CS6700 - Reinforcement Learning
Programming Assignment 1

February 22, 2022

1 Environment Description

This exercise aims to familiarize you with various learning control algorithms. The goal is to solve several variants of the Grid World problem (a sample world is shown in Figure 1). This is a typical grid world, with 4 deterministic actions ('up', 'down', 'left', 'right'). The actions might result in movement in the direction as intended with a probability of p, and equally likely to result in movement in other directions with a combined probability of $1-p$. For example, if the selected action is 'up', it will transition to the cell one above your current position with probability p and to one of the other neighboring cells with probability $(1-p)/3$. Transitions that take you off the grid will not result in any change. The dimensions of the grid are 10×10.

There is also a gentle wind blowing that will push you one **additional** cell to the east (right), regardless of the effect of the action you took, with a probability of 0.4. Some details regarding the world are enlisted below.

- ***Start state***: The agent starts from this state.
- ***Goal state***: The goal is to reach one of these states. There are 3 goal states in total.
- ***Obstructed state***: These are walls that prevent entry to the respective cells. Transition to these states will not result in any change.
- ***Bad state:*** Entry into these states will incur a higher penalty than a normal state.
- ***Restart state:*** Entry into these states will incur a very high penalty and will cause agent to teleport to the start state without episode ending
- ***Normal state:*** Entry into these states will incur a small penalty.
- ***Rewards:*** -1 for transition to normal states, -100 for restart states, -6 for bad states, +10 for goal states.

Additional Information

The code for the environment can be found here. The `env.step()` function takes as arguments, the current state and action and returns the reward and next state. The appropriate termination conditions have to be specified by the student in the code (as explained in the line above). `env.reset()` resets the environment. Cell 1 contains the environment class, cell 2 contains the environment instantiation, and cell 3 lists some environment variables. For each experiment, the start state is fixed and does not change. Different experiments may have different start states. The goal of the agent is to maximize the expected reward.

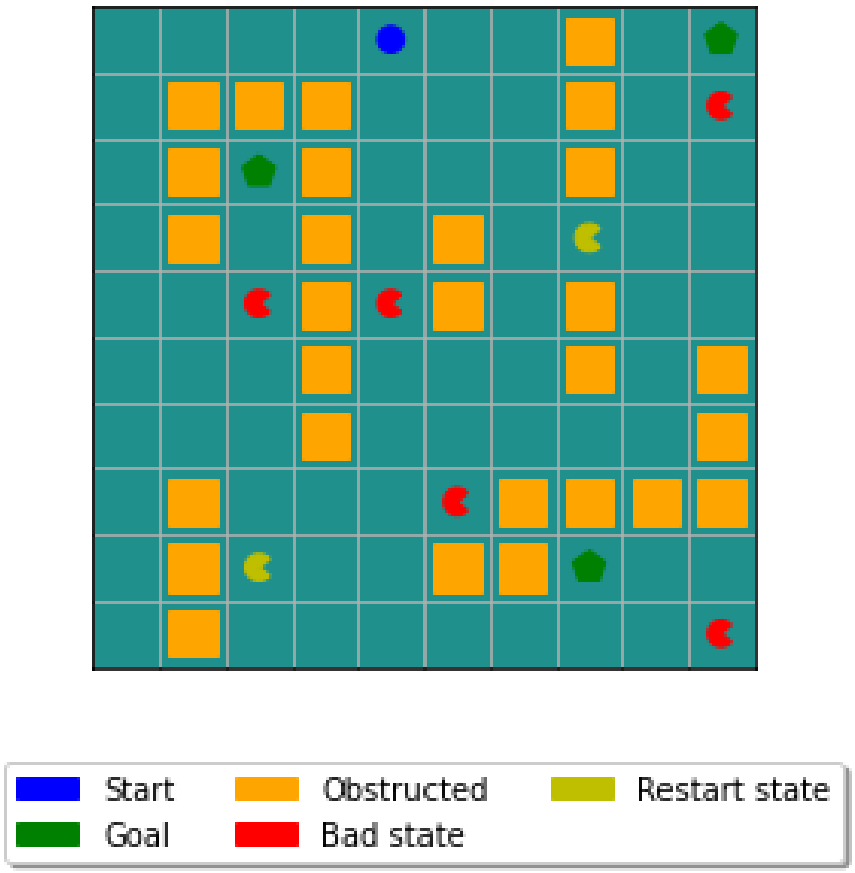


Figure 1: An example grid world with start position $(0, 4)$

2 Tasks

Note - A description of the total number of plots required is provided in Section 3. 'Best hyperparameter choice' mentioned in this section refers to the best option of all the variations that you have experimented with. This does NOT necessarily have to be a brute-force optimal set of hyperparameters for the problem.

- Implement SARSA and Q-learning using both ϵ-greedy and softmax policies.
- For each algorithm, run experiments with wind=False and wind=True, two different start states: $(0, 4)$, $(3, 6)$ and three values of p (1.0, 0.7, and 0.35), making it 12 different configurations in total.
- Tune the hyperparameters enlisted below (explore at least 4 values for each) and plot the total reward curves for all variations for each hyperparameter, fixing the others. We shall call these *'Hyperparameter Plots'*.
  1. Exploration strategy ϵ (for ϵ-greedy)
  2. Temperature factor β (for softmax)
  3. Learning rate α
  4. Discount factor γ

- For each of the 12 configurations, choose the best set of hyperparameters and plot the following (we shall call these *'Best Plots'*).

  1. Reward curves and the No. of steps to reach the goal in each episode (during the training phase with the best hyperparameter choice)
  2. Image of Grid World highlighting states visited and actions taken (using final policy learnt), i.e., state and action trajectories.
  3. Heatmap of Grid World with Q values and optimal actions (of final policy learnt)

  Provide a detailed report - For each of the 12 configurations, provide a written description of the policy learnt, explaining the behavior of the agent through different values of ϵ, β, α and γ

3 Plot Summary

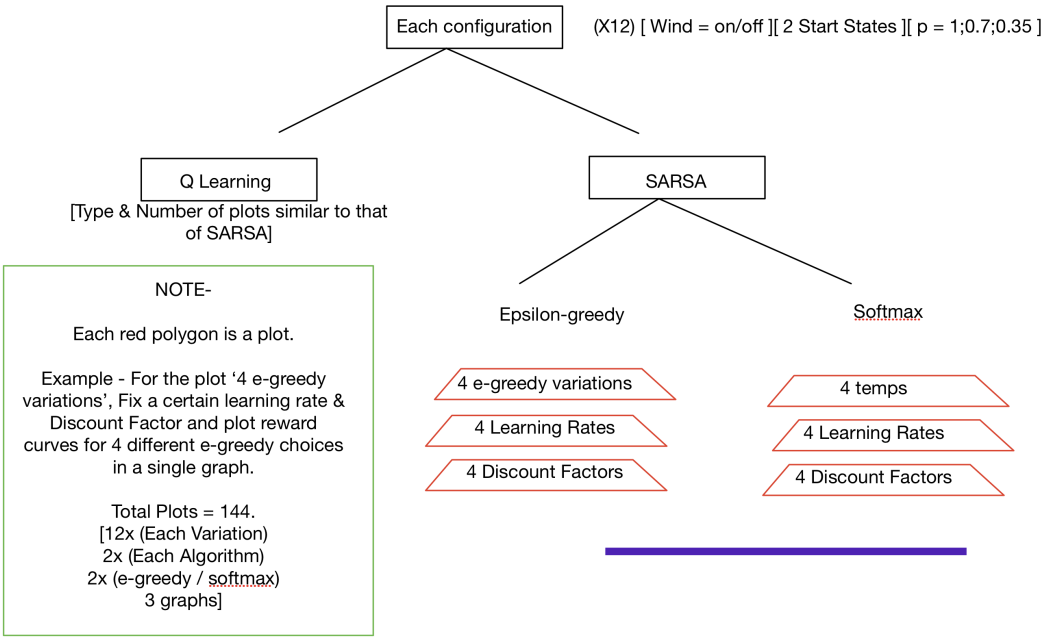


Figure 2: Summary of *'Hyperparameter plots'*

The final report is expected to have a total of 144 *'Hyperparameter Plots'* and 36 *'Best Plots'*. Please group them accordingly to match your inferences.

4 Submission Instructions

You are required to submit both your report and your code (details will be announced on Moodle). Please submit in **teams of two**. One submission per team will suffice. The due date for this programming assignment is **11:59 pm on Tuesday, March 8th**.

RLPA1

Sathvik Joel cs19b025 & R Rohith ee19b114

March 2022

Introduction

In this assignment 12 configurations were analyzed for the best algorithms and corresponding Hyper parameters. For every configuration the following process is followed to tune the Hyper parameters.

Wandb Analysis

For each configuration a sweep is set up in Wandb that runs 20 different Hyper parameter combinations and logs the following information for each run in each sweep.

- **Reward**: Reward for each episode. Essentially the reward curve
- **Steps**: Number of steps taken in each episode. The episode ends if it takes more than 100 steps.
- **Average Reward (Train)**: The average of all the rewards across training
- **Average Steps (Train)** : The average steps taken across training
- **Average Steps (Test)** : Once the agent is trained the agent is made to explore the environment with trained policy for 1000 episodes. The rewards for all these runs are averaged and logged. This metric was extensively used in selecting the best Hyper parameters
- **Average Steps (Test)** : Similar to above, the steps taken in episode are averaged and logged.

In this pdf, we have only shown best Wandb runs to make it look better, but all the experiments could be found using the link better.

All the experiments wehave performed could be found here in Wandb Dashboard here.

Plots

For each configuration we selected the best Hyper parameters from the above experiments. To also check our HP's we take three other values for each of the parameters and plot them. Each configuration involves three sections corresponding to varations we want to plot

1. **Greed Variations** : In this subsection, we fix Discount factor (γ), Step size (α) and we vary the policy greed, that is epislon in case of epsilon greedy or beta in case of softmax. We also vary the algorithm in these plots. All this can be inferred from the titles of the plots
2. **lr Variations**: Same idea as above except that, α Step size is varied.
3. **Gamma Variations** : Similar idea, excpet that, γ is varied.

Best Plots

This section involves plotting for the best hyper parameters

1. Reward Curve and Steps taken Curves
2. Image of Grid World highlighting states visited and actions taken (using final policy learnt), i.e., state and action trajectories. We ran it two times.
3. Heatmap of Grid World with Q values and optimal actions (of final policy learnt)

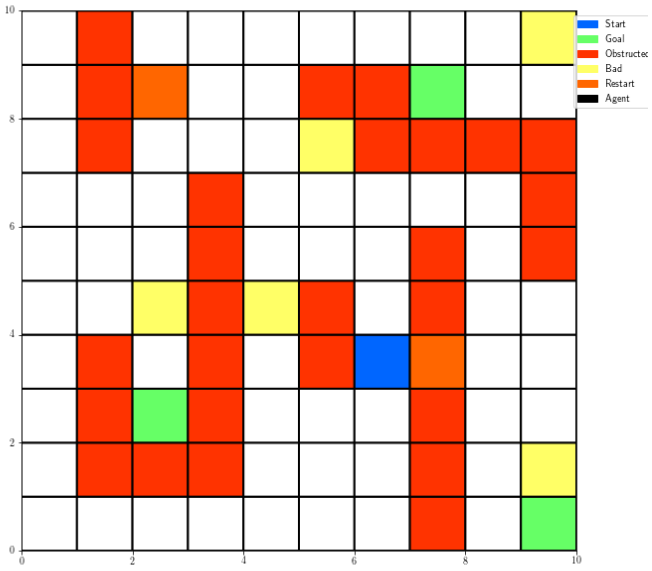


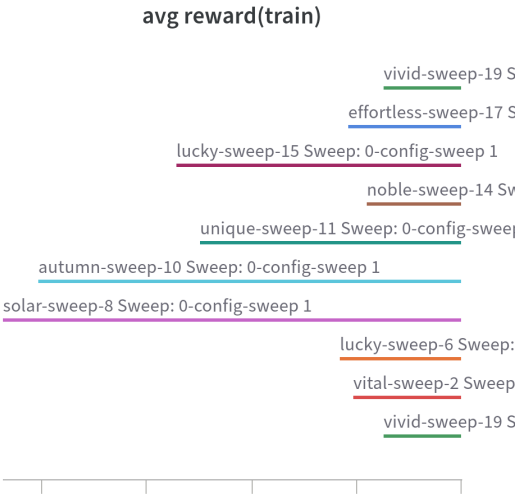
Figure 1: The Environment Visualization

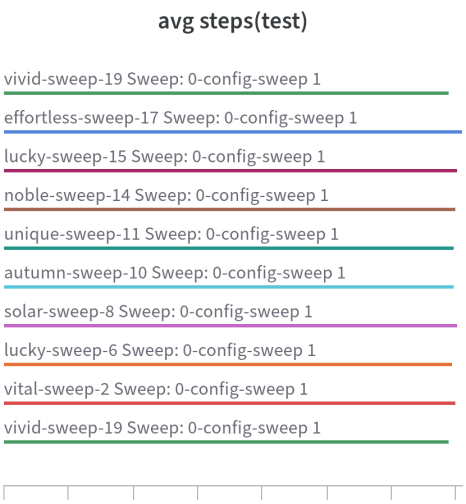
Configuration 0

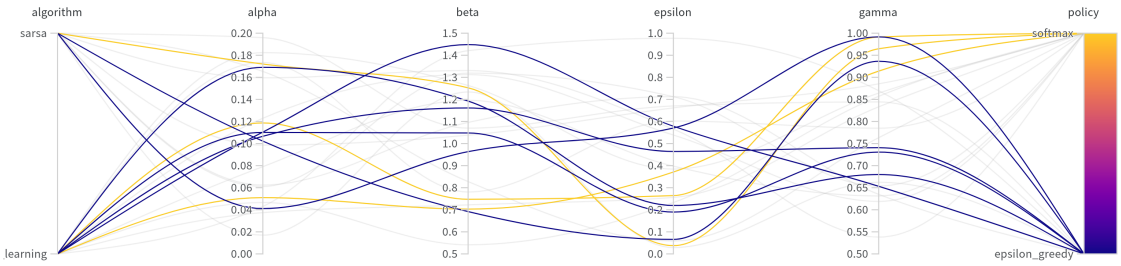
Configuration parameters

Wind = **True**, Start State = **[0,4]**, p = **1.0**

Wanb Analysis





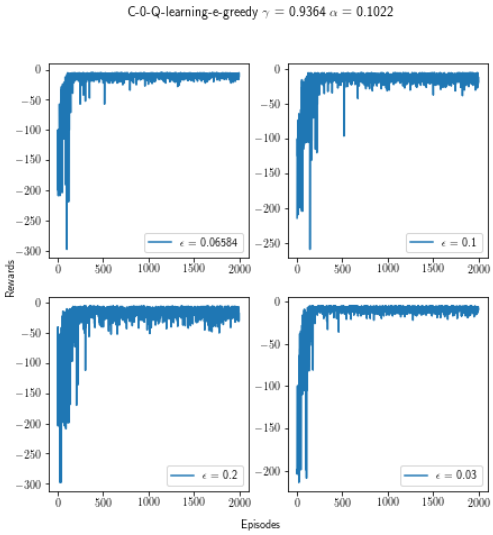


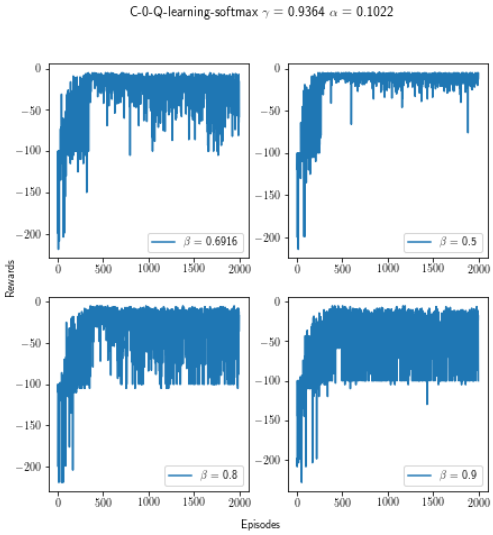
Hyper-parameter Plots

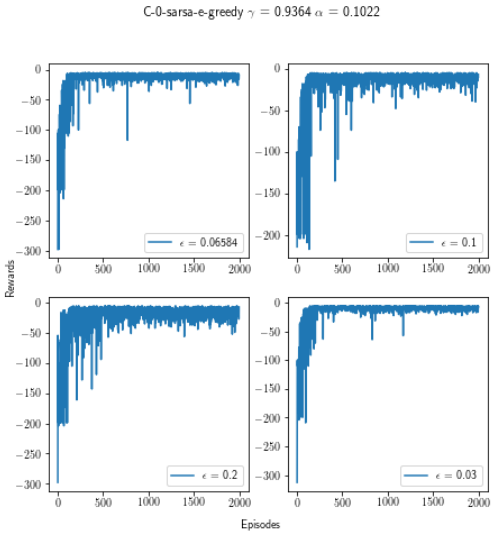
Fixing the best Hyper parameters found in the above section, other values are tried to in order to verify the observations.

