Reinforcement Learning Assignment 2 Report

In this report, I'll compare the implementation of two alternative RL algorithms in a windy grid world, each with a unique set of actions and an epsilon-greedy selection policy.

We have two algorithms for two different approaches with and without king's moves: (SARSA, Q-learning)

1- Without king's moves actions set: **(['UP','DOWN','LEFT','RIGHT'])**
2- With king's moves actions set: **(['UP', 'DOWN', 'LEFT', 'RIGHT', 'UP-right', 'UP-left', 'DOWN-right', 'DOWN-left'])**

There are Some fixed values for both of these algorithms such as Number of episodes = 1000 and Gama = 0.9 (We chose 0.9 because the doctor had mentioned that the best gamma value is 0.9 or 0.95.)

Without king's moves approach:

Algorithms	Alpha	Epsilon	The Total Numbers of Reward	The Total Time Steps to converge	Number of episodes to converge
SARSA	0.5	0.1	-15	8679	264
Q-Learning	0.5	0.1	-16	4797	101
SARSA	0.5	0.1	-14	18540	778
Q-Learning	0.5	0.1	-14	5105	112
SARSA	0.5	0.2	-14	11323	227
Q-Learning	0.5	0.2	-18	4828	86

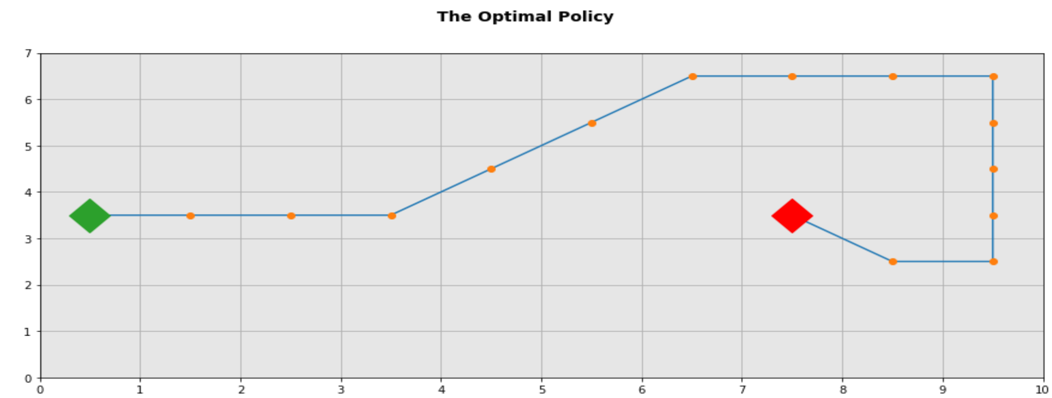
Observations : Results will vary due to randomness, but usually after ~4500 time steps, the learned policy is optimal and finishes the episode.

As we can see, for the same epsilon value, Q-learning performs better than SARSA on the Windy Grid world. It learns the optimal policy quicker than SARSA.

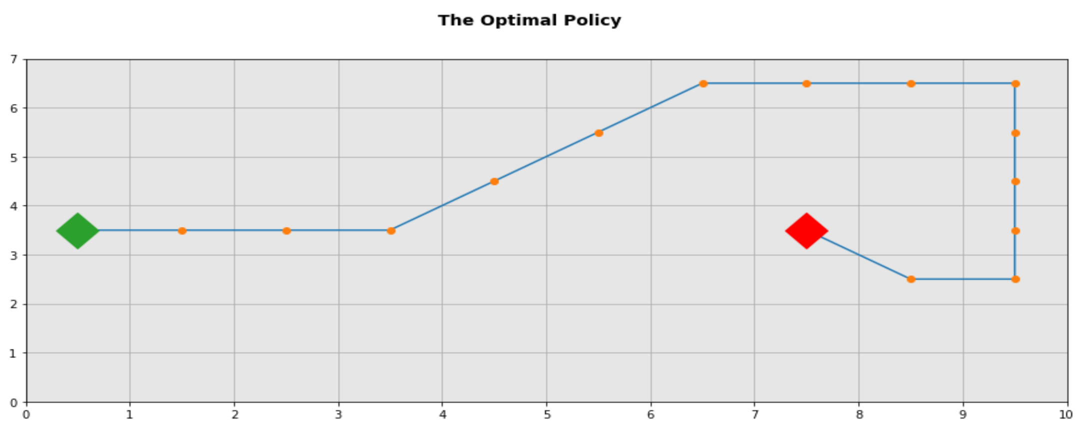
Also, ε = 0.1 gives better results, as is expected. ε = 0.2 doesn't seem to give any major advantage in the early phase of learning, as could of been the case.

