CS747-Assignment 3

Yash Khemchandani - 170050025

November 13, 2020

1 Implementation Details

- `windy_gridworld.py` contains the code for the Windy Gridworld as an MDP. The agent uses the `step` function of this class to get the next state and reward given the action and the `reset` function to reset the environment to the start state. The x-axis and y-axis are shown in the figure.

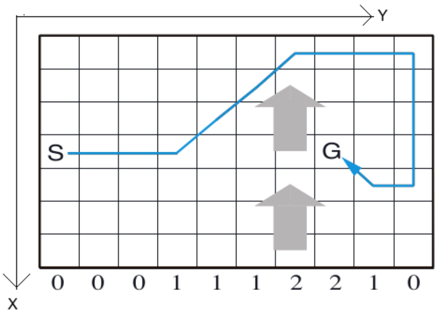

Figure 1: Windy Gridworld

- For all the experiments (baseline, king's moves and king's moves with stochasticity), the environment follows a constrained movement approach, i.e, for all the moves(including the wind direction), the x-coordinate is clamped between 0 and number of rows(7) and y-coordinate is clamped between 0 and number of columns(10).
- If an action takes the current state to the end state (G), then a reward of 0 is given to the agent, otherwise a reward of -1 is given. This ensures that the agent reaches the end state using the shortest path.
- `agent.py` contains the code for the agent, including the 3 control algorithms – SARSA, Expected SARSA and Q Learning. All these control algorithms are implemented using $\lambda = 0$, $\epsilon = 0.1$ and $\alpha = 0.5$. For practical purposes ϵ and α are not annealed with time.
- `generate_plots.py` contains the code for generating a plot given the command line arguments. In addition to plotting the **episodes vs timesteps** plot, it also has the feature to plot the **shortest path** and the **value function** estimated by the agent.

- All the plots for this assignment can be generated by running `bash generate_all_plots.sh`. Each plot is obtained by averaging the statistics over 10 independent runs using 10 different seeds.