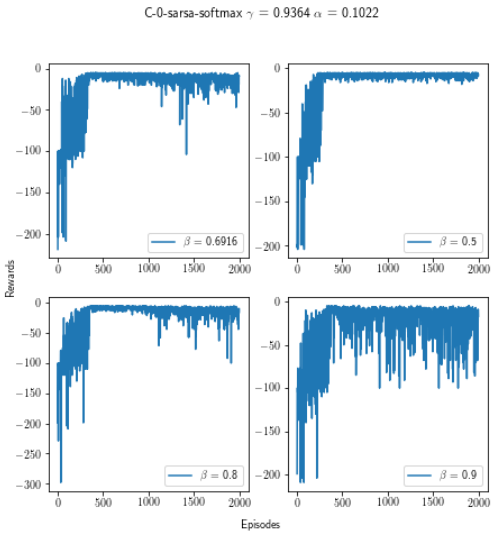
Policy Greed Variations

In this for each Algorithm and policy, α and γ are fixed and the policy greed i.e., ϵ or β , depending on the policy, is varied.





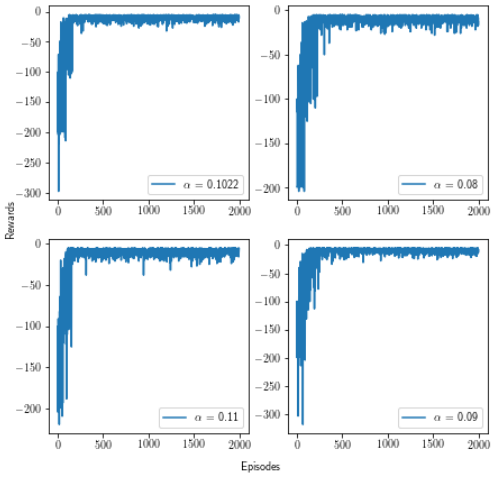




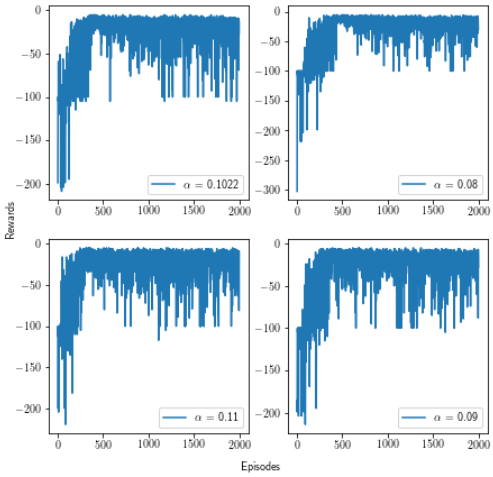
Learning Rate Variations

In this for each Algorithm and policy, policy greed and γ are fixed and Learning Rate α is varied.

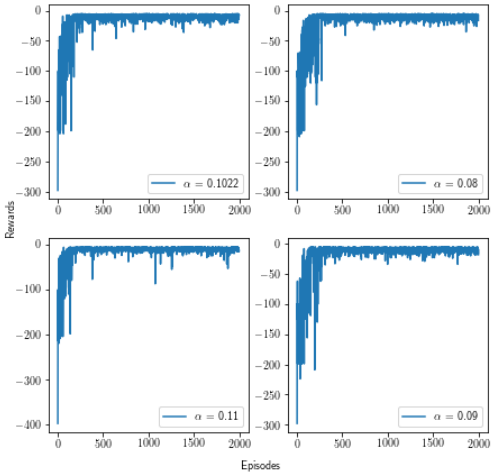
C-0-Q-learning-e-greedy $\gamma = 0.9364$ $/\epsilon = 0.06584$



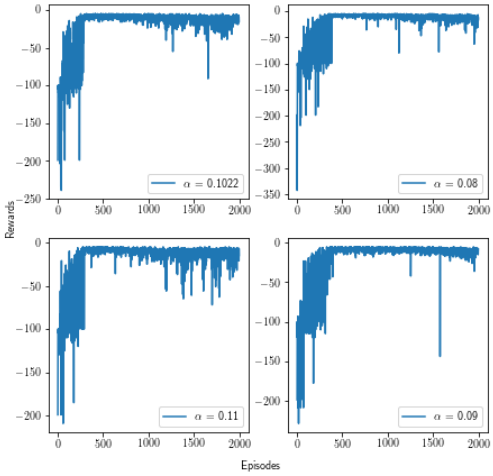
C-0-Q-learning-softmax $\gamma = 0.9364$ $\beta = 0.6916$



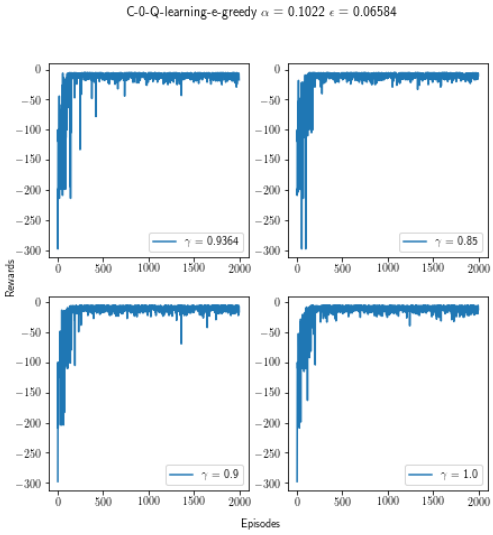
C-0-sarsa-e-greedy $\gamma = 0.9364$ $/\epsilon = 0.06584$

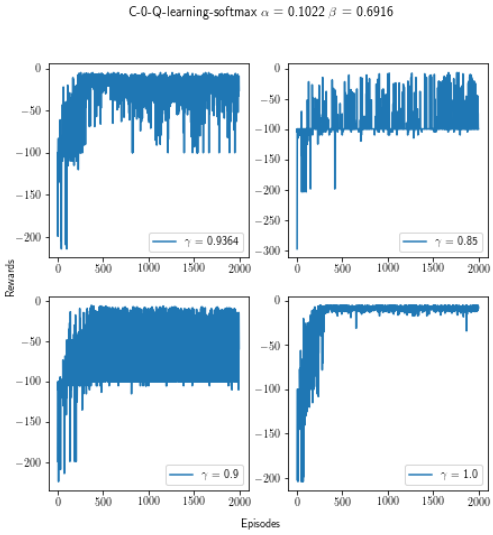


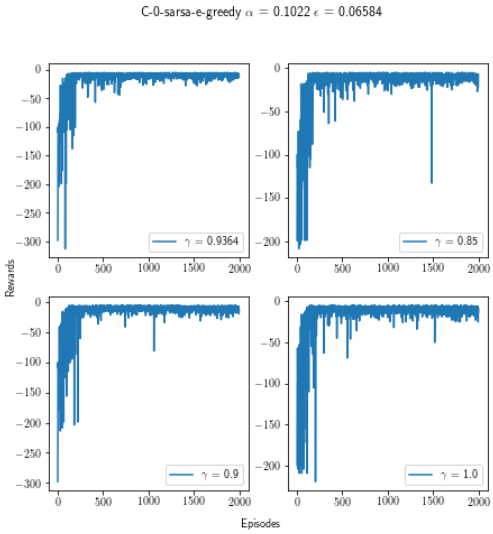
C-0-sarsa-softmax $\gamma = 0.9364$ $\beta = 0.6916$

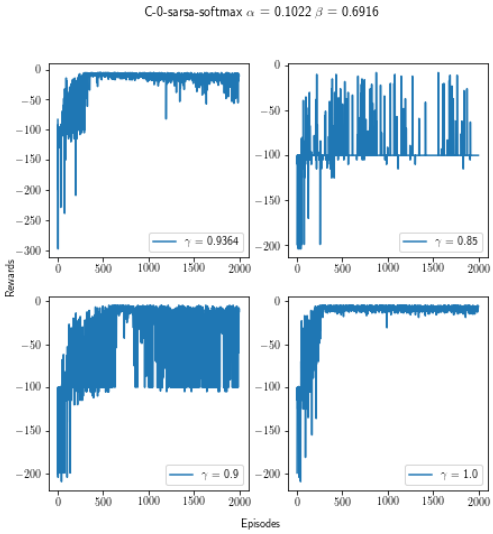


Discount Rate Variations





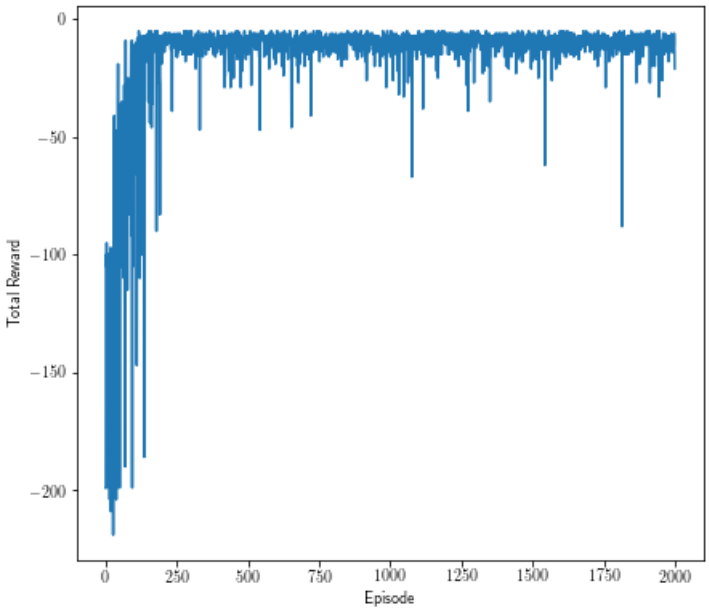


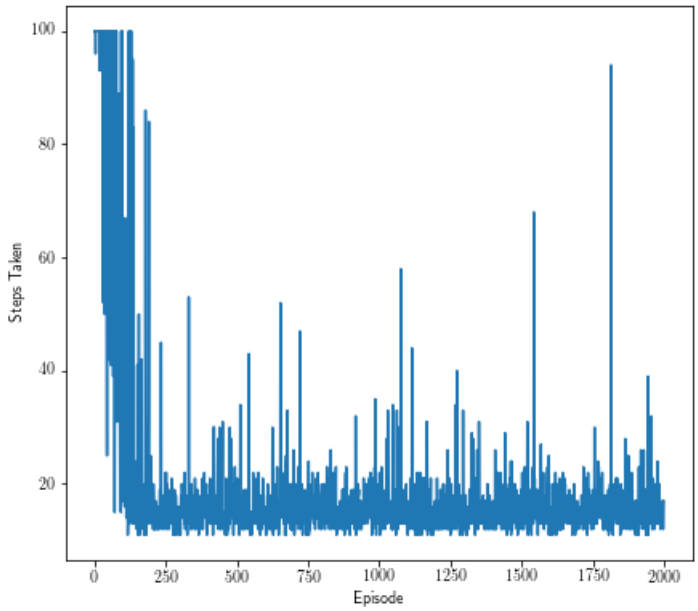


Best Plots

The plots for best set of Hyper Parameters found using Wandb analysis and also supported by variations in above section.

Reward Curve and no of steps to reach goal





Best plots for:

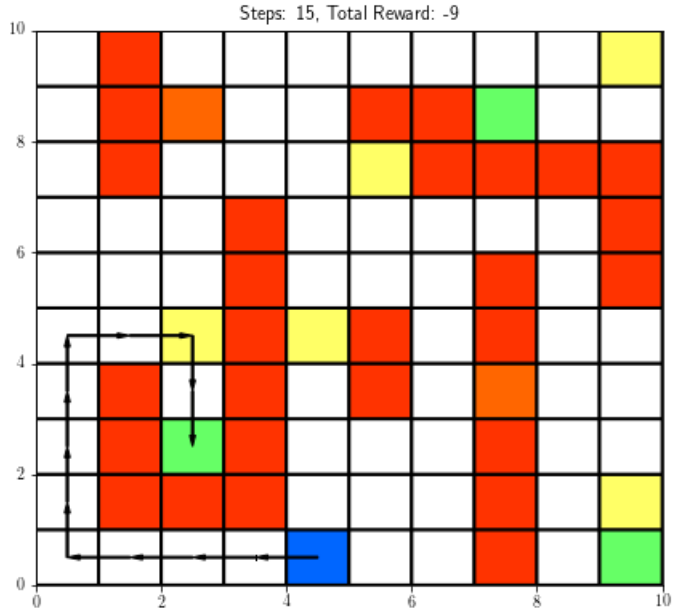
- Algorithm - Sarsa
- Policy - e-greedy
- Epsilon - 0.06584
- Alpha - 0.1022
- Gamma - 0.9364

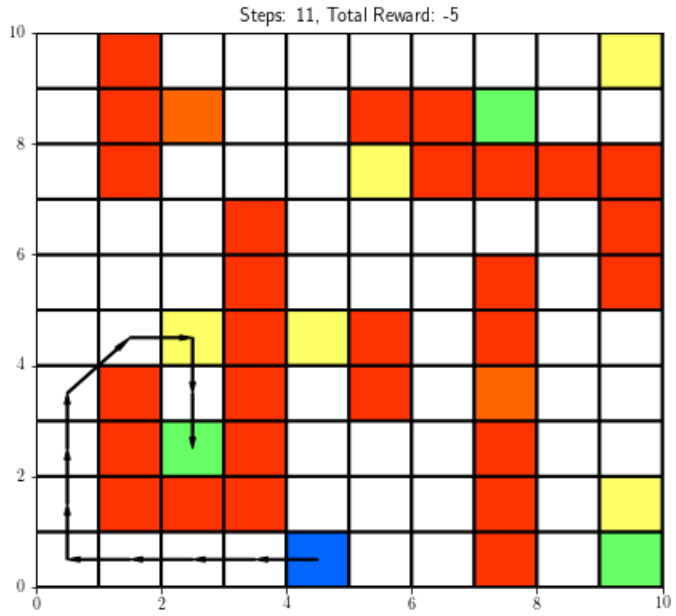
Final Learned Policy

For this configuration, we have wind as the only stochasticity for the transitions within states. That adds a bit of variance in the rewards obtained and in the no of steps taken to reach the goal.

For example in the second run shown in the next page, the wind helped the agent move along a diagonal by-passing one square and increasing the reward. Well, in some other cases wind could act while the agent is moving along the last row of the grid(reducing the reward and increasing the steps in this case).

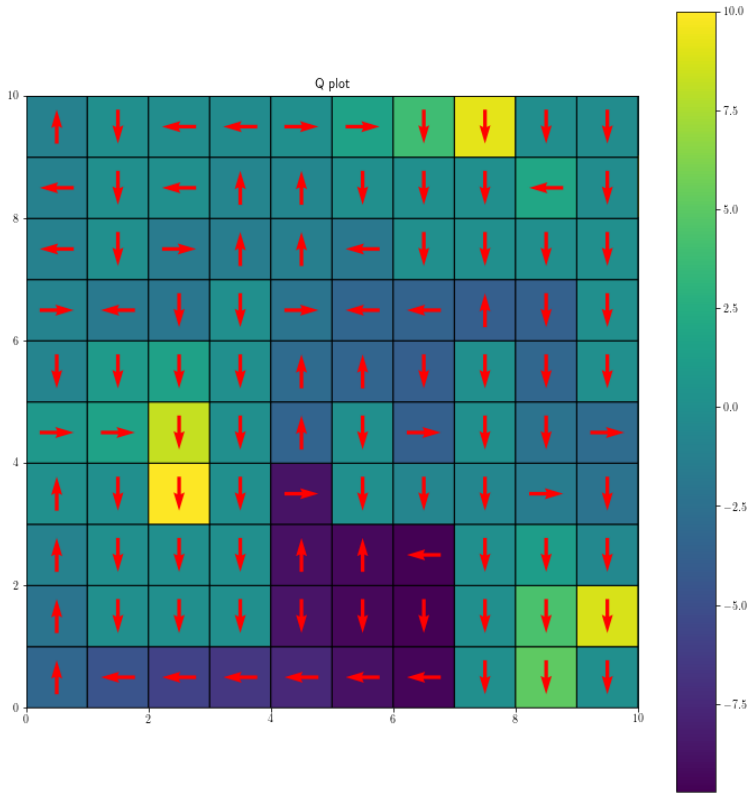
The trained agent is made to explore the environment for 2 runs. The visited states and actions are plotted in the next page.


