CS 747: Programming Assignment 3

Dipesh Tamboli - 170070023

October 23, 2020

1 Task 1 - Implement Windy Gridworld as an episodic MDP

I have created the grid of height, width = (7,10) and I am indexing the grid cell by row, column from the left-top of the grid starting from zero.

I have created a class **GridWorld** with the following functions:

- Functions for getting next state(Different for four and eight actions)
- Separate update functions for Sarsa(0), Q-Learning and expected Sarsa.
- Two separate main functions,
  - one for Sarsa(0) as we need to sample next action before updating the Q values
  - another for Q-Learning and expected Sarsa where we need to sample the action after updating the Q-values.
- A function to get the current status of the agent

Boundary conditions:

- For the case where the wind is taking the agent out of the grid, I am keeping the agent at the boundary of the grid.
- For it, I am first making a move and then adding the wind movement, after than I am considering if the agent has crossed the boundary or not.

Some parameters:

- I have fixed $\alpha = 0.6$ and $\epsilon = 0.05$.
- I have fixed the timesteps to 10k
- I am averaging the graph over 10 seeds

How to run code:
I have created a **tasks.sh** file which can be run and generate all the required plots for the tasks.
all_functions.py has the class and all the required functions.
task_runner.py runs the algorithm and plot the graph.

2 Task 2 - Implementing a Sarsa(0) agent

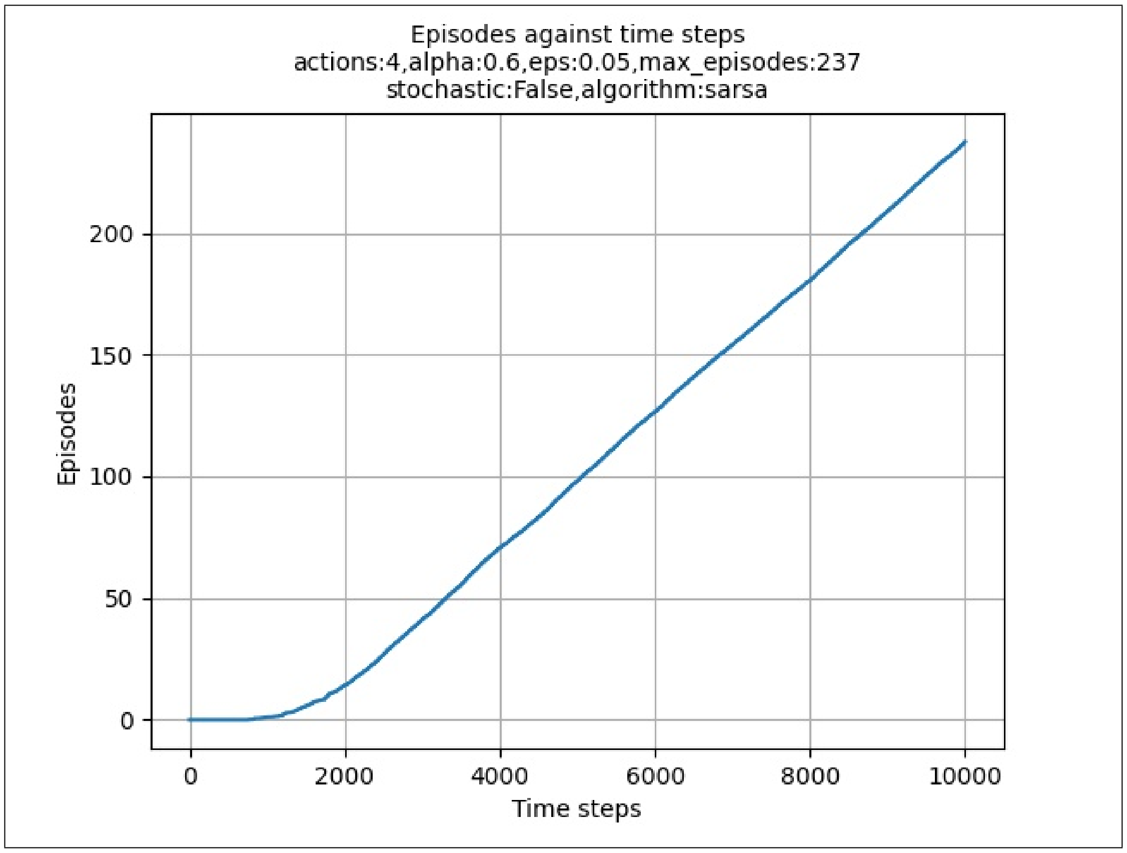


Figure 1: Task 2 plot - baseline

Observations:

- After a few iterations, the slope of the graph is constant which means time steps required for completing each episode is stabilizing to a constant
- This implies that our agent is learning and taking lesser and lesser till it reaches the optimize number
- Minimum actions when the agent will learn properly is 15.