2 Plots and Observations

2.1 Episodes vs Timesteps for SARSA

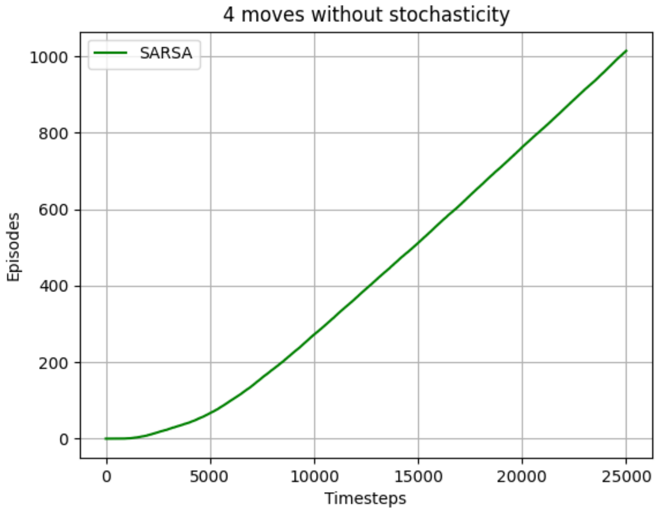

Figure 2: Episodes vs Timesteps for Baseline

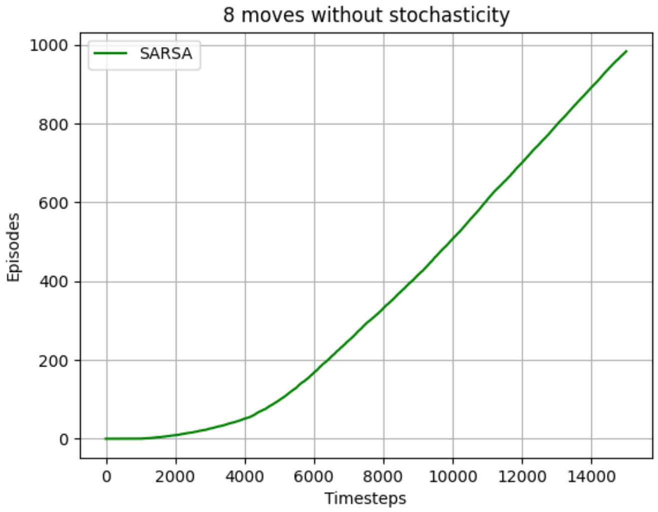

Figure 3: Episodes vs Timesteps for King's Moves

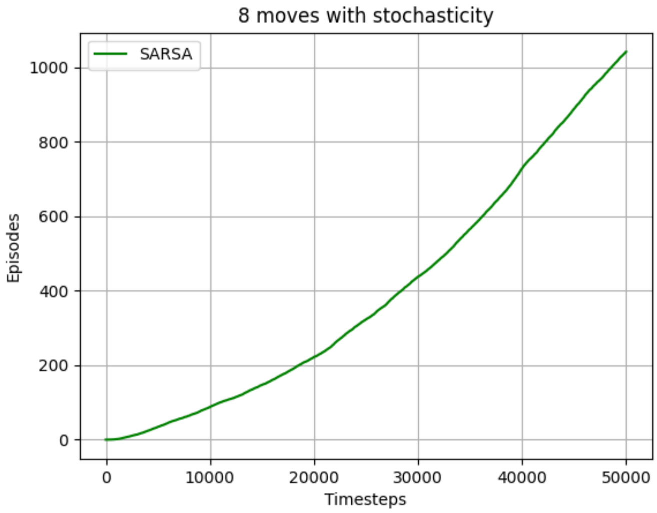

Figure 4: Episodes vs Timesteps for King's Moves with stochasticity

- The number of episodes in the above 3 plots is around 1000 but the timesteps taken by each are different.
- As expected the timesteps taken by using King's moves is the least because of the larger action space and more freedom of movement.
- In case of stochastic wind with King's moves, the number of timesteps taken to reach 1000 episodes is even more than that of the baseline which uses only 4 moves. This is because of the stochastic winds, because of which a deterministic optimum policy is not possible by the agent despite the larger action space.
- We also observe that in the case of stochastic winds, the slope of the plot has not reached a constant value like it does in case of baseline and king's moves without stochastic winds. This can also be attributed to the inherent stochasticity of the underlying MDP.

2.2 Shortest Paths

- The shortest paths for Baseline and King's Moves are shown in Figure 5 and 6 respectively. The purple box represents the start state, orange box represents the end state and the white boxes represent the path taken by the agent
- For the baseline, the length of the shortest path is 15 with the optimal policy : ['Right' 'Right' 'Right' 'Right' 'Right' 'Right' 'Right' 'Right' 'Right' 'Down' 'Down' 'Down' 'Down' 'Left' 'Left']
- For King's moves without stochasticity, the length of the shortest path is 7 with the optimal policy: ['Down-Right' 'Right' 'Right' 'Down-Right' 'Down-Right' 'Down-Right' 'Down-Right']