Steps: 15, Total Reward: -9
0
2
4
6
8
10


Steps: 11, Total Reward: -5
0
2
4
6
8
10

HeatMap

Heatmap of Grid World with Q values and optimal actions (of final policy learnt)

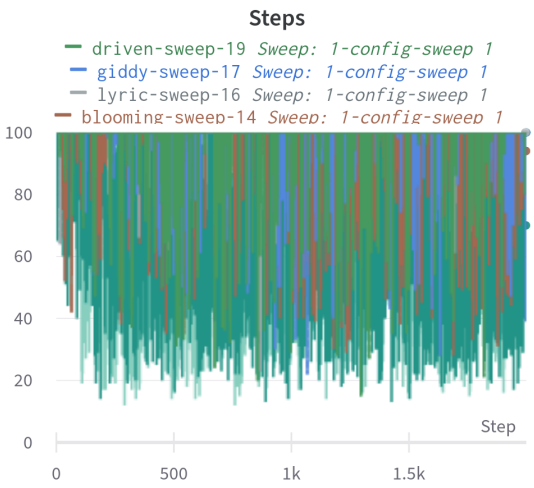


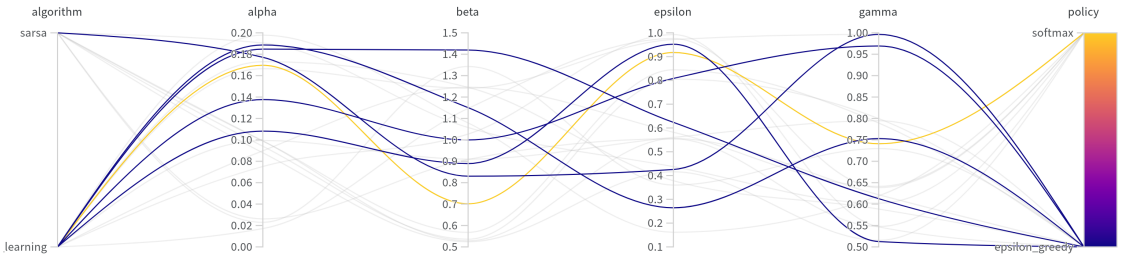
Configuration 1

Configuration parameters

Wind = **True**, Start State = **[0,4]**, p = **0.7**

Wandb Analysis (Best 6 Runs)



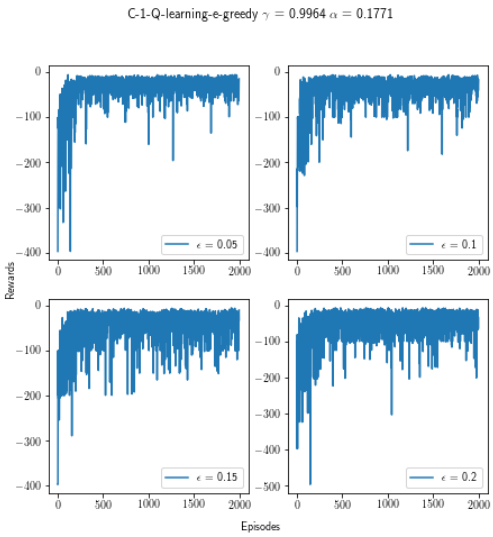


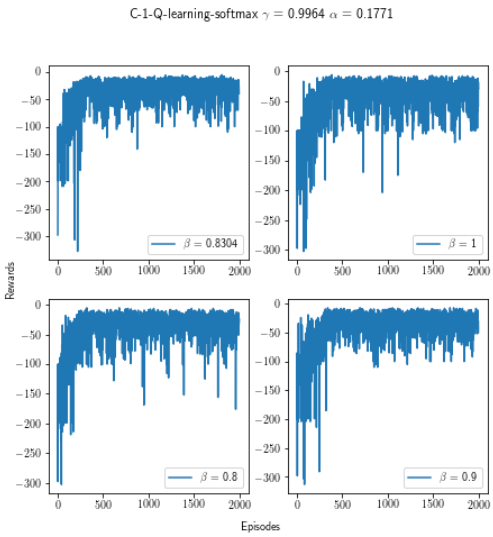
Hyper-parameter Plots

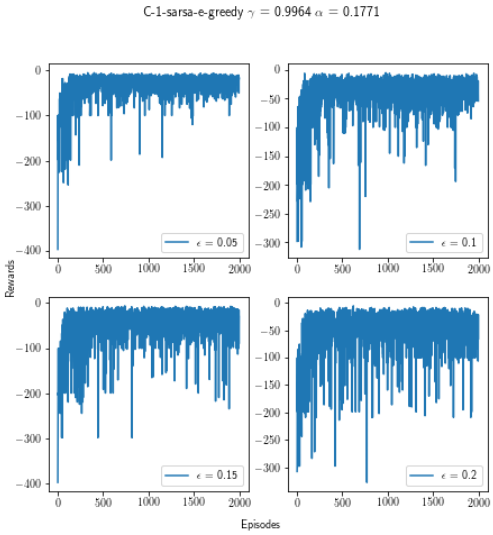
Fixing the best Hyper parameters found in the above section, other values are tried to in order to verify the observations.

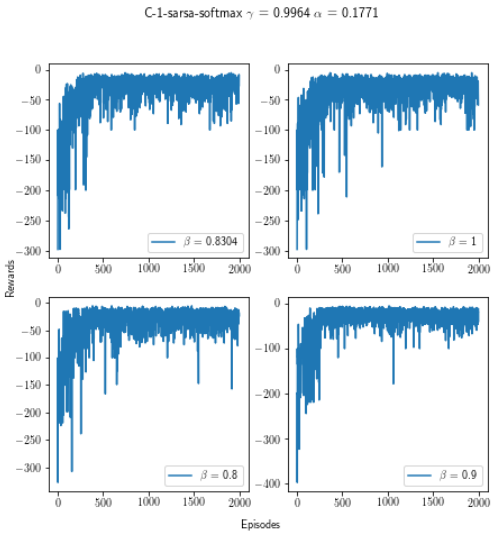
Policy Greed Variations

In this for each Algorithm and policy, α and γ are fixed and the policy greed i.e., ϵ or β , depending on the policy, is varied.





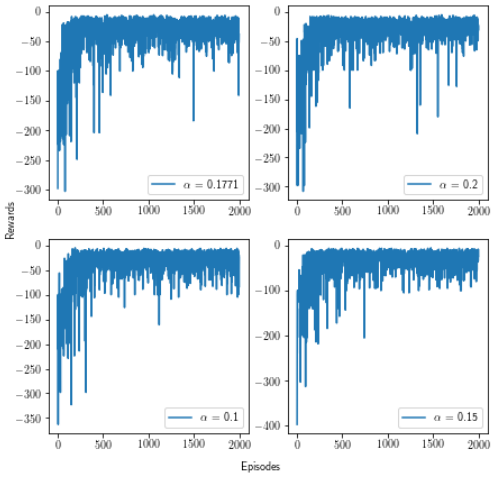




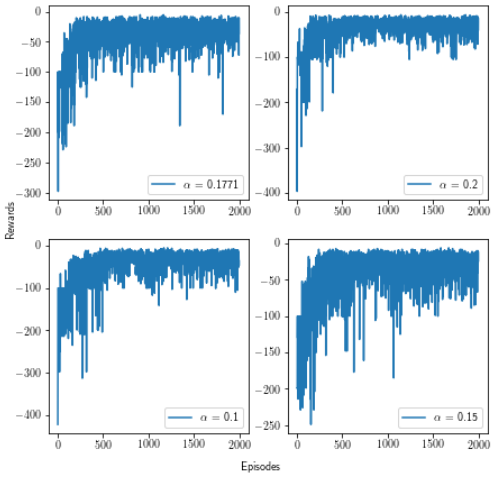
Learning Rate Variations

In this for each Algorithm and policy, policy greed and γ are fixed and Learning Rate α is varied.

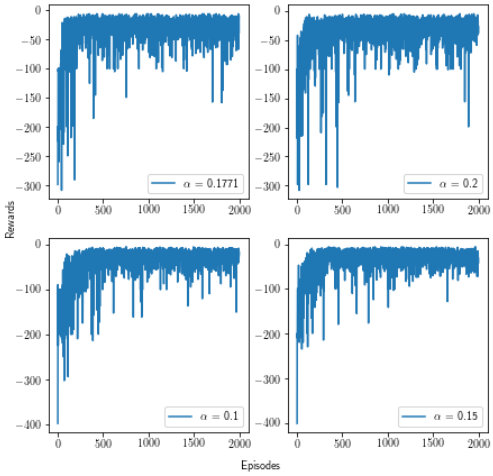
C-1-Q-learning-e-greedy $\gamma = 0.9964$ $/\epsilon = 0.05$



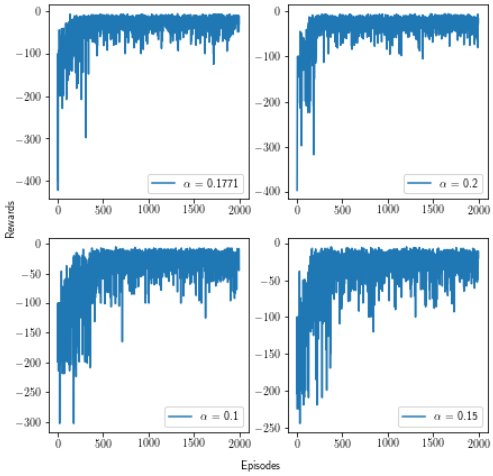
C-1-Q-learning-softmax $\gamma = 0.9964$ $\beta = 0.8304$



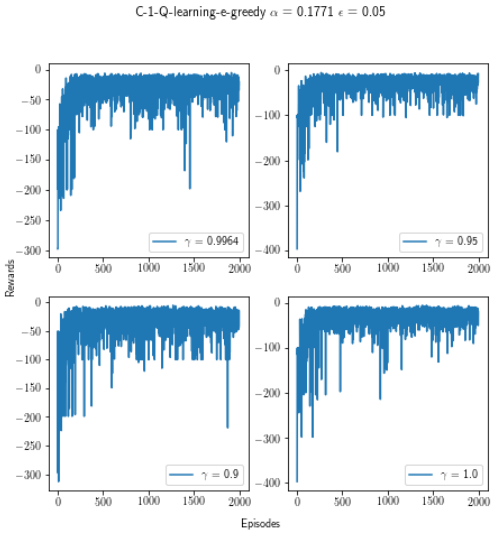
C-1-sarsa-e-greedy $\gamma = 0.9964$ $/\epsilon = 0.05$

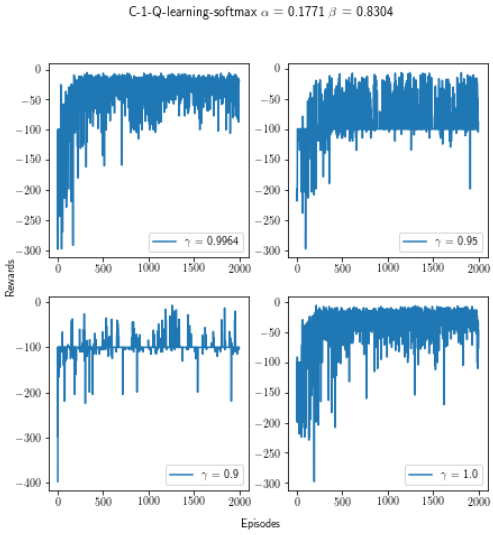


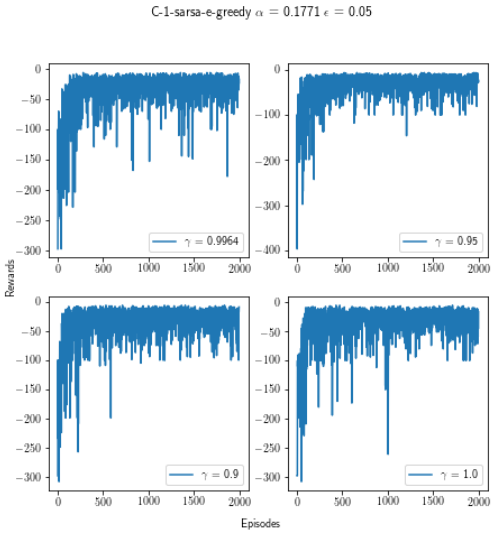
C-1-sarsa-softmax $\gamma = 0.9964$ $\beta = 0.8304$

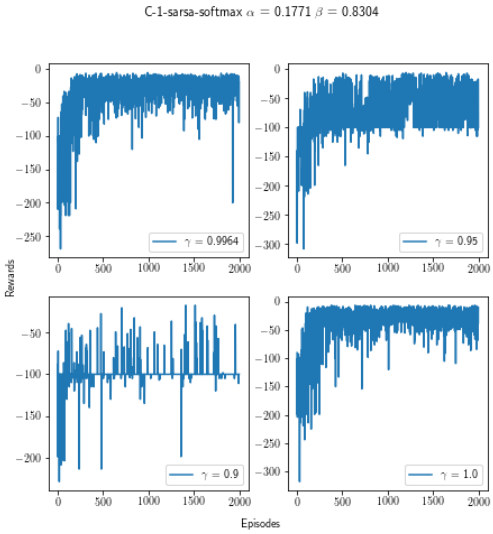


Discount Rate Variations





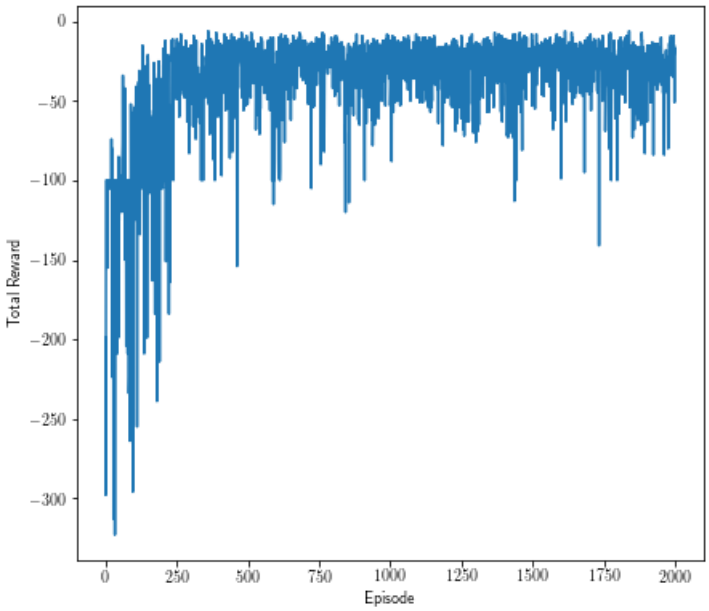


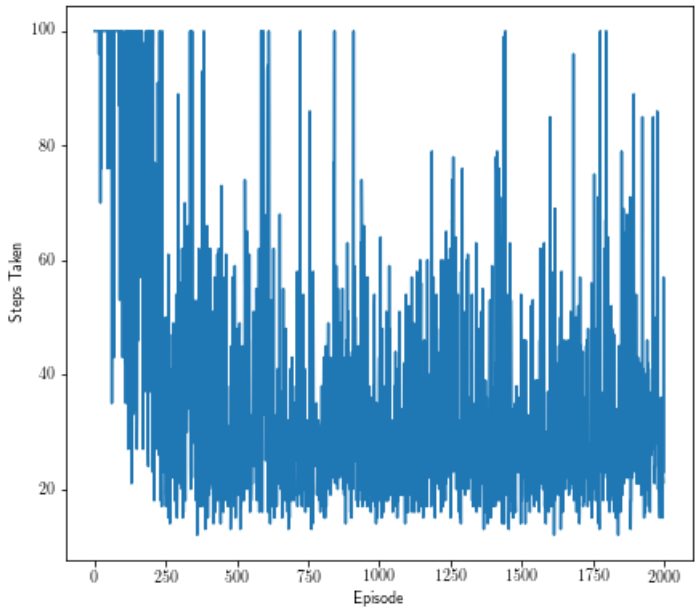


Best Plots

The plots for best set of Hyper Parameters found using Wandb analysis and also supported by variations in above section.

Reward Curve and number of steps to reach the goal





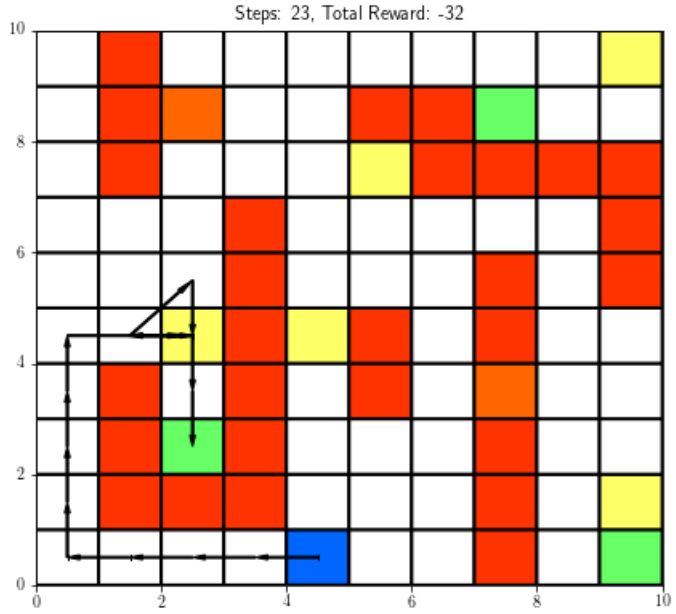
Best plots for:

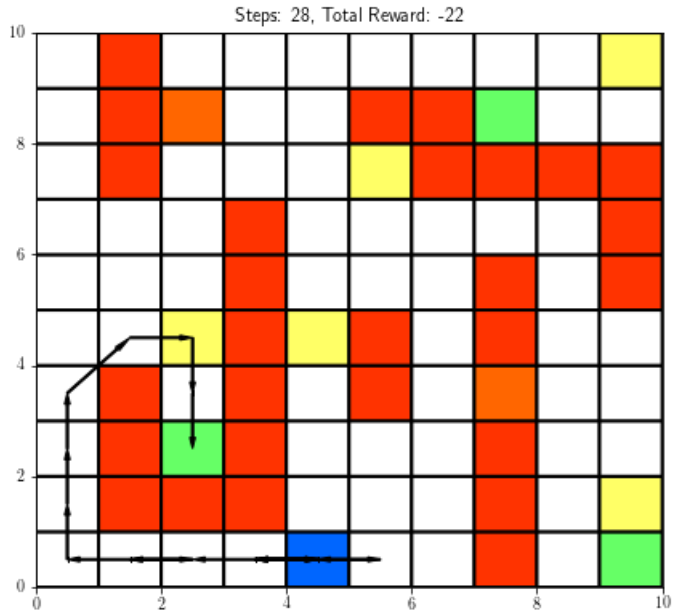
- Algorithm - Sarsa
- Policy - Softmax
- Beta - 0.8304
- Alpha - 0.1771
- Gamma - 0.9964

Final Learned Policy

In this configuration along with the wind we have p=0.7, which further adds the stochasticity w.r.t. the transitions. In fact, this time even for an optimal policy the agent could take some offbeat paths like the one given below, increasing the variance in the rewards and number of steps to reach the goal.

