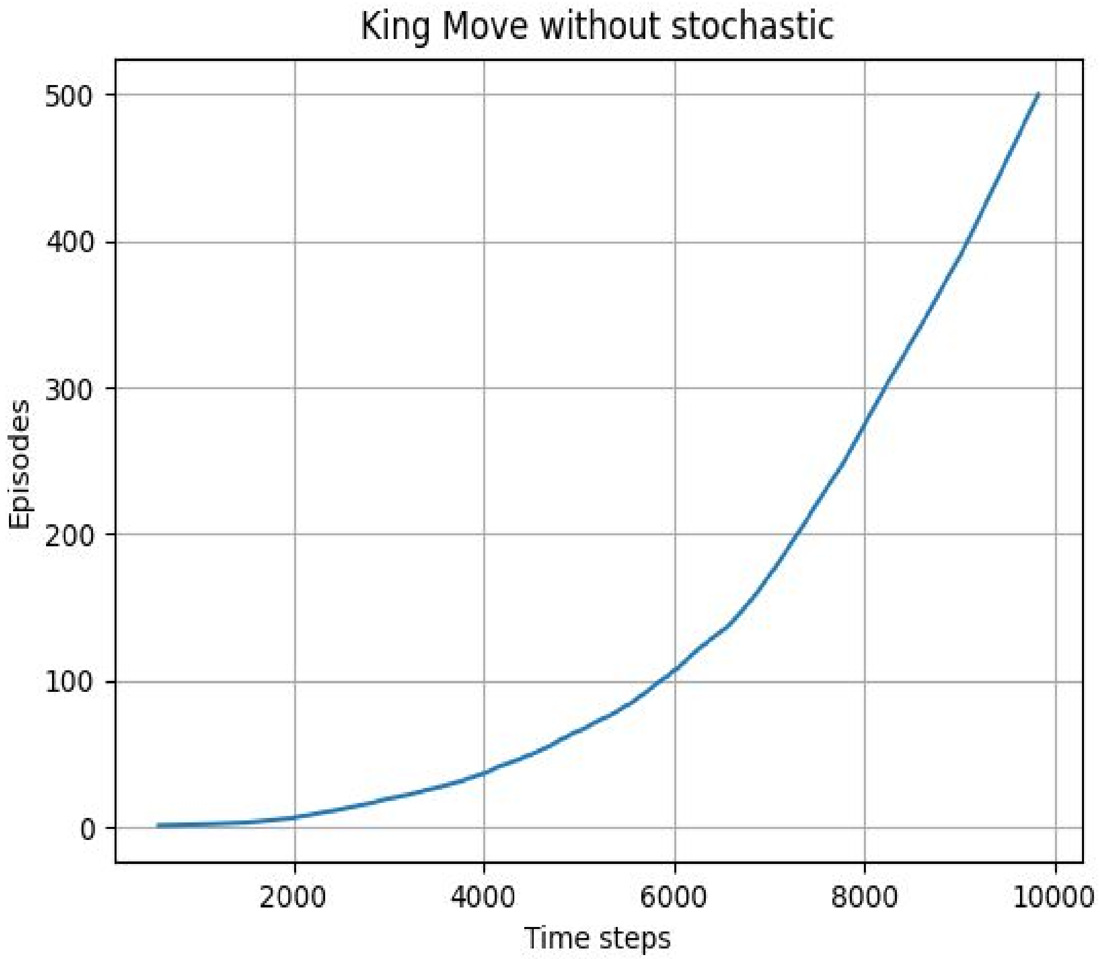
Plots and Observations:

1. **Task2: Sarsa for 4 moves**



Comments: As we can see this is quite similar to the baseline plot given in the book. Initially slope is increasing as we know that the goal state is reaching sooner as episode is increasing means taking lesser time. This is because of Q value which is converging.