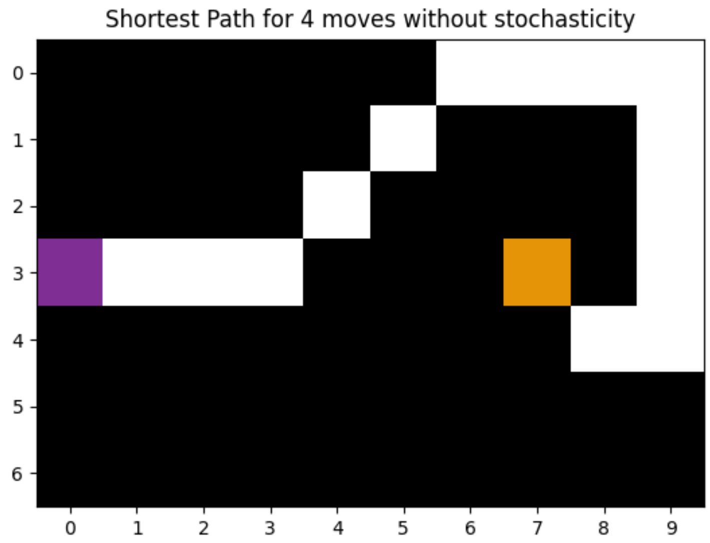

Figure 5: Shortest path for Baseline

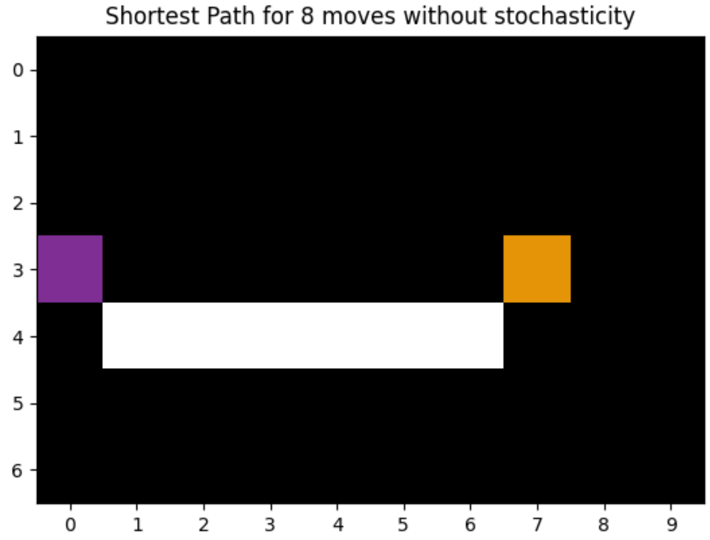

Figure 6: Shortest path for King's Moves

2.3 Value functions

- The plots for the value functions as estimated by the agent for Baseline, King's Moves and King's Moves with stochastic winds are shown in Figure 7,8 and 9 respectively. These are the mean

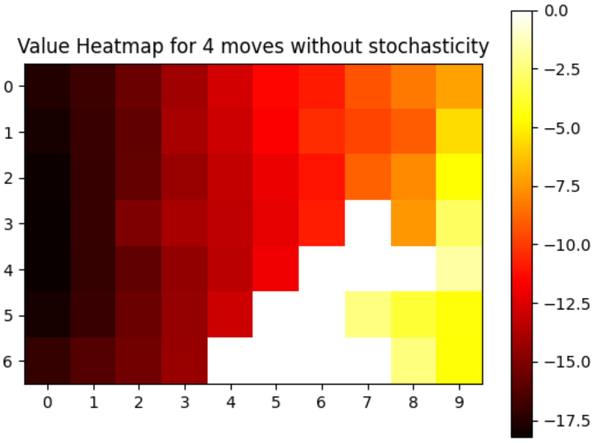

Figure 7: Value function for Baseline

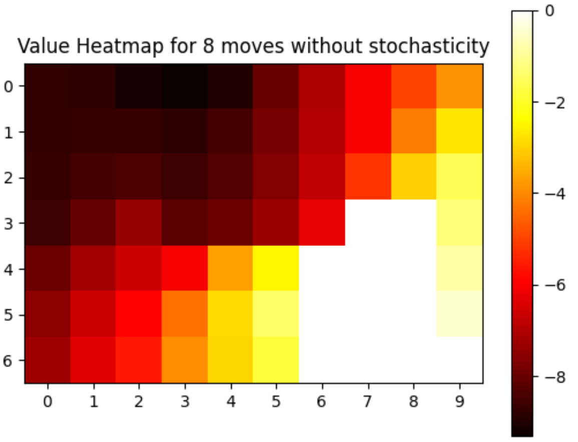

Figure 8: Value function for King's Moves

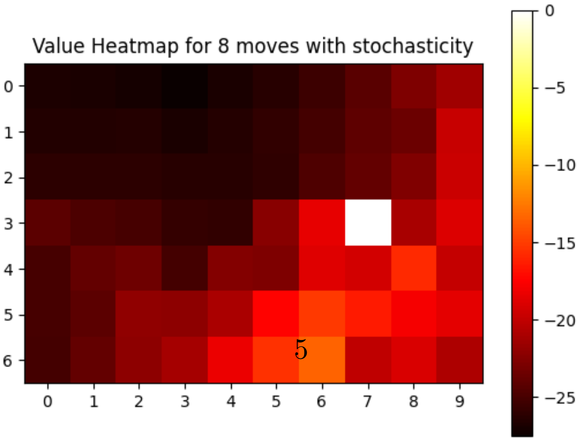

Figure 9: Value function for King's Moves with stochasticity

estimates over 10 independent runs

- The value for the cells below the goal is more than those above because when the agent is in the cells below, it has an added advantage of the wind pushing it towards the goal while when the agent is in the cells above, the wind pushes it away from the goal.
- The value for the cells in case of King's Moves is more than that of the Baseline because in the former case, the agent can reach the goal in less steps from a cell because of the increased action space, and hence accumulate larger reward.
- The value for the cells in case of King's Moves with stochasticity is even less than that of the Baseline because of the stochastic winds resulting in an unpredictability of the moves.