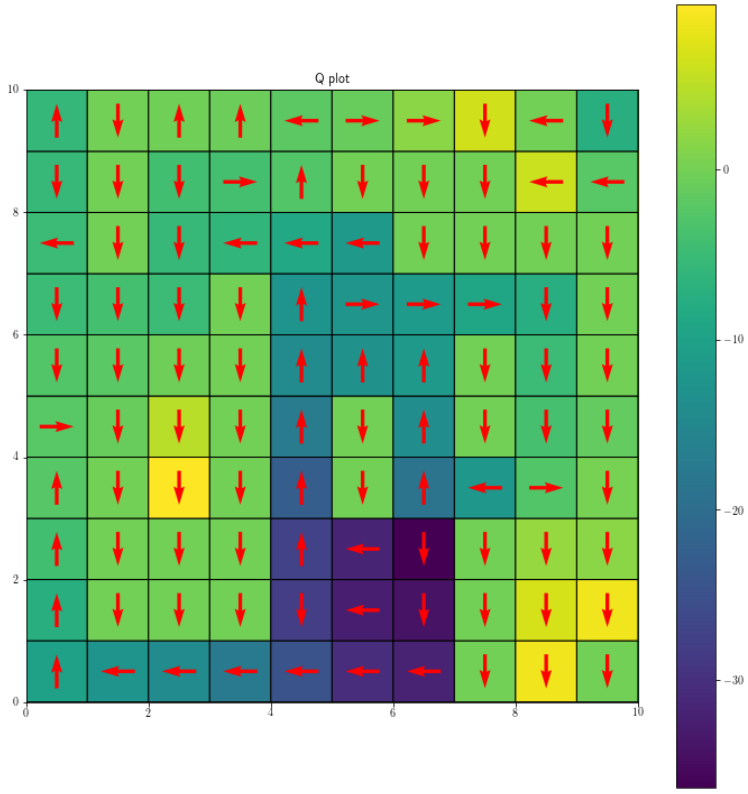
The trained agent is made to explore the environment for 2 runs. The visited states and actions are plotted in the next page.


Steps: 23, Total Reward: -32
10
8
6
4
2
0
0
2
4
6
8
10


Steps: 28, Total Reward: -22
10
8
6
4
2
0
0
2
4
6
8
10

HeatMap

Heatmap of Grid World with Q values and optimal actions (of final policy learnt)

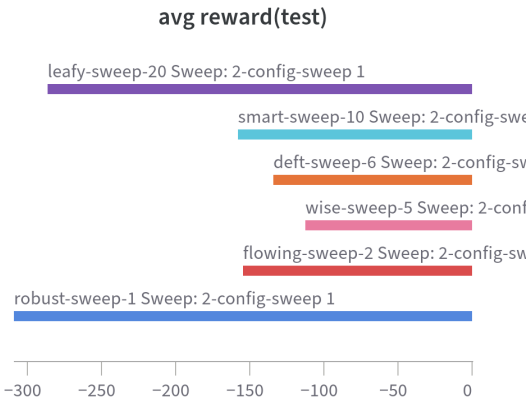


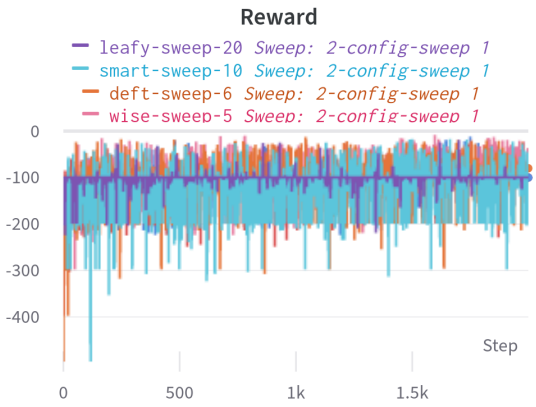
Configuration 2

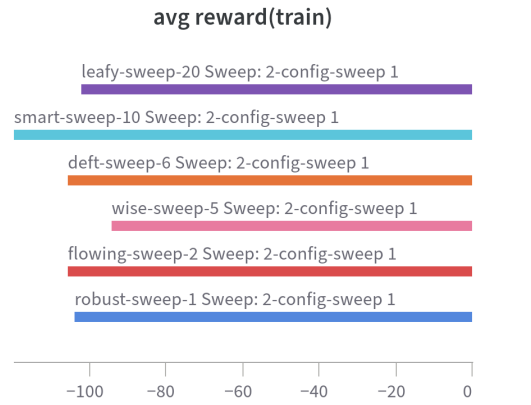
Configuration parameters

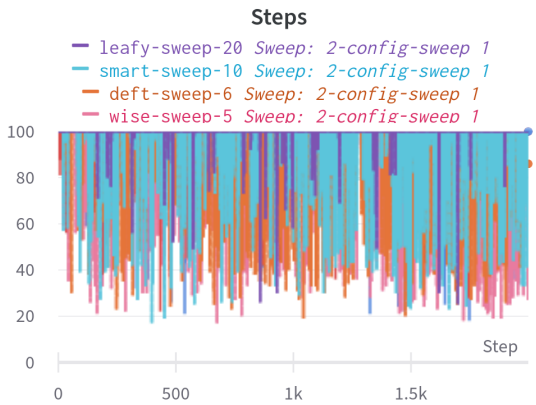
Wind = **True**, Start State = **[0,4]**, p = **0.35**

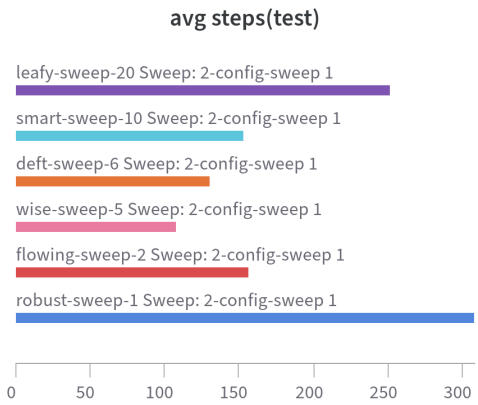
Wandb Analysis

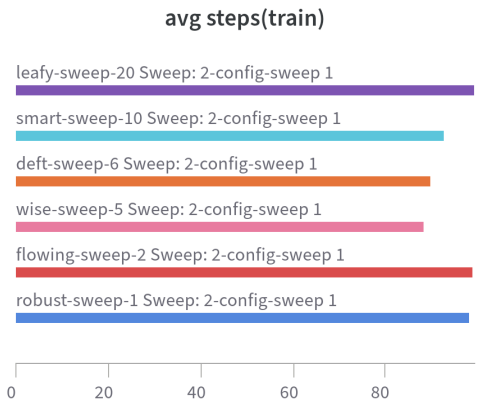


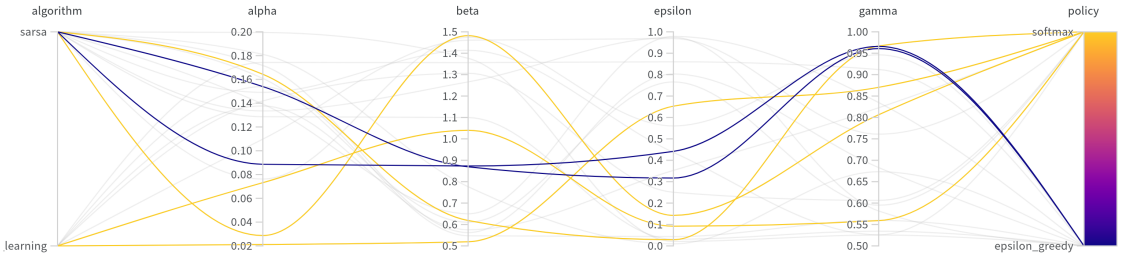










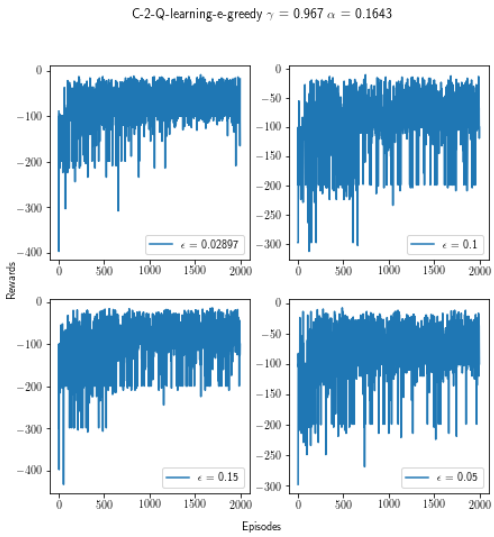


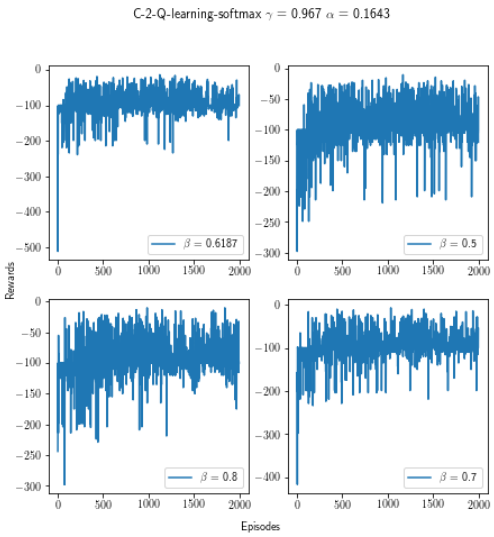
Hyper-parameter Plots

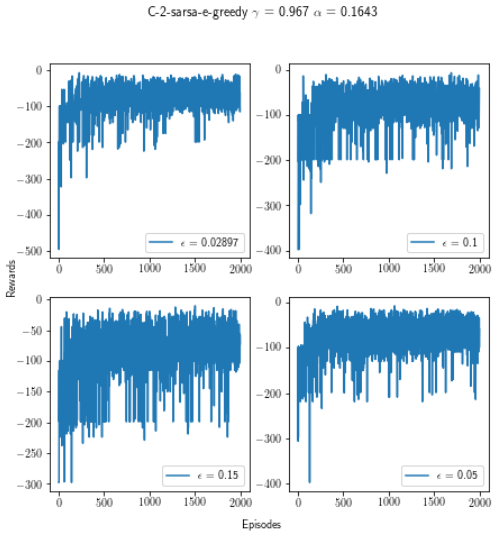
Fixing the best Hyper parameters found in the above section, other values are tried to in order to verify the observations.

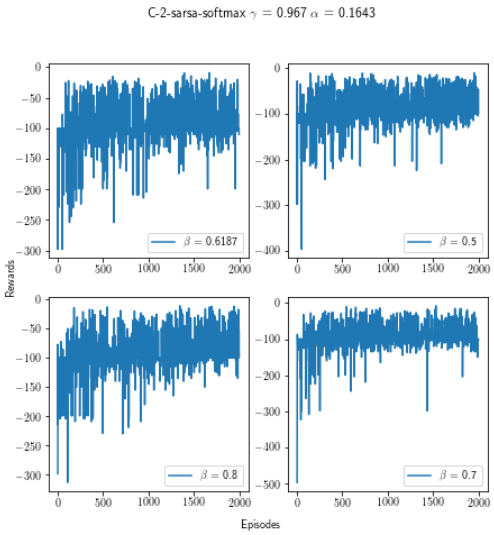
Policy Greed Variations

In this for each Algorithm and policy, α and γ are fixed and the policy greed i.e., ϵ or β , depending on the policy, is varied.







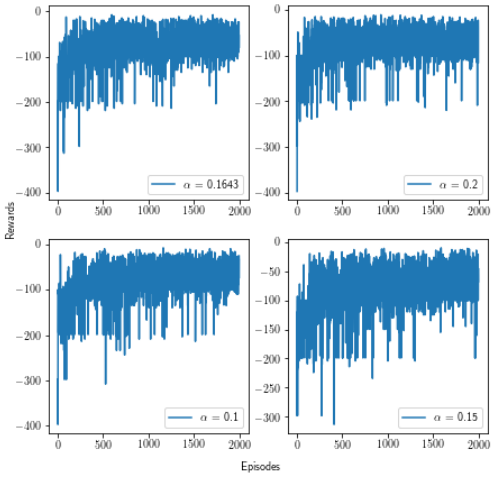


Inferences:

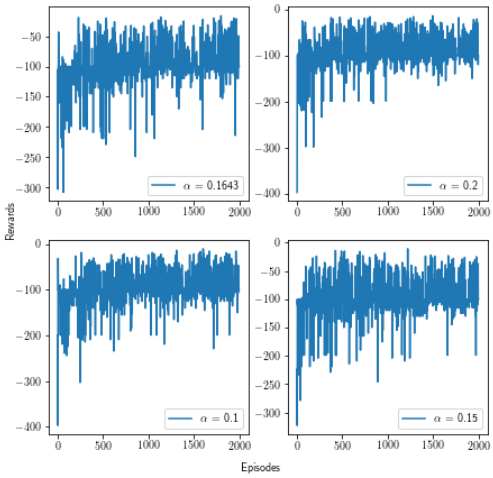
Learning Rate Variations

In this for each Algorithm and policy, policy greed and γ are fixed and Learning Rate α is varied.

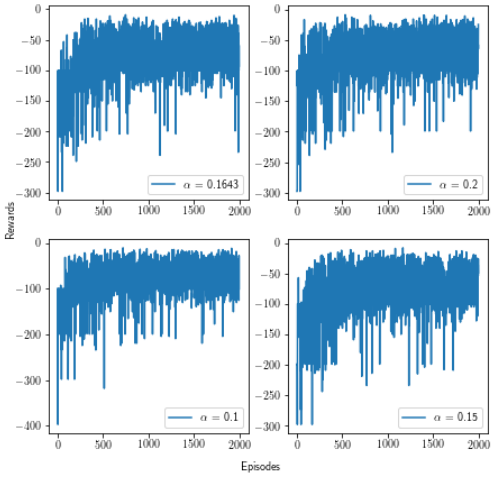
C-2-Q-learning-e-greedy $\gamma = 0.967$ $/\epsilon = 0.02897$



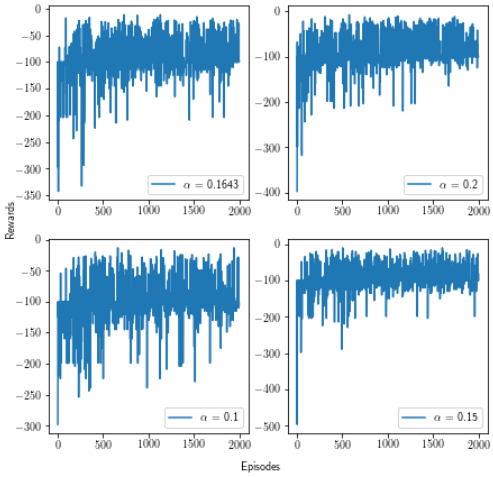
C-2-Q-learning-softmax $\gamma = 0.967$ $\beta = 0.6187$



C-2-sarsa-e-greedy $\gamma = 0.967$ $/\epsilon = 0.02897$

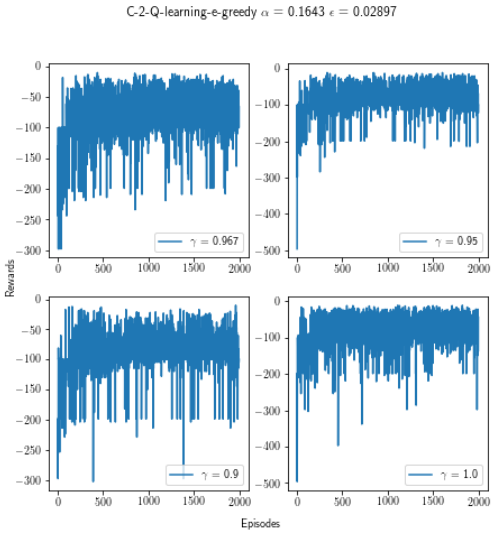


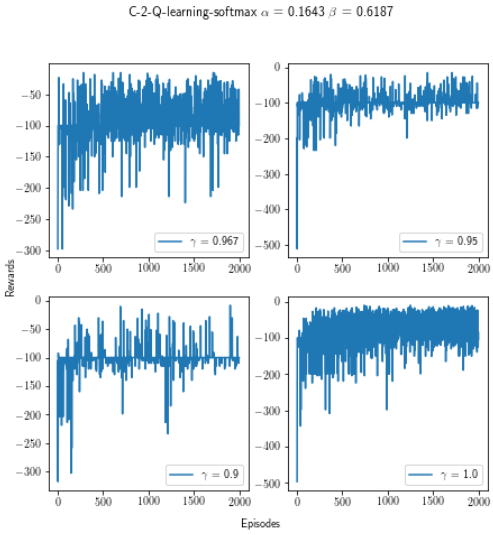
C-2-sarsa-softmax $\gamma = 0.967$ $\beta = 0.6187$