3 Task 3 - Implementing a Sarsa(0) agent with King's moves

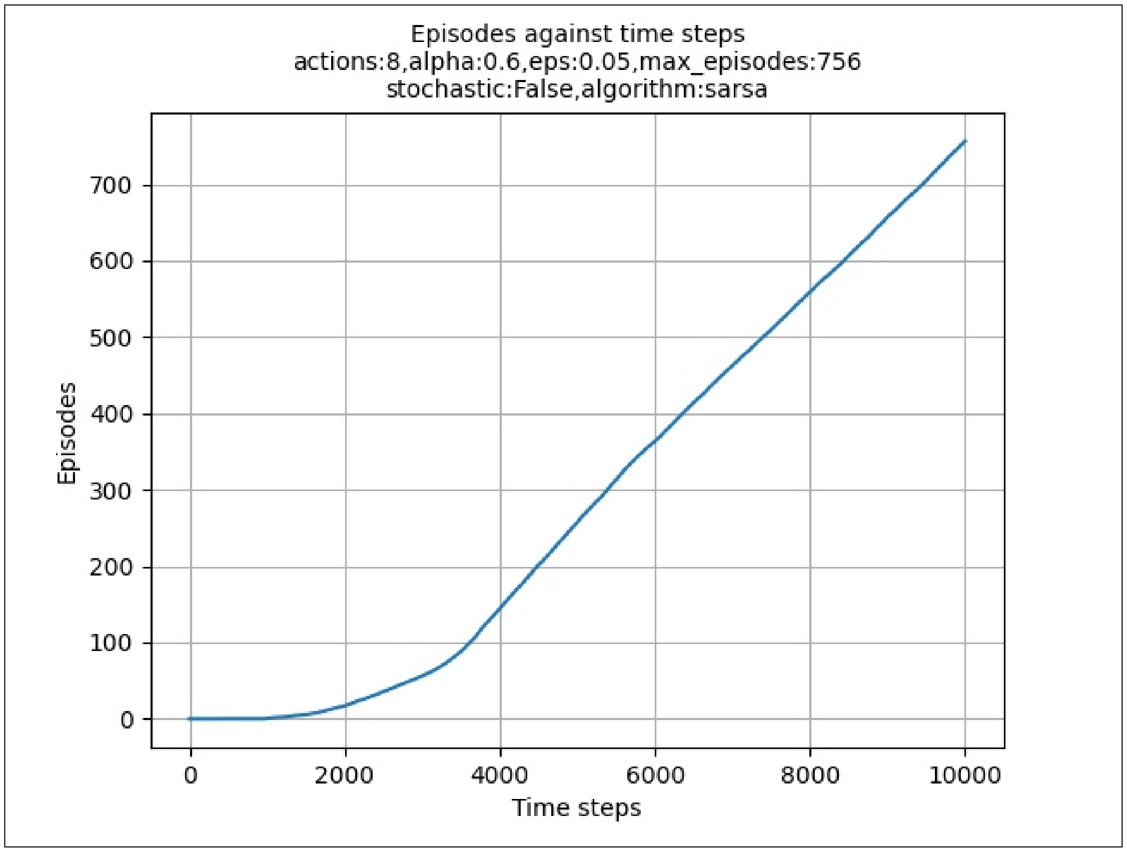


Figure 2: Task 3 plot- King's move

Observations:

- Compared to Sarsa with four actions, here number of episodes completed in 10k time steps is much more
- Minimum actions when the agent will learn properly is 8.
- After a few iterations, the slope of the graph is constant which means time steps required for completing each episode is stabilizing to a constant
- This implies that our agent is learning and taking lesser and lesser till it reaches the optimize number