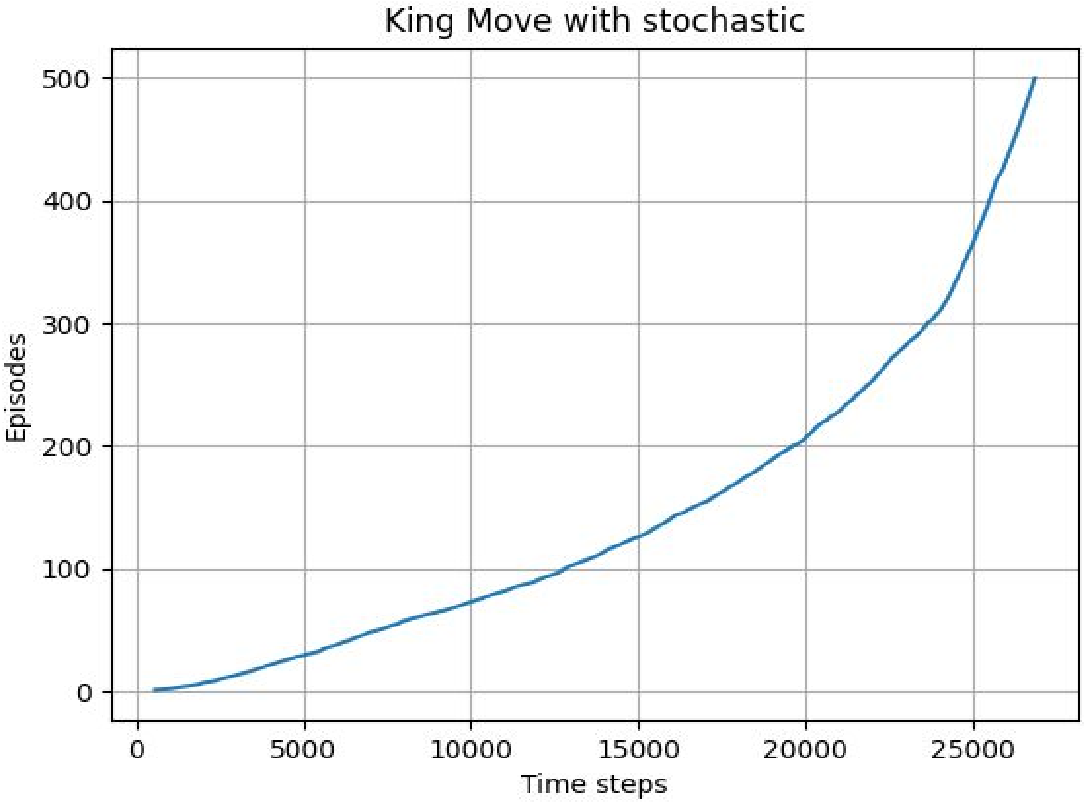
2. Task3: Windy Gridworld with King Move

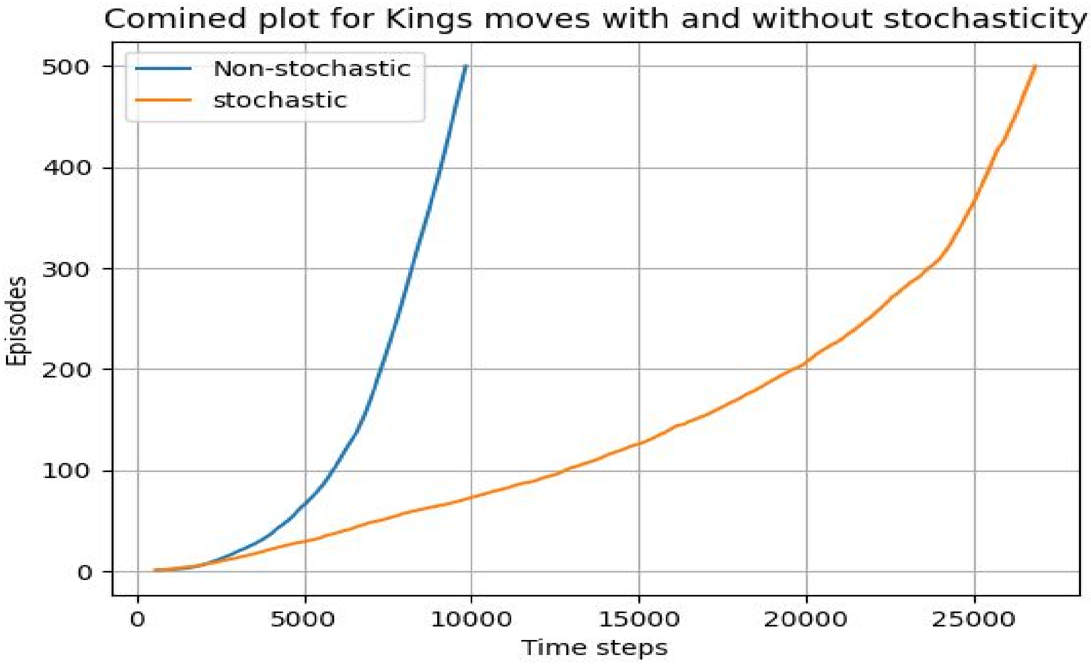


Comments:

We can see that the slope for King moves allowed is more steep than the 4 moves allowed. This is because the Goal state is reaching sooner with less time taken. Also Q value is converging fast as we can see that the plot slope is very steep after 8000 steps.

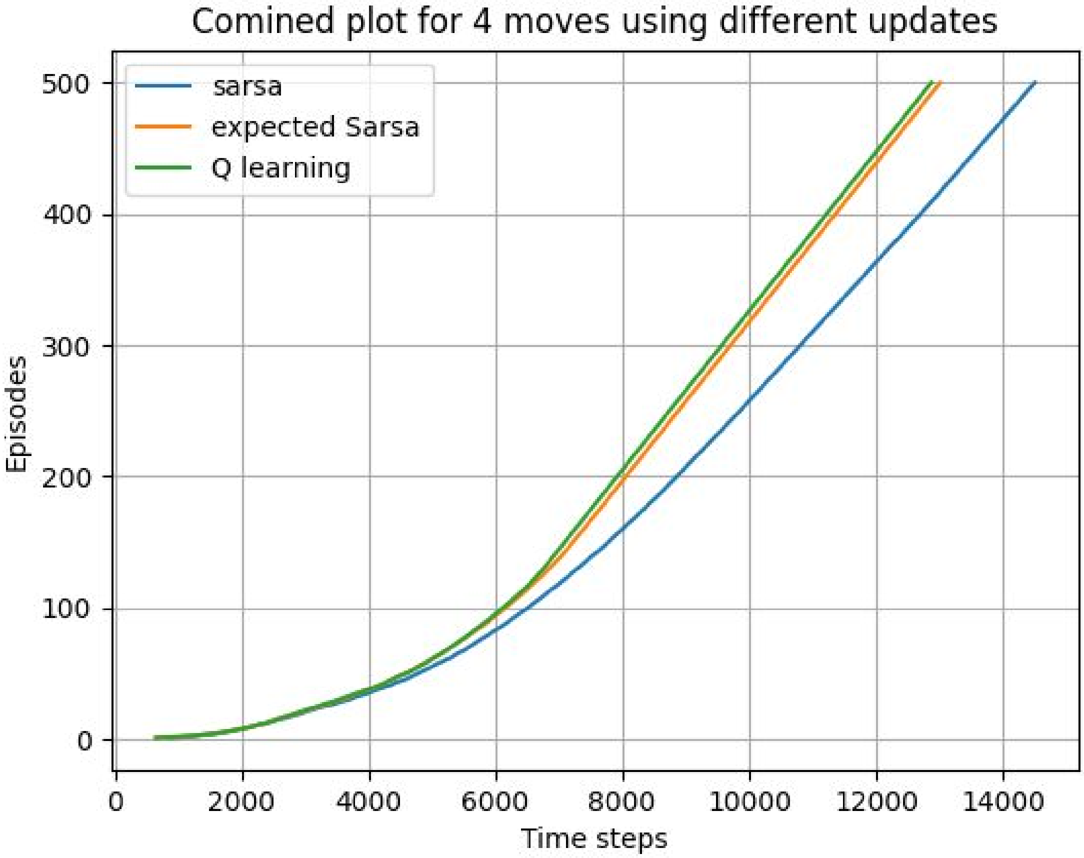
3. Task4: Kings Move with Stochastic:





Comments: as we can see from above plot that the convergence for Kings Move with Stochastic taking much more time steps to converge. It means for every episode is taking much more time steps to reach Goal state, this is due to randomness of Stochasticity.