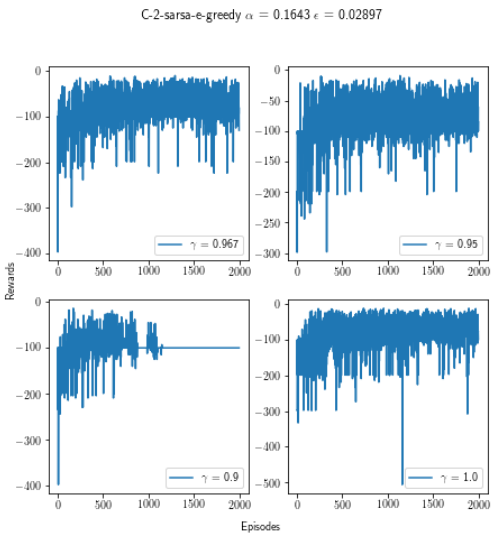


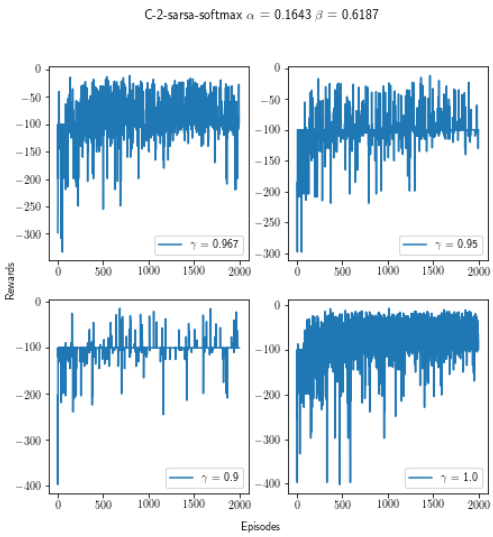
Inferences:

Discount Rate Variations







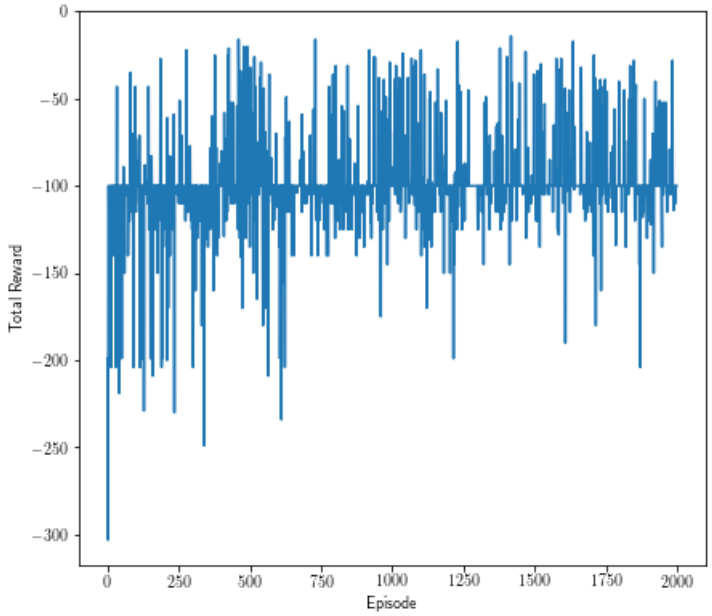


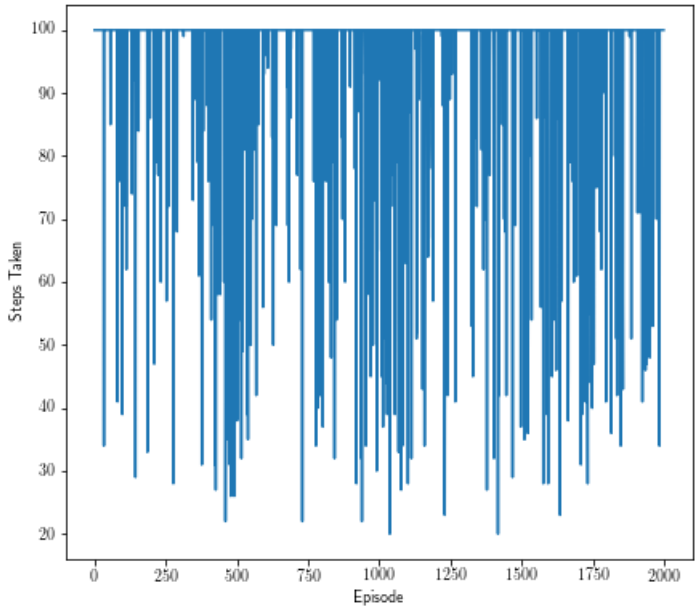
Inferences:

Best Plots

The plots for best set of Hyper Parameters found using Wandb analysis and also supported by variations in above section.

Reward Curve





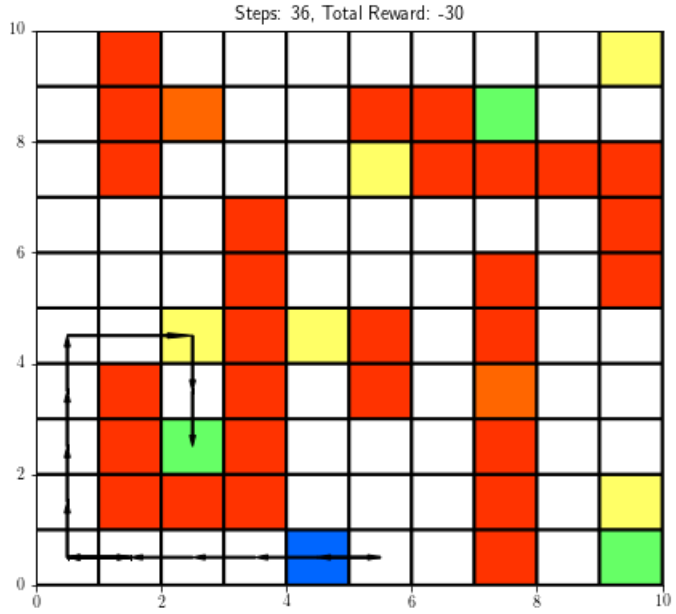
Best plots for:

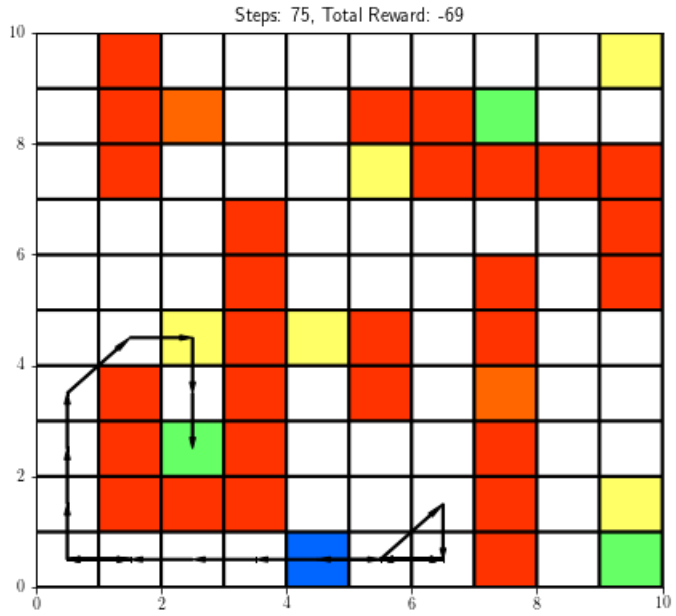
- Algorithm - Sarsa
- Policy - Softmax
- Beta - 0.6187
- Alpha - 0.1643
- Gamma - 0.967

In this configuration, the p value is reduced to 0.35, which further increases the stochasticity, and yet again we observe an increased variance in the reward curve and steps taken to reach the goal curve.

Final Learned Policy

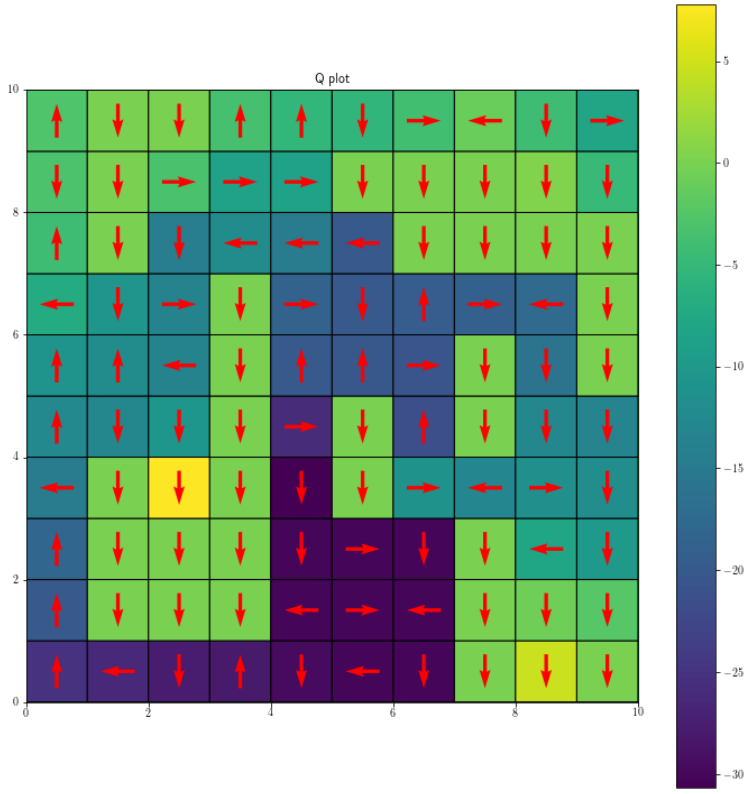
The trained agent is made to explore the environment for 2 runs. The visited states and actions are plotted


Steps: 36, Total Reward: -30
0
2
4
6
8
10


Steps: 75, Total Reward: -69
0
2
4
6
8
10

HeatMap

Heatmap of Grid World with Q values and optimal actions (of final policy learnt)



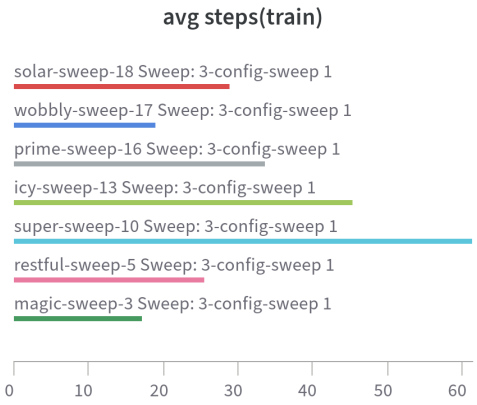
We can also observe that the optimal actions in the previous 3 configurations(config-0,1,2) are very much same. Yet, config-1,2 could give rise to some off-beat paths from the shortest ones because p is not equal to one in these cases.

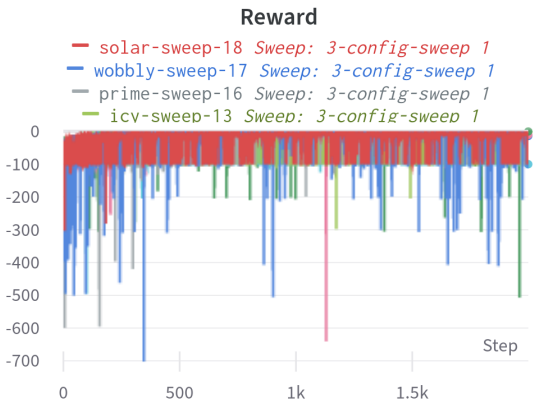
Configuration 3

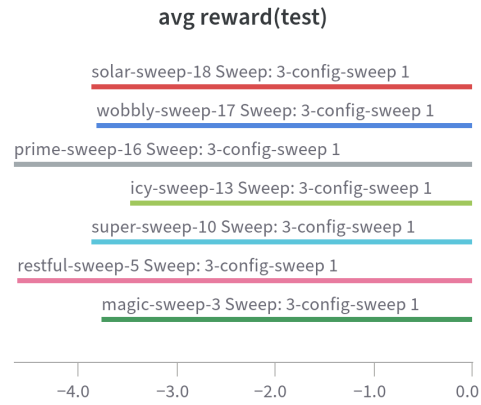
Configuration parameters

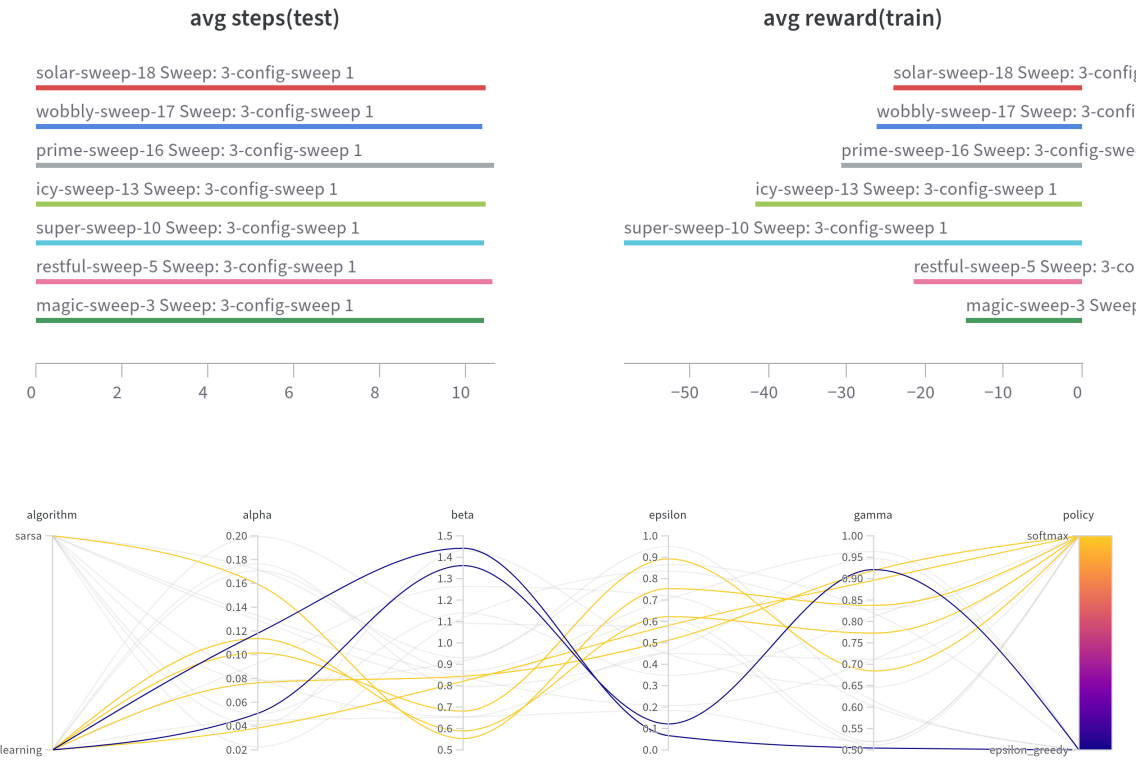
Wind = **True**, Start State = **[3,6]**, p = **1.0**

Wandb Analysis







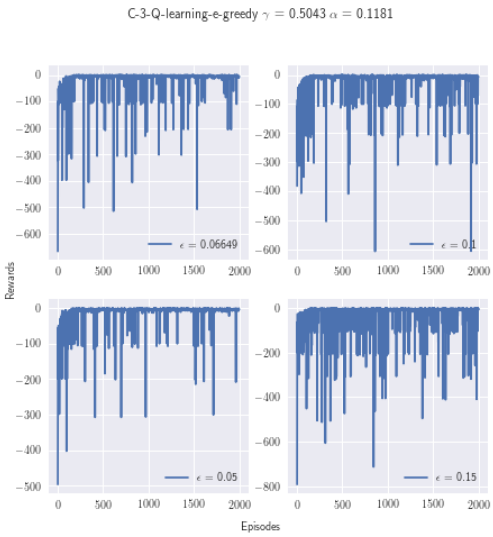


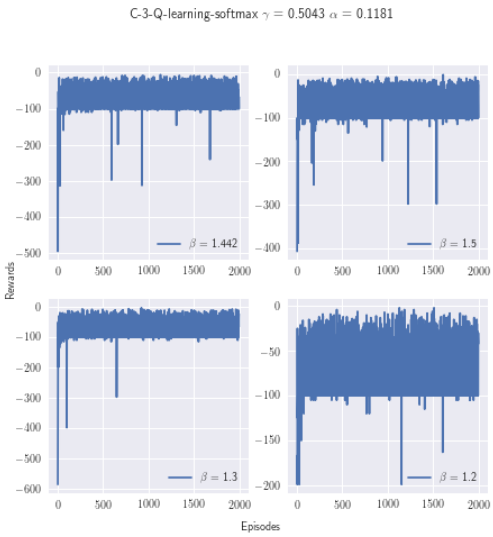
Hyper-parameter Plots

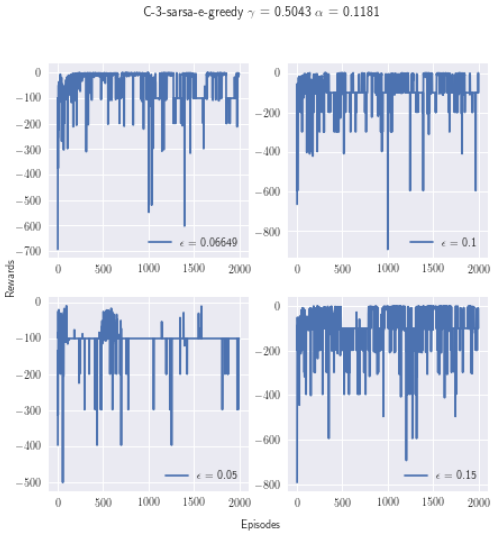
Fixing the best Hyper parameters found in the above section, other values are tried to in order to verify the observations.