Maybe a higher ε value would make a difference, but then again, in the early phase, the greedy actions are not set in stone yet as the optimal policy is far from learned.

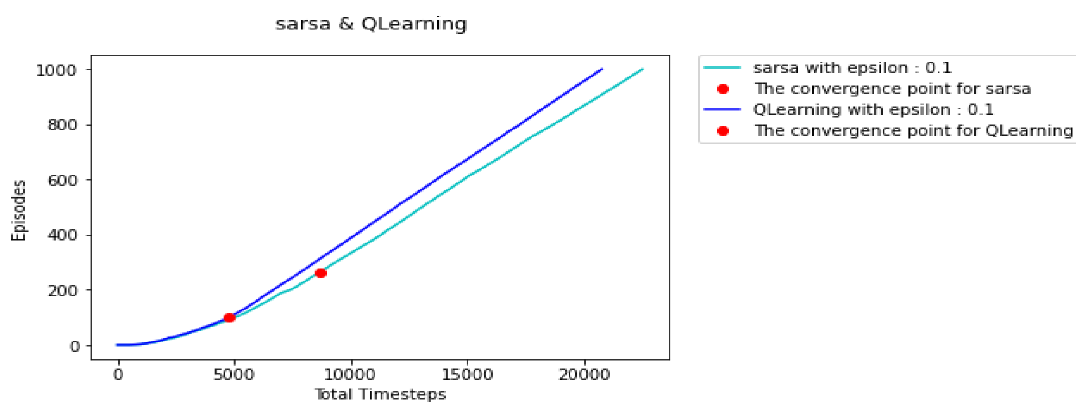
SARSA : The optimal path

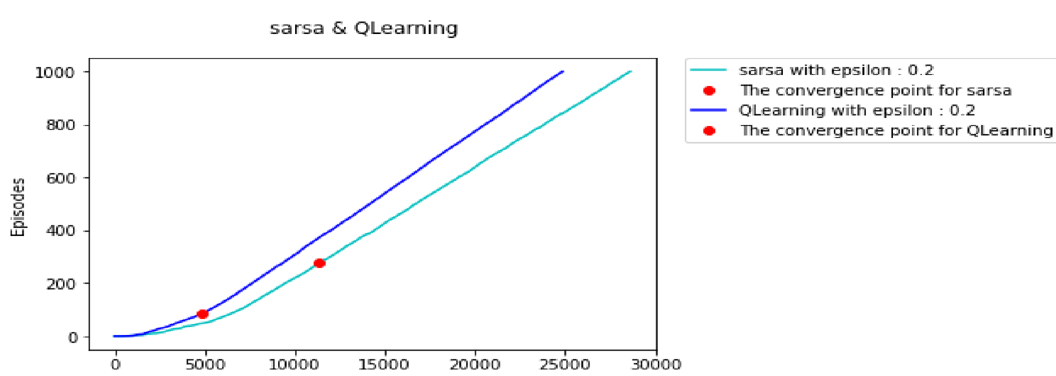


Q-Learning: The optimal path



SARSA vs Q-Learning with different epsilon value:





With king's moves approach:

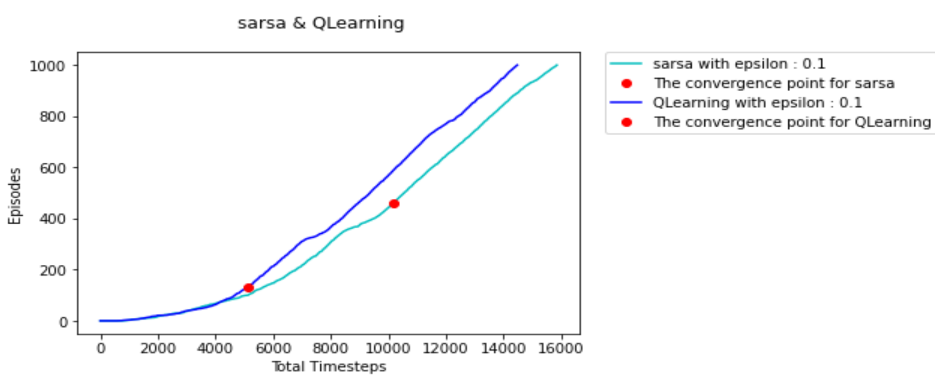
Algorithms	Alpha	Epsilon	The Total Numbers of Reward	The Total Time Steps to converge	Number of episodes to converge
SARSA	0.4	0.1	-7	6163	170
Q-Learning	0.2	0.2	-7	5713	92
SARSA	0.4	0.3	-8	8732	208
Q-Learning	0.4	0.3	-6	4989	111
SARSA	0.4	0.1	-7	10178	461
Q-Learning	0.4	0.1	-6	5127	133

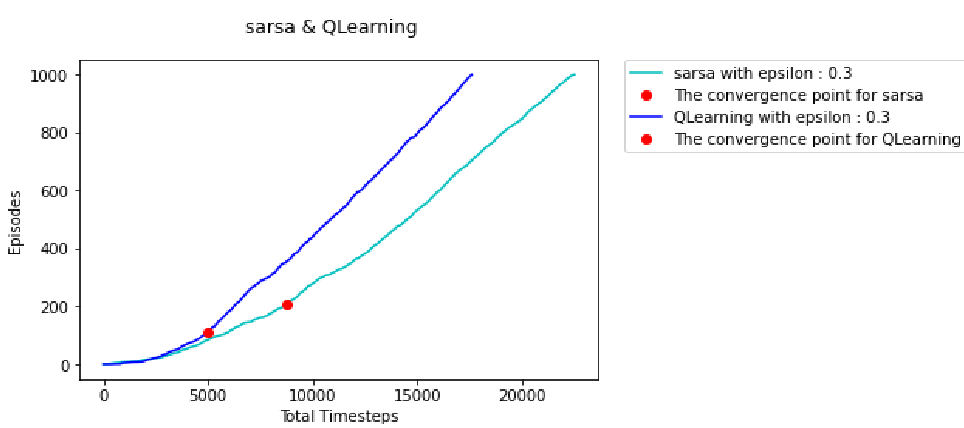
Observations : As we've seen, SARSA and Q-learning find a quicker route thanks to the four new actions. It shows in the graph, as the number of episodes terminated within ~5000 time steps has more than doubled.

Again, Q-learning performs better than SARSA by learning the optimal policy quicker. ε = 0.1 is again better, as expected, though to a greater extent it seems than with only four possible actions.