4 Task 4 - Implementing a Sarsa(0) agent with King's moves and wind stochasticity

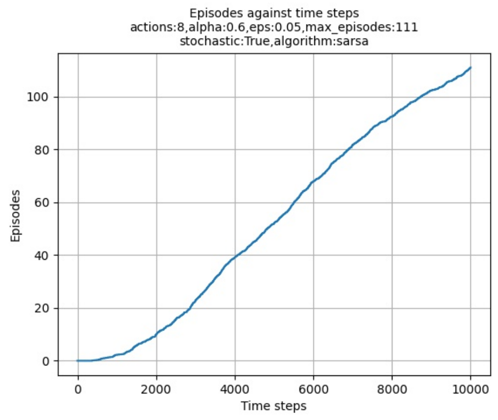


Figure 3: Task 4 plot- stochastic with King's moves

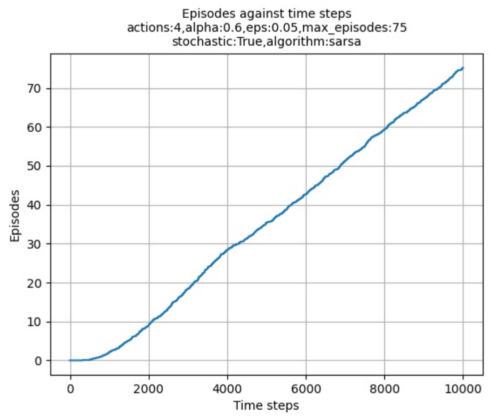


Figure 4: Task 4 plot- stochastic with four moves

Figure 5: Task 4 plots - stochastic

Observations:

- Because of the wind vector giving random actions, it is tough for the agent to generalise the situation and thus we can see that agent is taking more steps than the existing one
- It takes roughly 20k time steps to reach 200 episodes