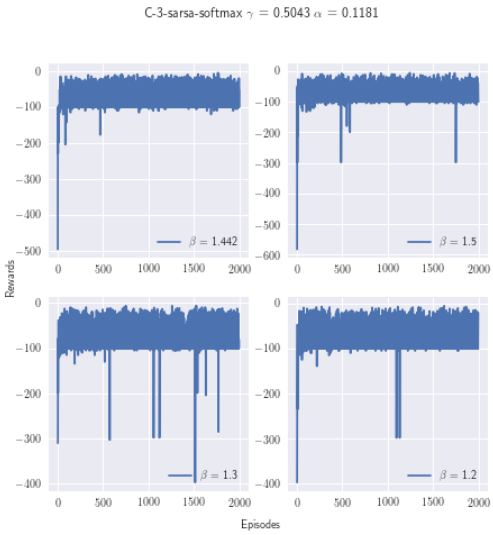
Policy Greed Variations

In this for each Algorithm and policy, α and γ are fixed and the policy greed i.e., ϵ or β , depending on the policy, is varied.





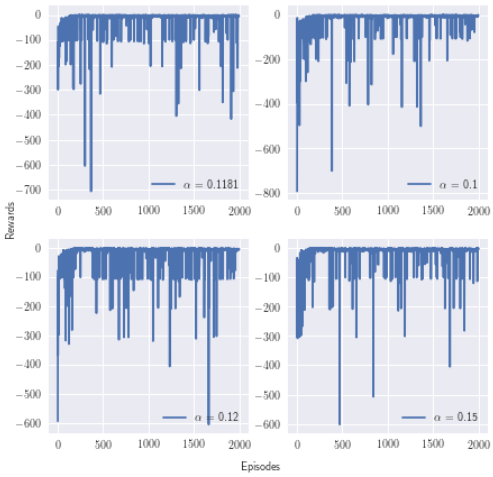




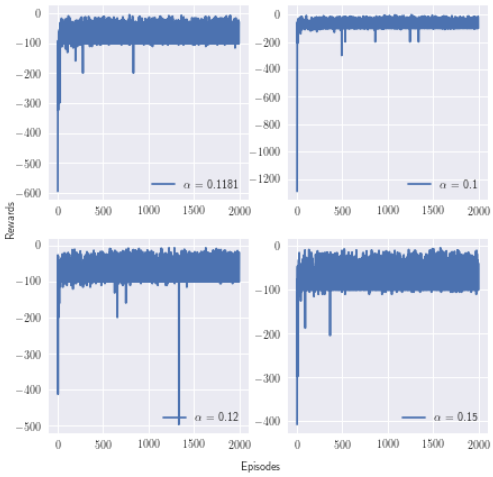
Learning Rate Variations

In this for each Algorithm and policy, policy greed and γ are fixed and Learning Rate α is varied.

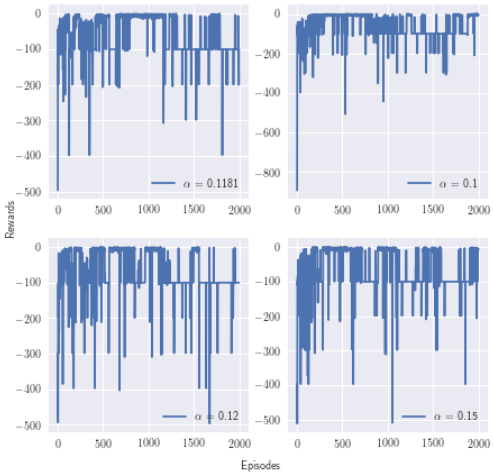
C-3-Q-learning-e-greedy $\gamma = 0.5043$ $/\epsilon = 0.06649$



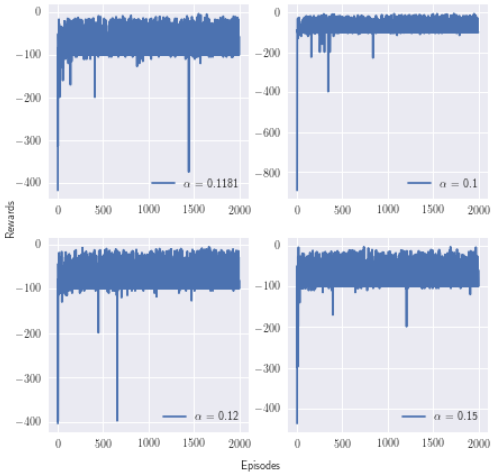
C-3-Q-learning-softmax $\gamma = 0.5043$ $\beta = 1.442$



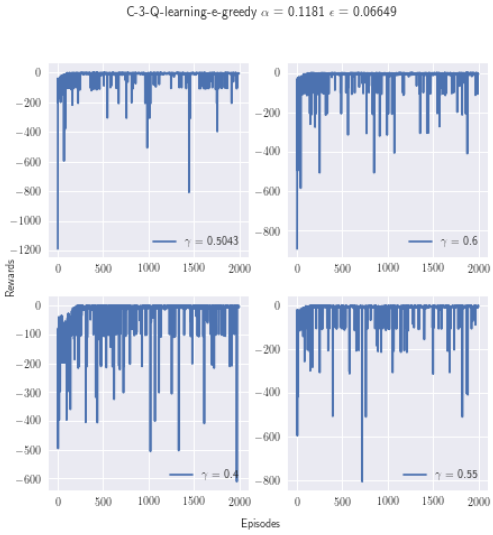
C-3-sarsa-e-greedy $\gamma = 0.5043$ $/\epsilon = 0.06649$

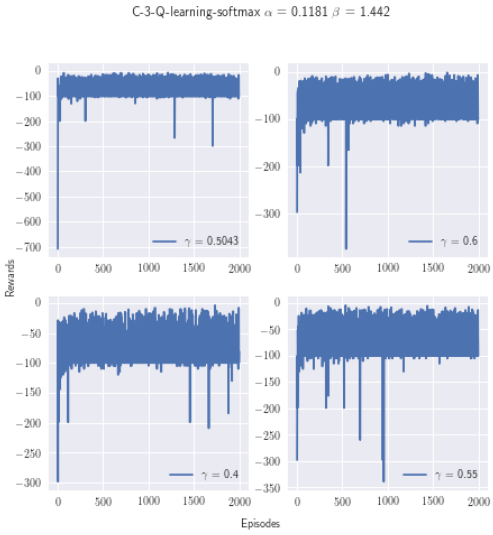


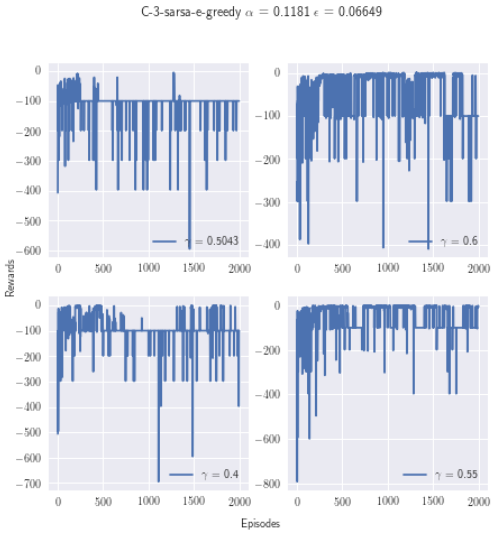
C-3-sarsa-softmax $\gamma = 0.5043$ $\beta = 1.442$

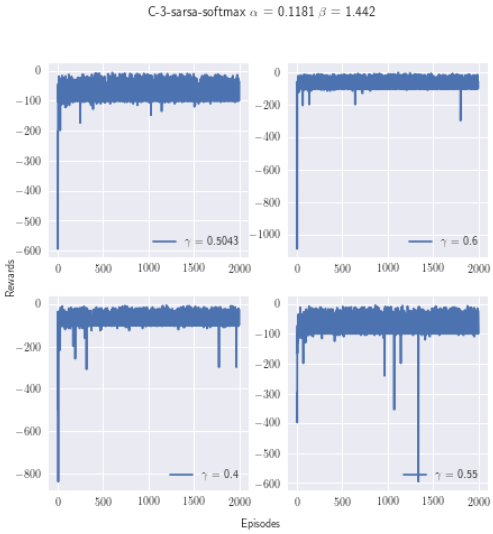


Discount Rate Variations





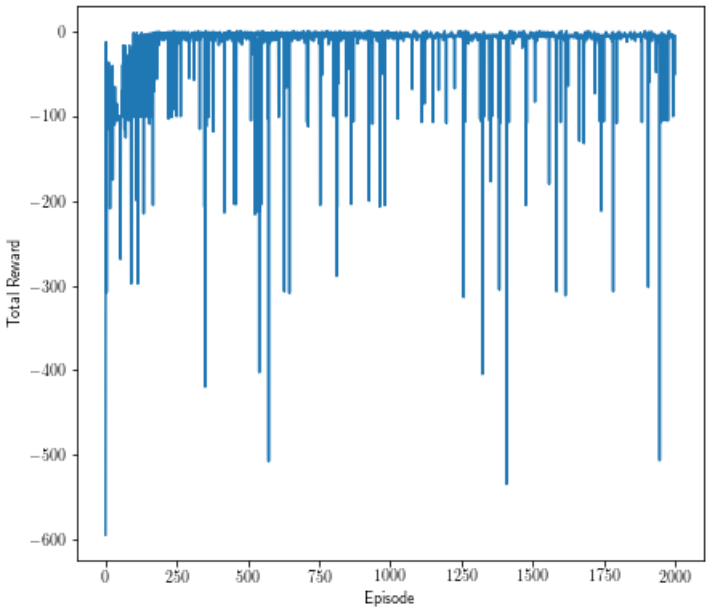


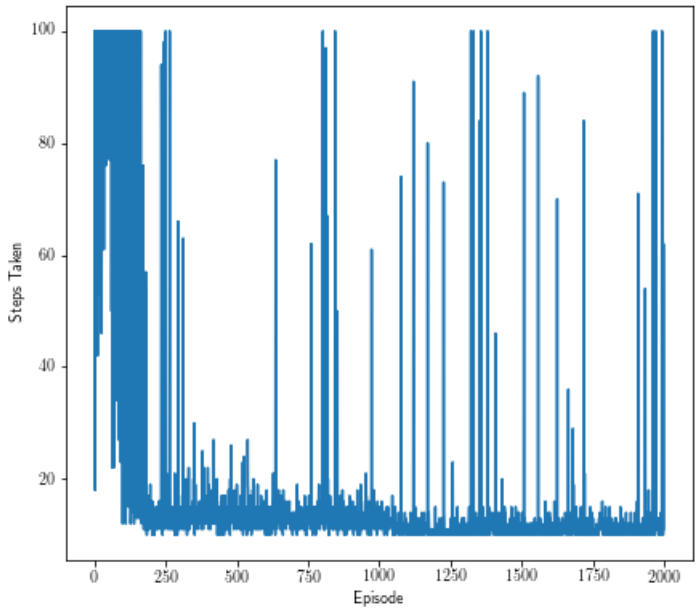


Best/interesting Plots

The plots for best set of Hyper Parameters found using Wandb analysis and also supported by variations in above section.

Reward Curve and number of steps to reach goal





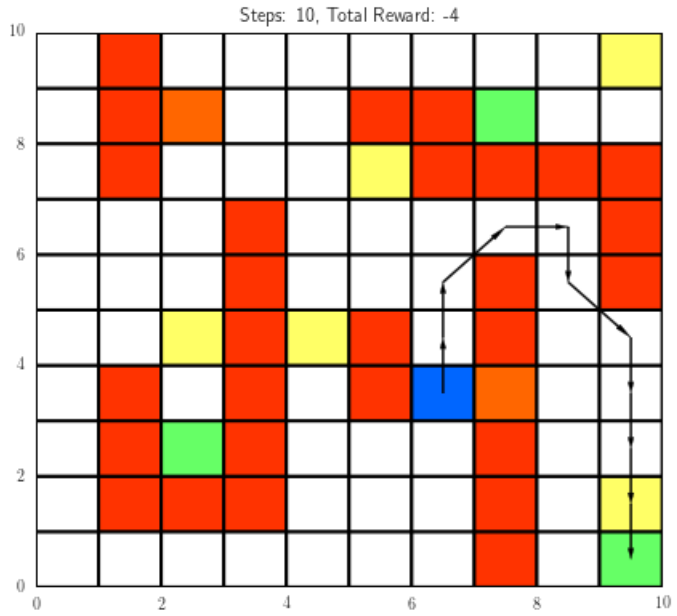
Best/interesting plots for:

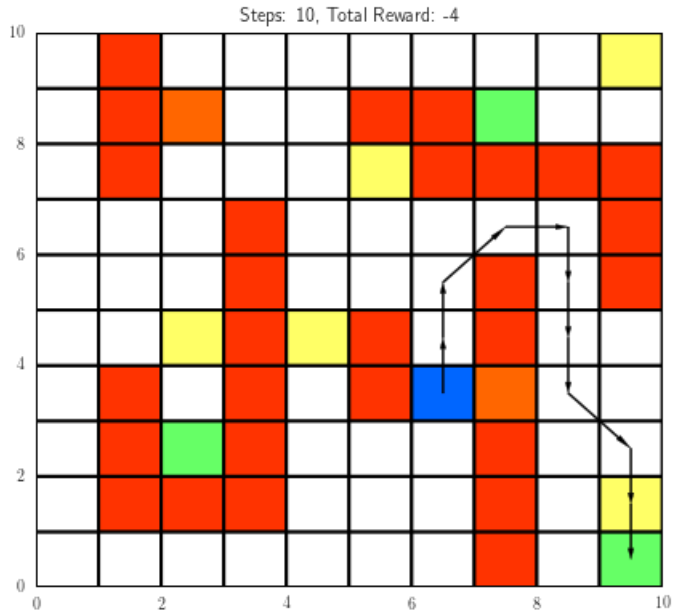
- Algorithm - Q-learning
- Policy - e-greedy
- Epsilon - 0.06649
- Alpha - 0.1181
- Gamma - 0.9615

Final Learned Policy

Now we start from a different state, for which the restart start is just to the right of it. The agent could immediately get penalised with '-100' reward with probability 0.4 by falling off into the restart state .
And even after learning the optimal route, there is a chance of the agent again falling into the restart state by a left deviation along the optimal path.
That's the reason for reward plots(particularly for e-greedy) having sudden reward drops.