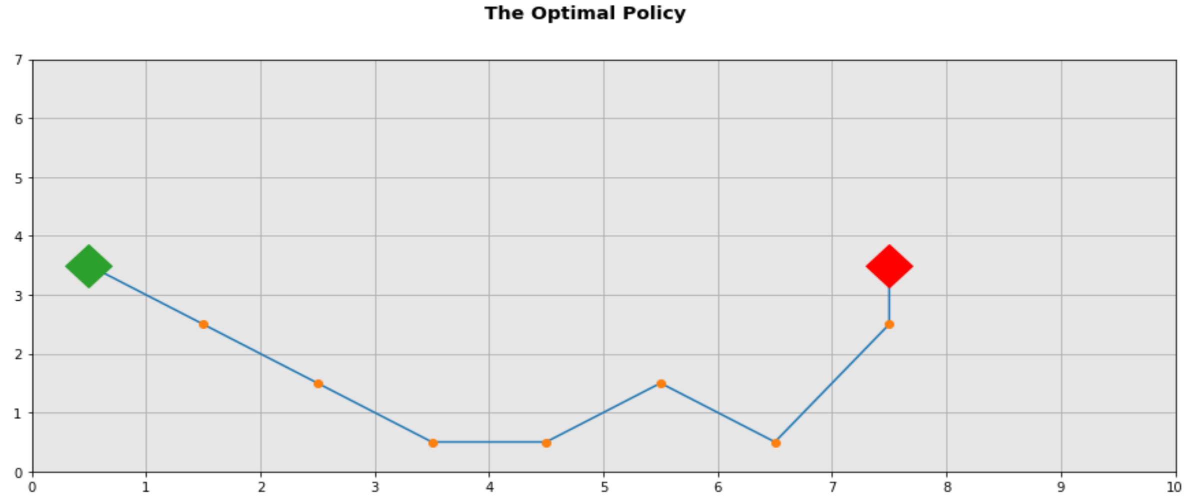
This makes sense since learning the optimal policy with a higher ε takes longer and when choosing a non greedy action, the chance of picking the optimal one is now lower since there are more possible actions.

SARSA vs Q-Learning with different epsilon value:

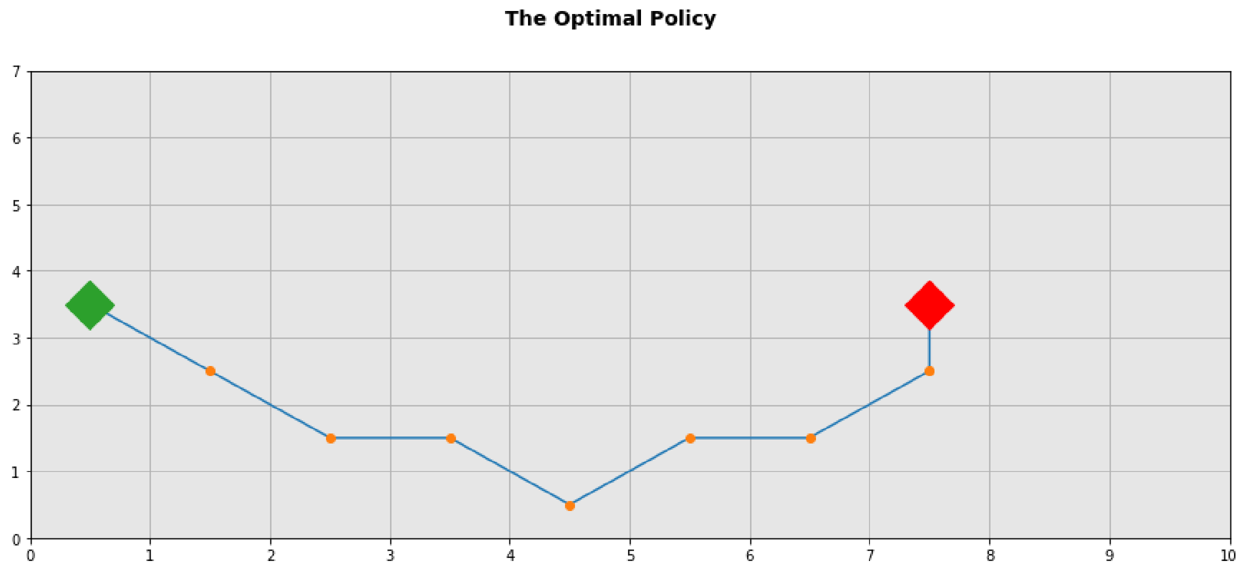




SARSA : The optimal path



Q-Learning: The optimal path



Conclusion:

We've implemented and compared the SARSA and Q-learning algorithm on the Windy Grid World environment. All in all, Q-learning performed better than SARSA with equal ε (except in the stochastic wind case, Q-learning always did better even with varying ε).

It would be interesting to try and compare other on-policy and off-policy algorithms on the Windy Grid world environment, to see if off-policy algorithms always beat on-policy algorithms.

The same could be said about trying and comparing SARSA and Q-learning on other environments, to see if Q-learning always beats SARSA.