5 Comparison of Task 2,3 and 4

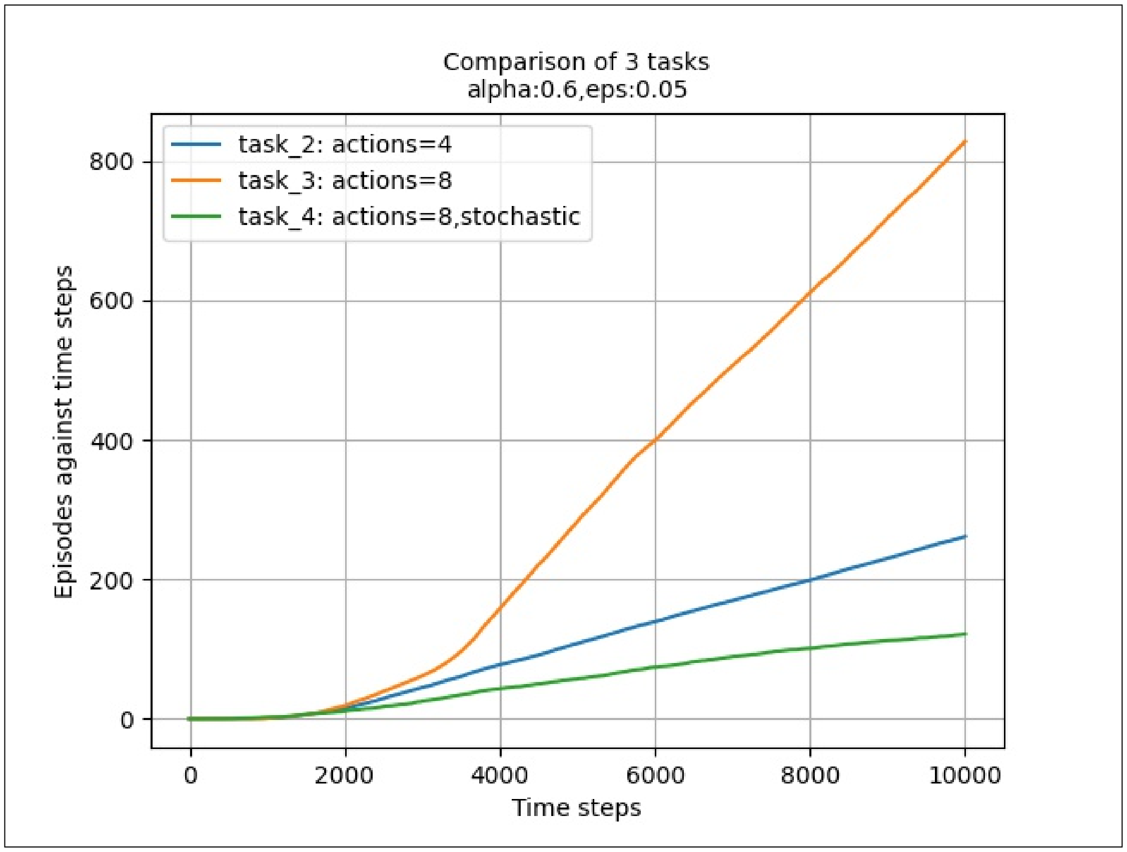


Figure 6: Comparison between task 2,3 and 4

Observations:

- Task 2 vs Task 3: we can see that the increases flexibility in the moves of the agent has reduced the time taken to reach the goal state
- Task 4 vs Task 3: A huge difference when we add randomness in the action caused by the wind. Here, the agent is not able to generalise the wind move for the states.