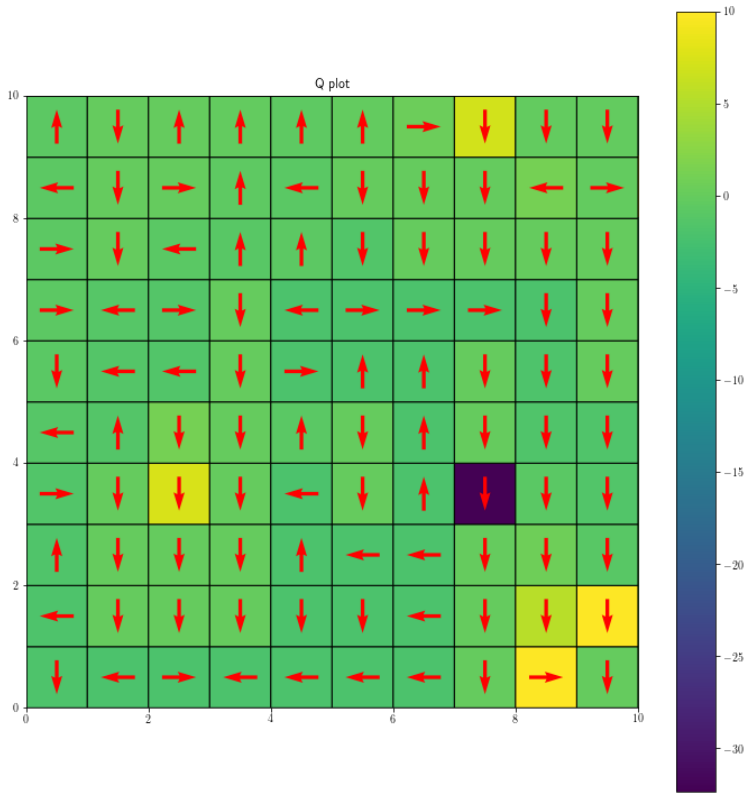
The trained agent is made to explore the environment for 2 runs. The visited states and actions are plotted


Steps: 10, Total Reward: -4
10
8
6
4
2
0
0
2
4
6
8
10


Steps: 10, Total Reward: -4
10
8
6
4
2
0
0
2
4
6
8
10

HeatMap

Heatmap of Grid World with Q values and optimal actions (of final policy learnt)



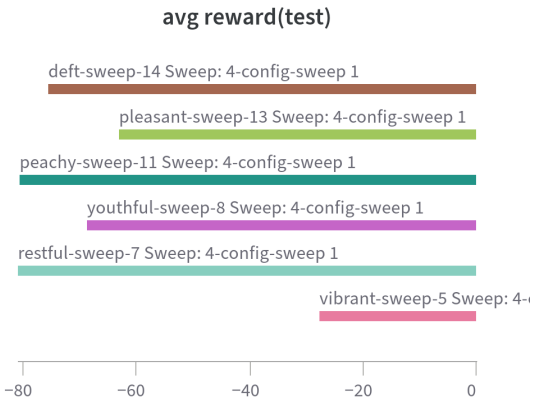
Clearly, epsilon-greedy strategy is not good here and neither is Q-learning(not very safe), but we still included it here as it would still make up for an interesting case to compare while we move to the next configuration.

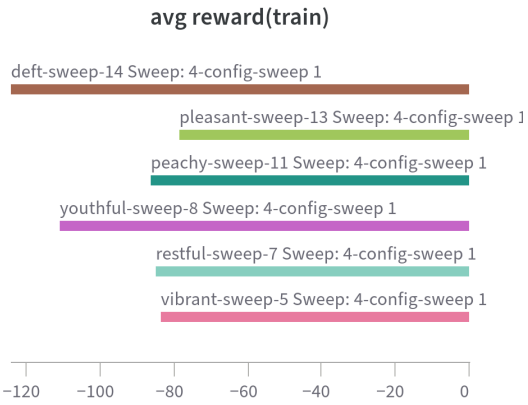
Configuration 4

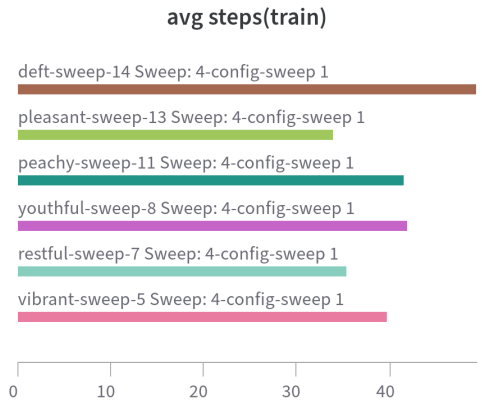
Configuration parameters

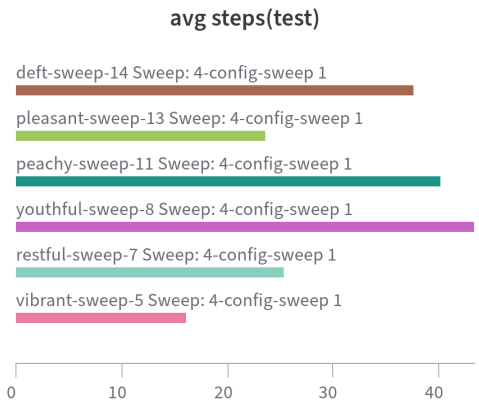
Wind = **True**, Start State = **[3,6]**, p = **0.7**

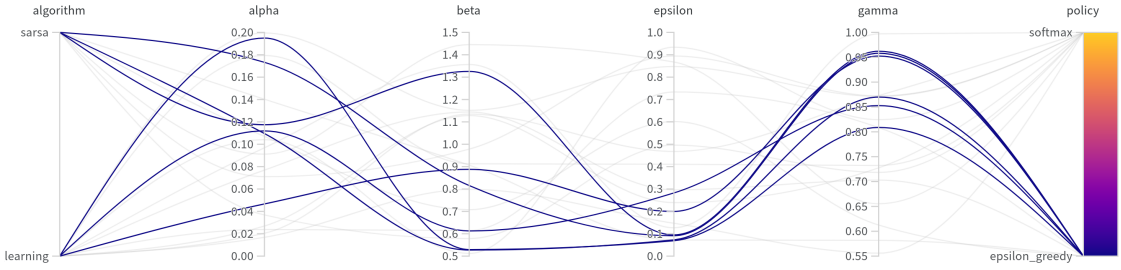
Wandb Analysis









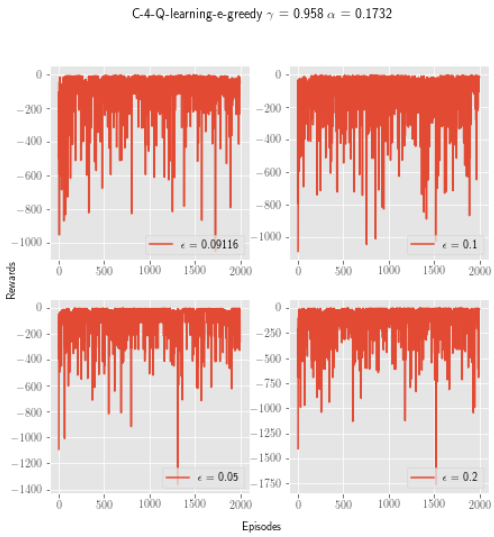


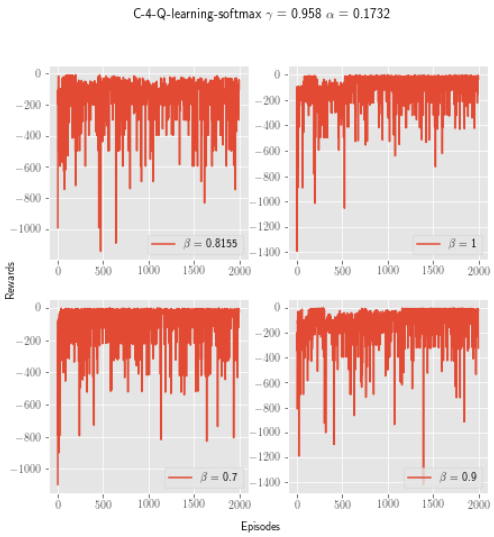
Hyper-parameter Plots

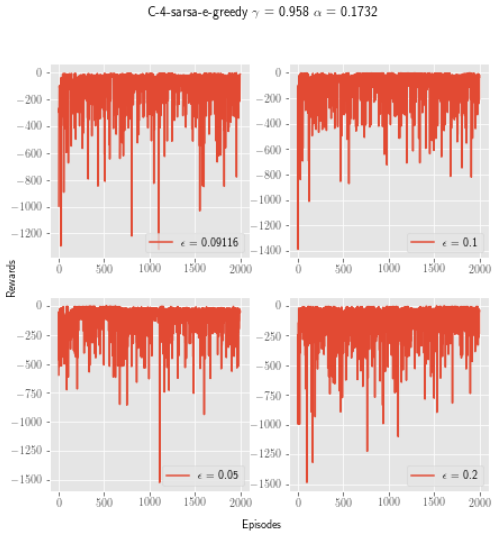
Fixing the best Hyper parameters found in the above section, other values are tried to in order to verify the observations.

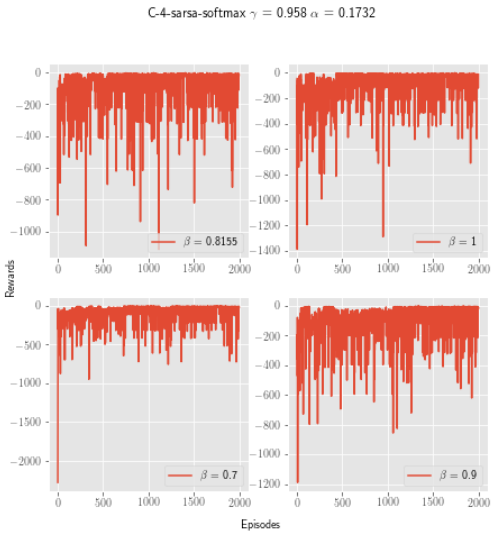
Policy Greed Variations

In this for each Algorithm and policy, α and γ are fixed and the policy greed i.e., ϵ or β , depending on the policy, is varied.



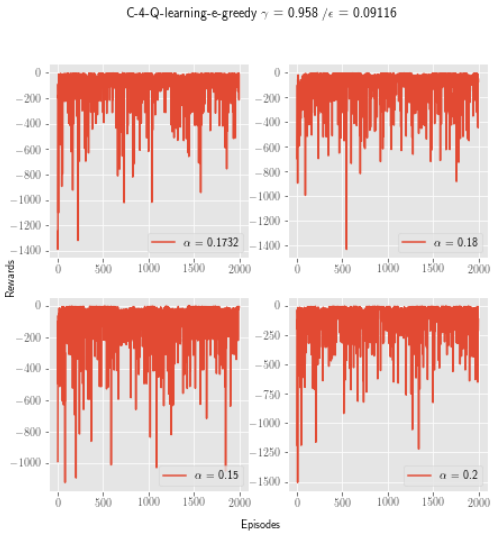


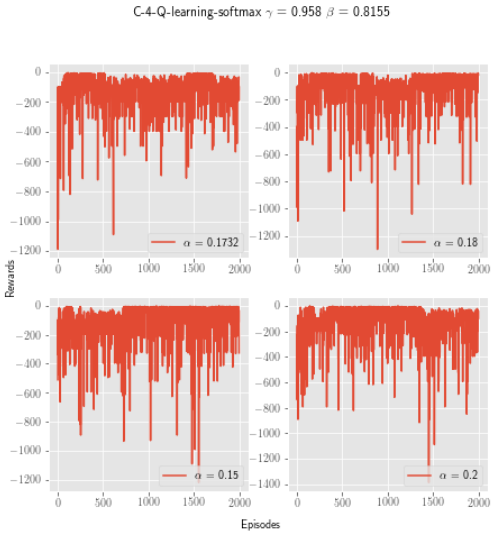


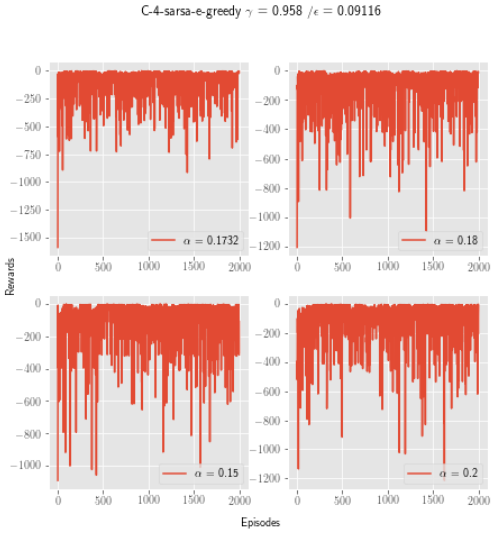


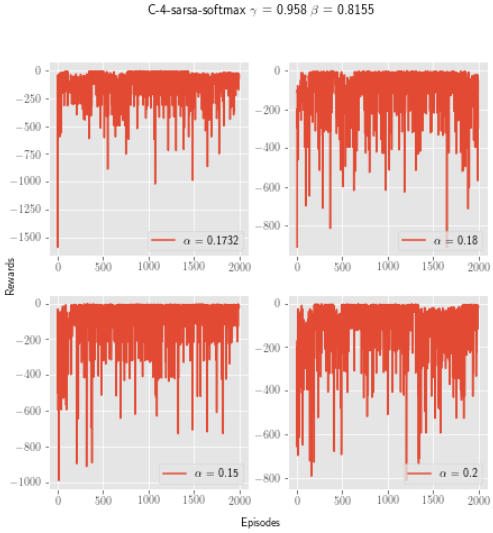
Learning Rate Variations

In this for each Algorithm and policy, policy greed and γ are fixed and Learning Rate α is varied.

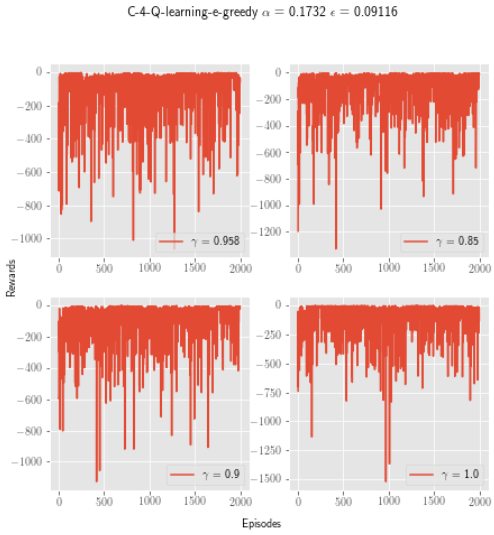

C-4-Q-learning-e-greedy $\gamma = 0.958$ $/\epsilon = 0.09116$
$\alpha = 0.1732$
$\alpha = 0.18$
$\alpha = 0.15$
$\alpha = 0.2$
Rewards
Episodes

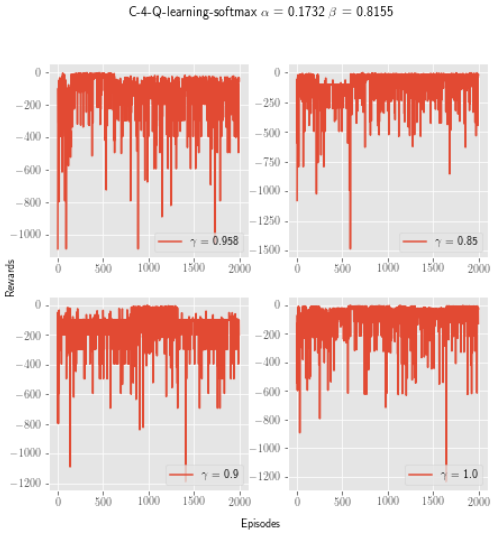

C-4-Q-learning-softmax $\gamma = 0.958$ $\beta = 0.8155$
$\alpha = 0.1732$
$\alpha = 0.18$
$\alpha = 0.15$
$\alpha = 0.2$
Rewards
Episodes

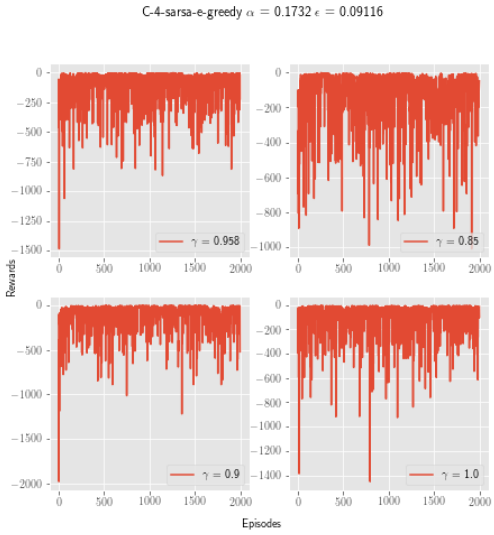

C-4-sarsa-e-greedy $\gamma = 0.958$ $/\epsilon = 0.09116$
$\alpha = 0.1732$
$\alpha = 0.18$
$\alpha = 0.15$
$\alpha = 0.2$
Rewards
Episodes

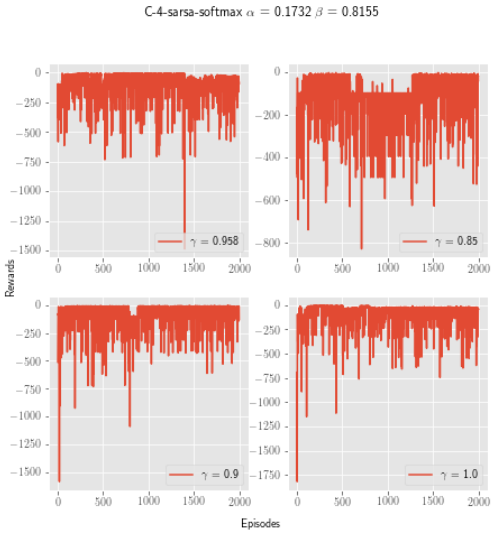

C-4-sarsa-softmax $\gamma = 0.958$ $\beta = 0.8155$
$\alpha = 0.1732$
$\alpha = 0.18$
$\alpha = 0.15$
$\alpha = 0.2$
Rewards
Episodes

Discount Rate Variations





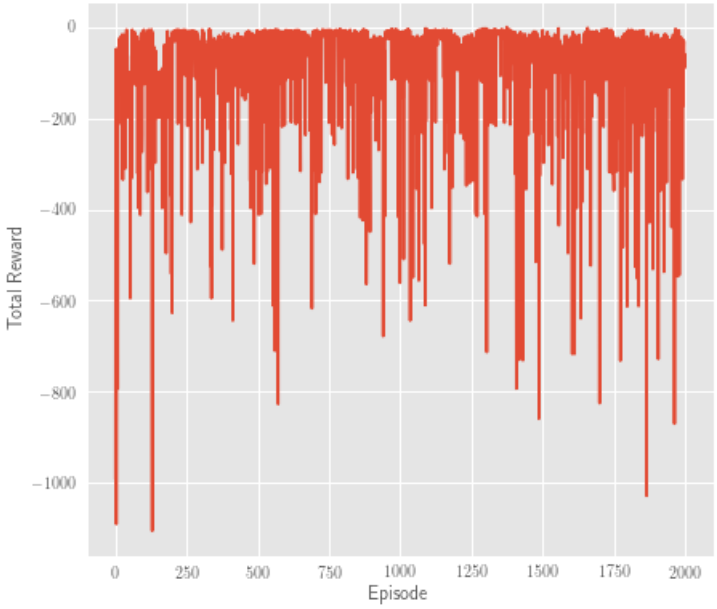


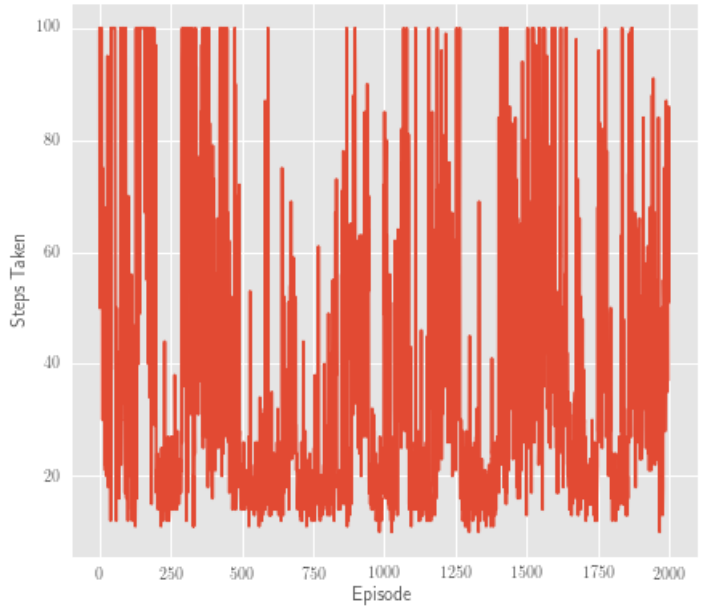


Best Plots

The plots for best set of Hyper Parameters found using Wandb analysis and also supported by variations in above section.