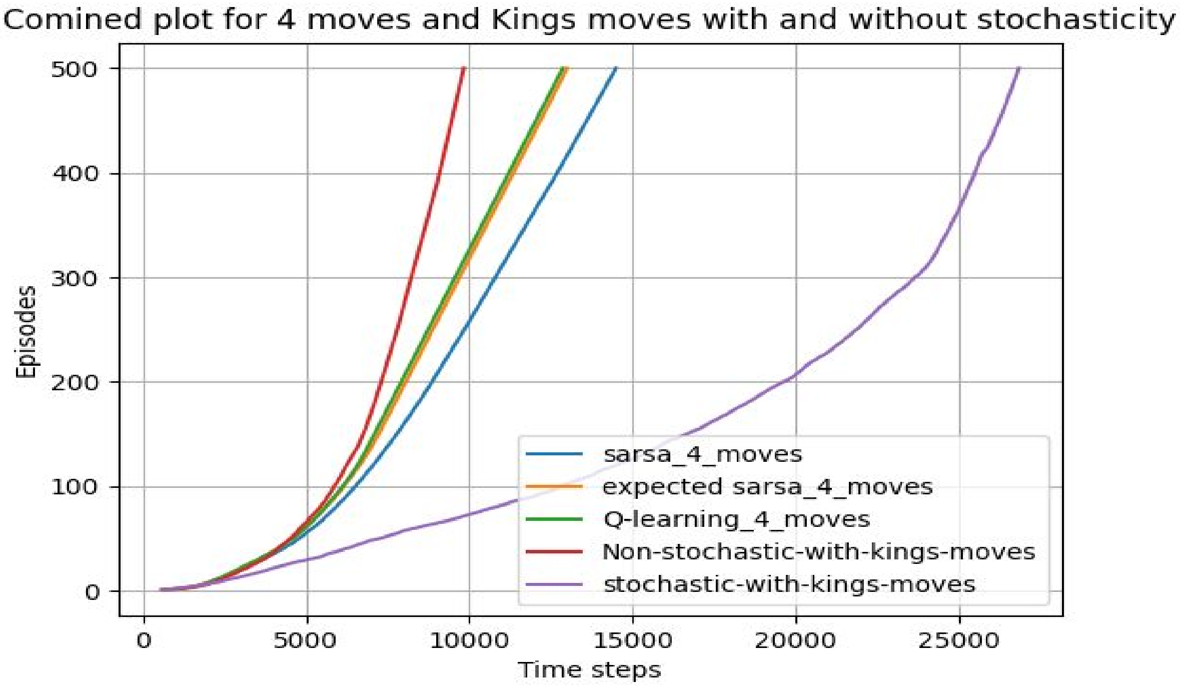
4. Task5: Expected Sarsa and Q-learning agents for 4 moves



Comments: We can see that Q learning works better then Sarsa and Expected Sarsa both. On-policy SARSA learns action values relative to the policy it follows, while off-policy Q-Learning does it relative to the greedy policy, so Q learning works better than sarsa and expected sarsa.

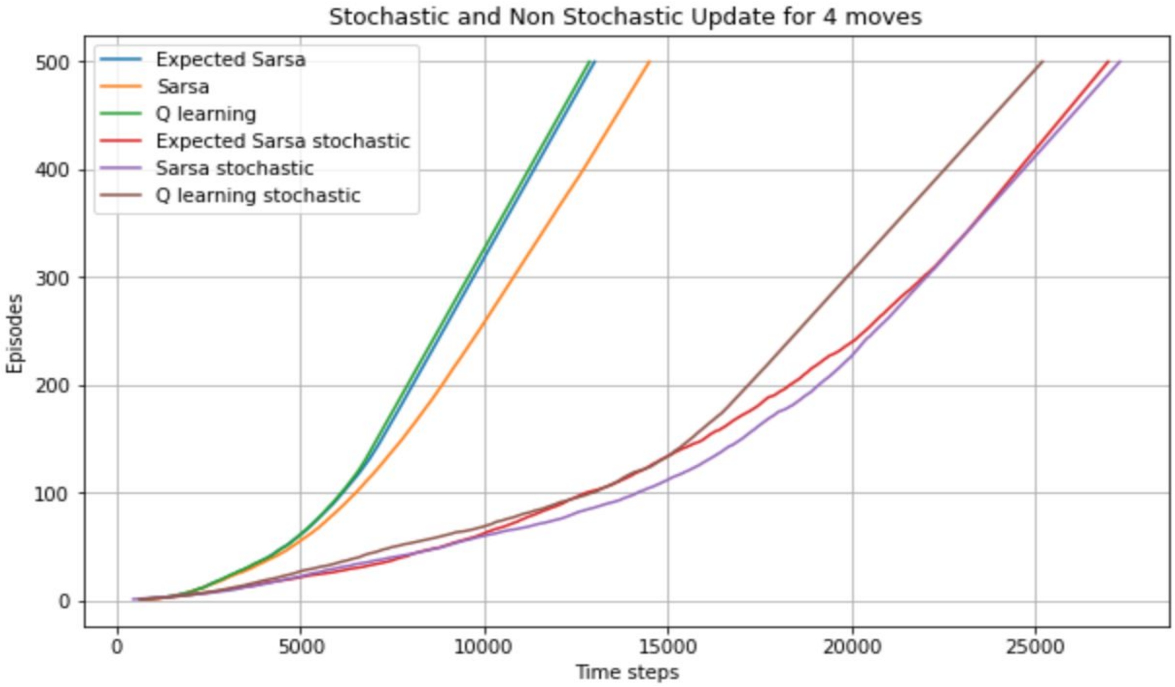
5. Additional Plots:

- **Combined Plots of when Only 4 moves allowed and when Kings move allowed**



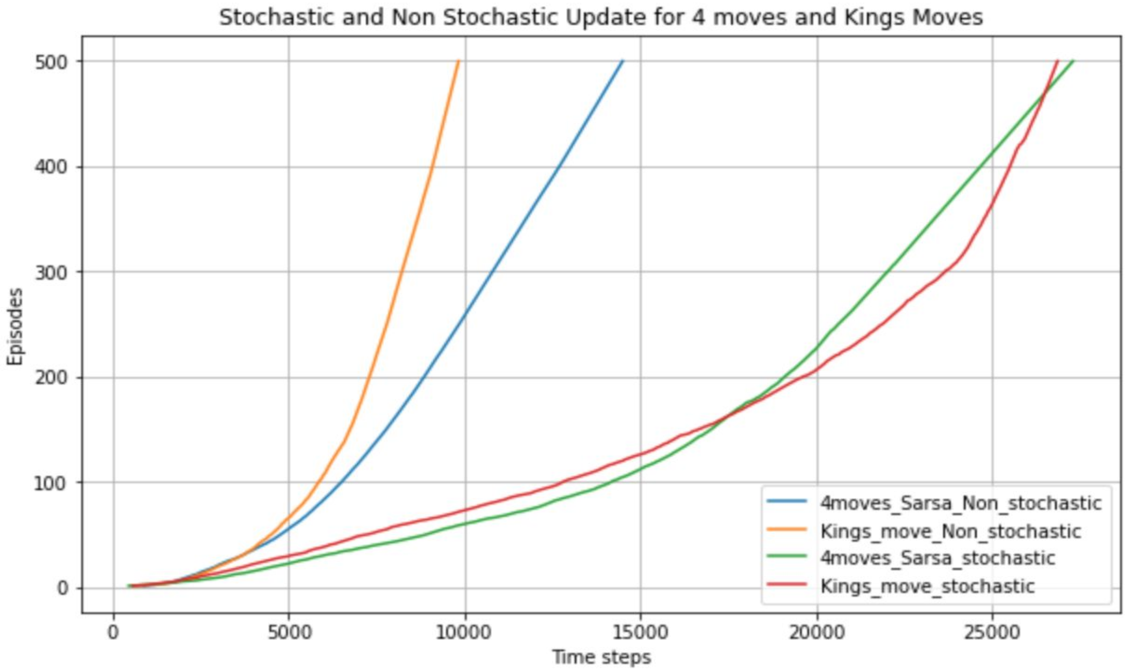
Comments: Sarsa for Kings move is performing better than Q learning. This is obvious due to more actions allowed.

- **Stochastic and Non-Stochastic For 4 moves for different Control Algorithms**



Comments: Non Stochastic update is converging much faster than stochastic.

- **Stochastic and Non Stochastic Comparison for only 4 moves and Kings Moves**



Comments: Kings Move with non-stochastic performing best, this is reasonable because more action available and no stochasticity so giving best results, followed by 4 moves non-stochasticity which is also obvious. But Kings Moves stochastic initially performing better, but in time steps 17000 to 28000 not performing better, which is surprising and I was not able to find reason for this. In long term the KIngs move will perform better which is reasonable to assume.