2.4 Combined Plots for Episodes vs Timesteps

- Figures 10, 11 and 12 show the combined plots for Baseline, King's Moves and King's Moves with stochasticity.
- As we can observe, Expected SARSA and Q Learning perform much better than SARSA since they reach larger number of episodes in the same timesteps.
- Expected SARSA and Q Learning perform similarly in cases without stochasticity, but for the case of stochastic winds, Q Learning performs much better than the other two algorithms.
- For the case of stochastic winds, the slopes for all the 3 plots is noisy and hasn't stabilized like the previous two cases.

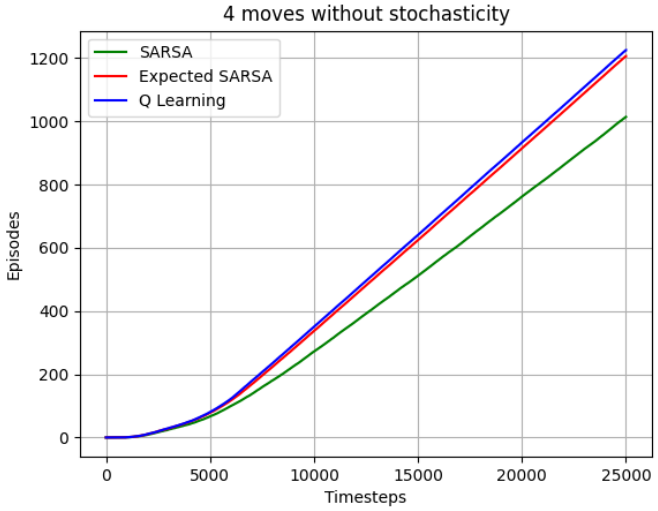

Figure 10: Episodes vs Timesteps for Baseline

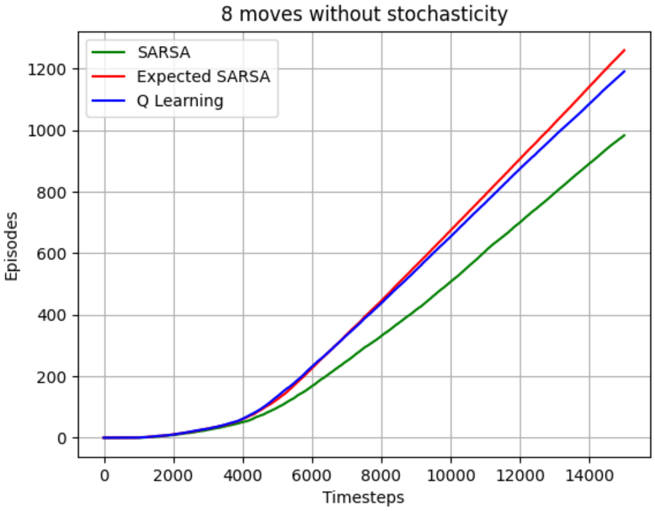

Figure 11: Episodes vs Timesteps for King's Moves

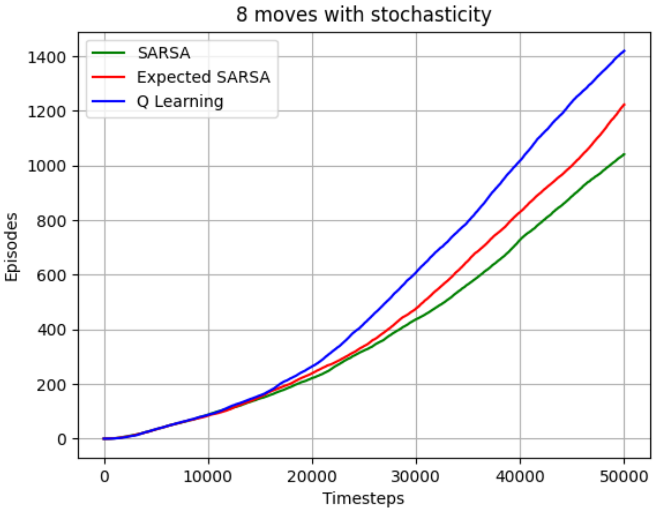

Figure 12: Episodes vs Timesteps for King's Moves with stochasticity