6 Task 5 - Comparison of the different algorithms

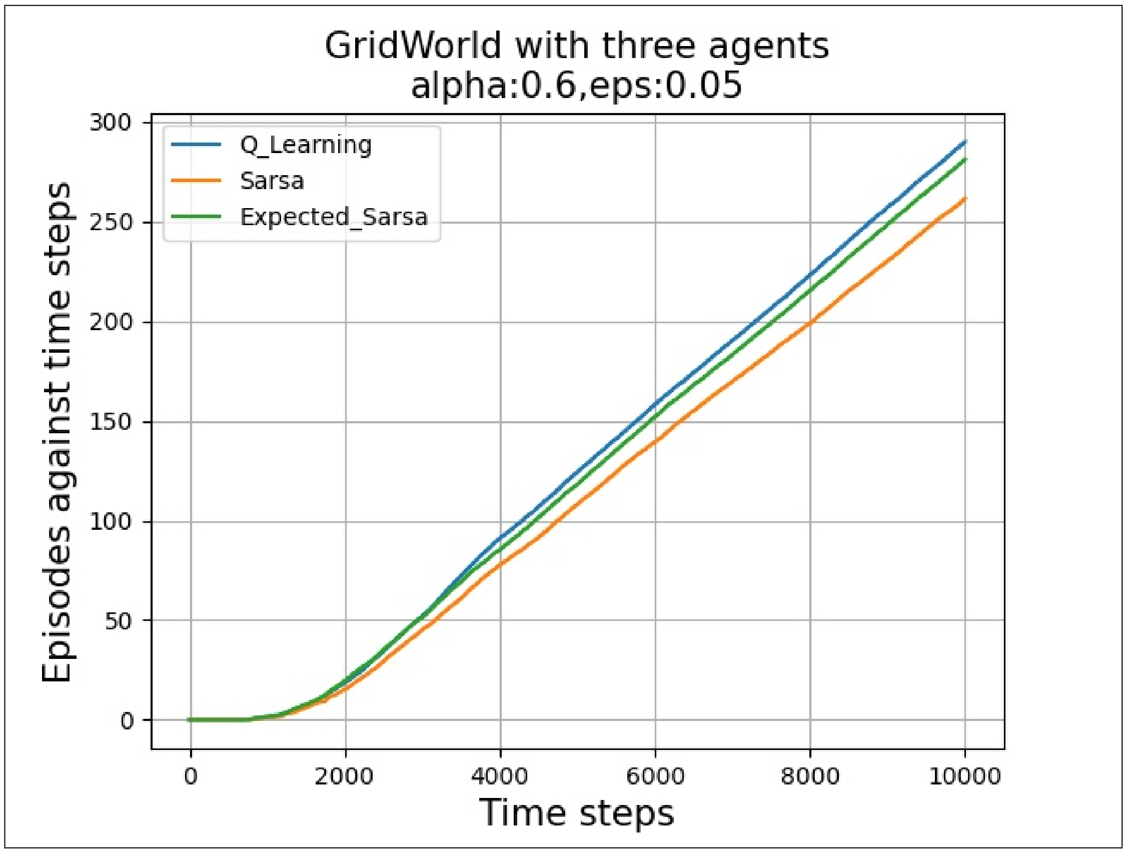
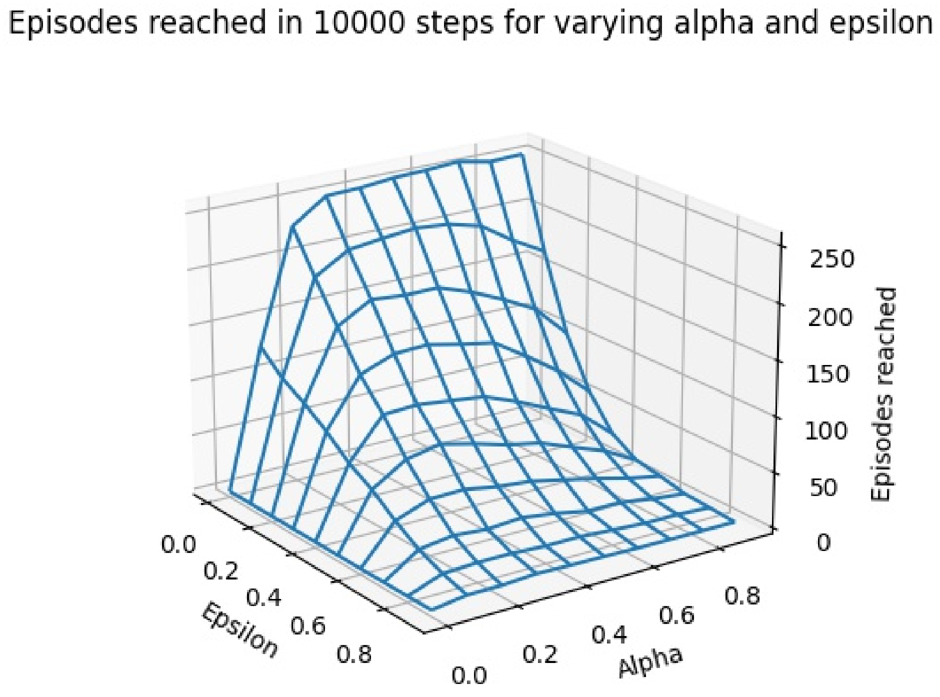
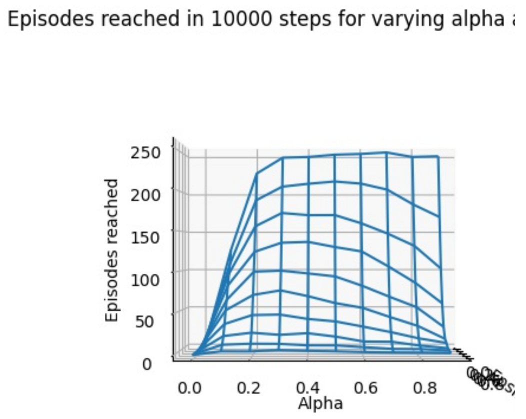


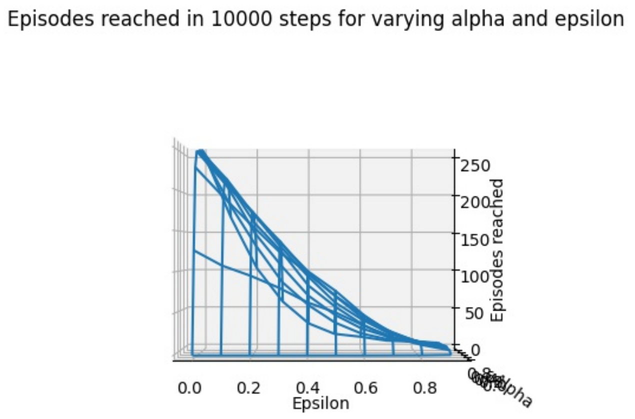
Figure 7: Task 5 plot- Comparison

7 Variation of max episodes reached in 10k steps wrt alphas and epsilon



Episodes vs alpha and epsilon





Episodes vs alpha

Episodes vs epsilon

Figure 8: Max Episodes reached wrt alpha and epsilon