Reward Curve





Best plots for:

- Algorithm - Sarsa
- Policy - e-greedy
- Epsilon - 0.09116
- Alpha - 0.1732
- Gamma - 0.9615

Final Learned Policy

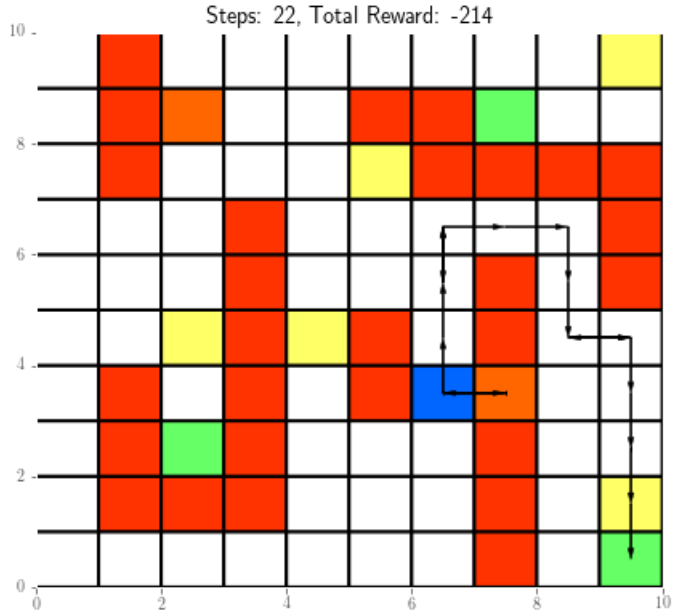
One important thing to note here is that p is not equal to one but 0.7, which would further increase the chances of the agent falling into the restart state.

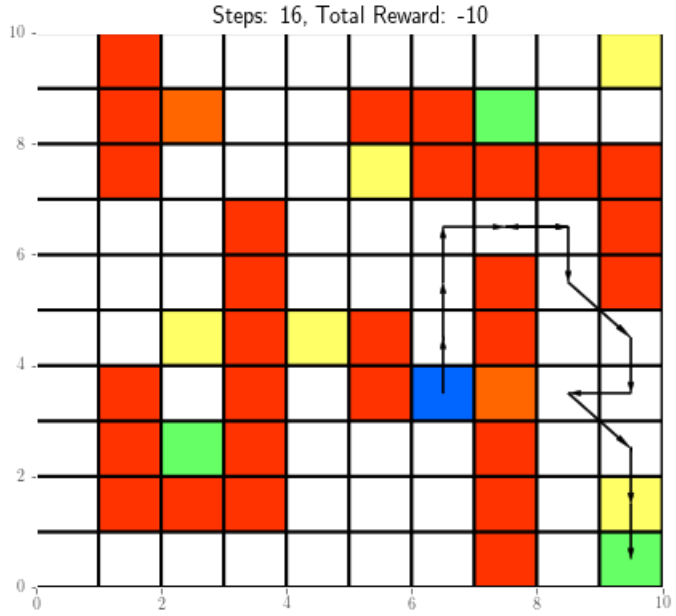
Also note that the we've chosen the best plot for Sarsa, which has resulted in a change in the optimal route to end-state(it's much safer now).

Also e-greedy/ softmax really doesn't make much of a difference here as the p=0.7 does the job of deviations required to push the agent into the restart state.

The number of steps have also increased, this could imply that the agent has learnt paths to other end-states as well.

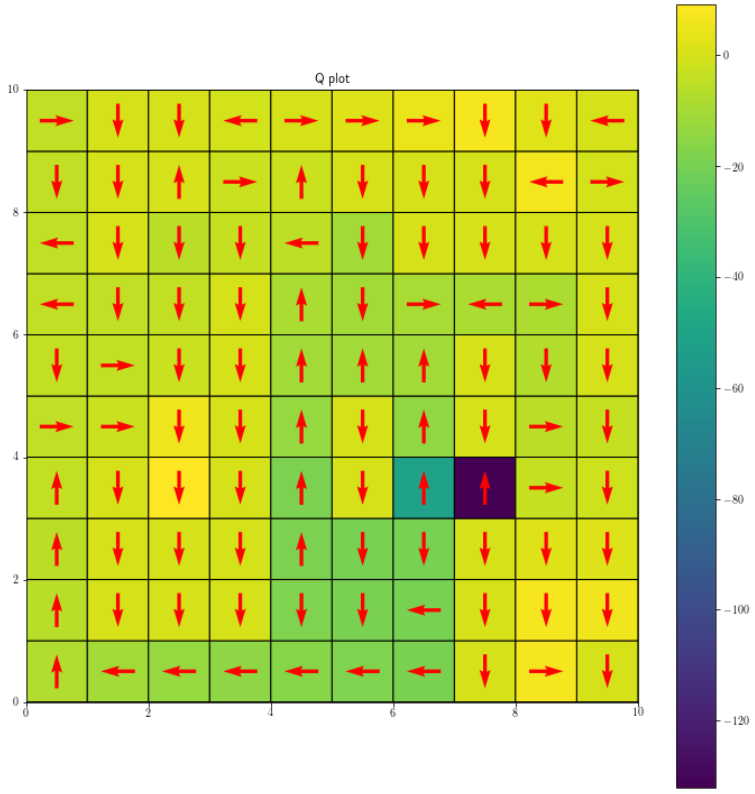
The trained agent is made to explore the environment for 2 runs. The visited states and actions are plotted.


Steps: 22, Total Reward: -214
0
2
4
6
8
10


Steps: 16, Total Reward: -10
0
2
4
6
8
10

HeatMap

Heatmap of Grid World with Q values and optimal actions (of final policy learnt)



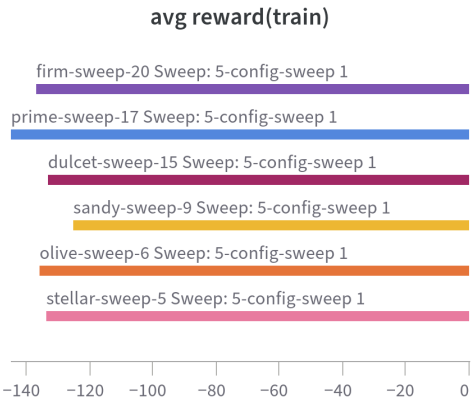
Much safer optimal route is indicated above.

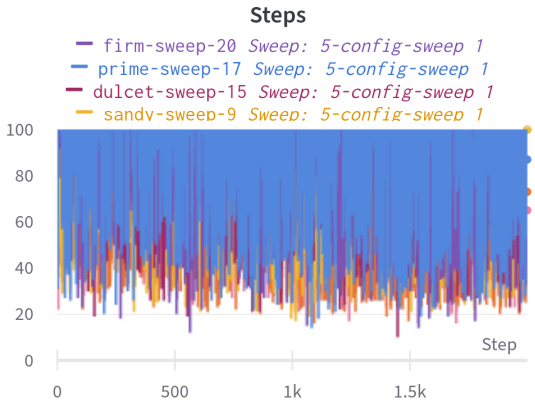
Configuration 5

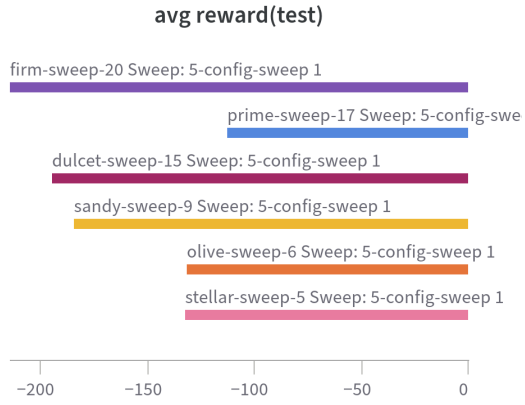
Configuration parameters

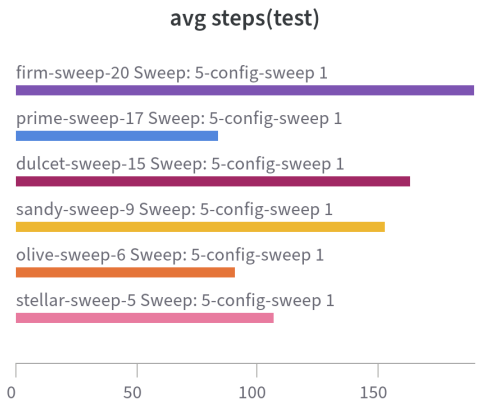
Wind = **True**, Start State = **[3,6]**, p = **0.35**

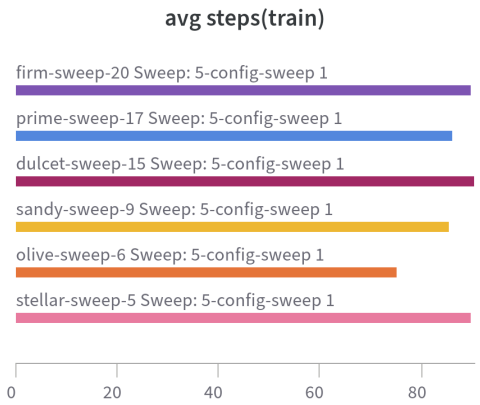
Wandb Analysis

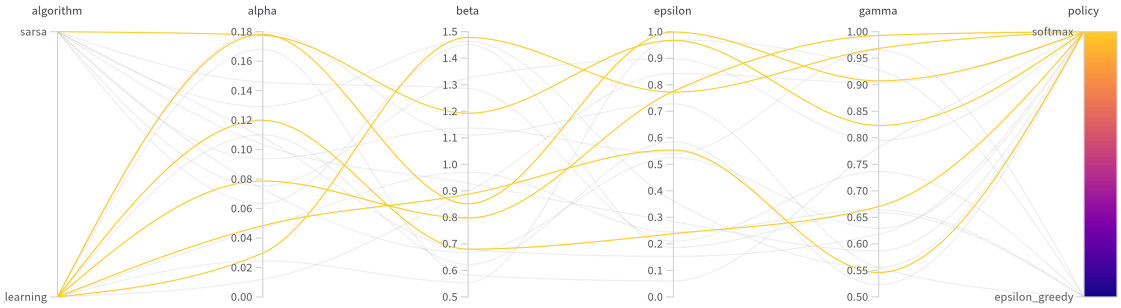










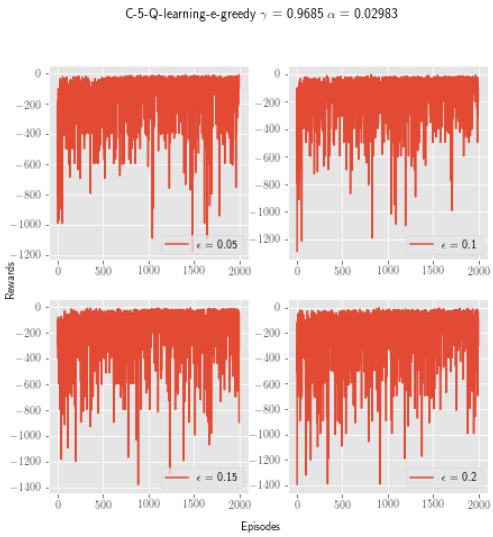


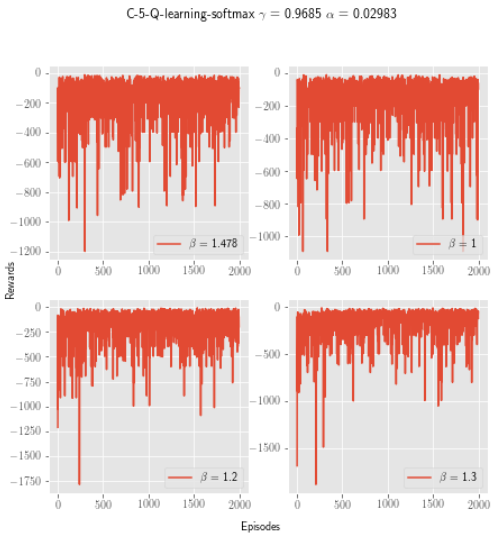
Hyper-parameter Plots

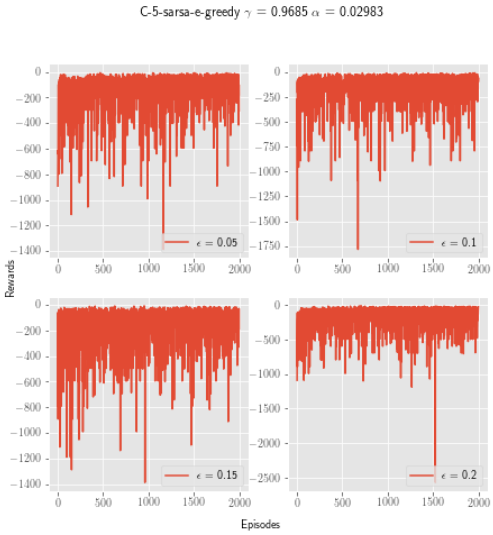
Fixing the best Hyper parameters found in the above section, other values are tried to in order to verify the observations.

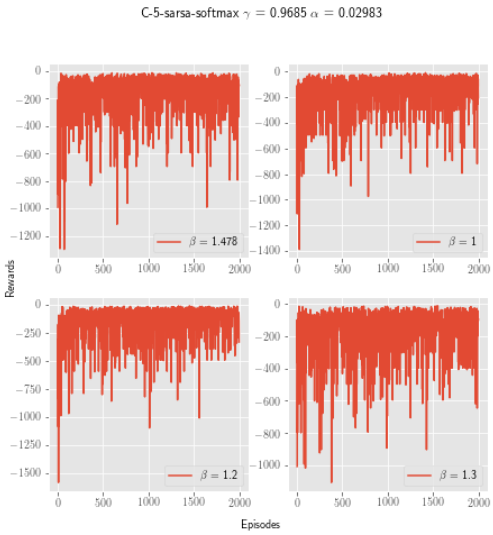
Policy Greed Variations

In this for each Algorithm and policy, α and γ are fixed and the policy greed i.e., ϵ or β , depending on the policy, is varied.





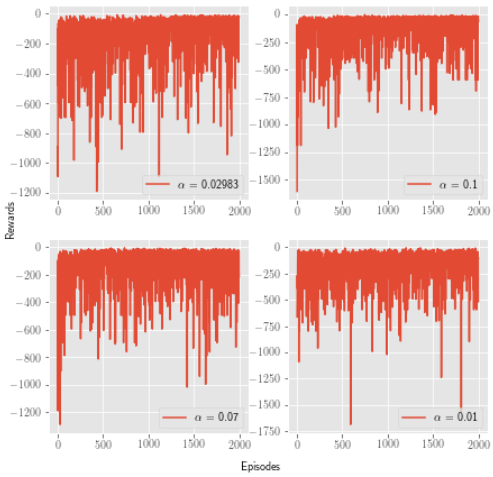




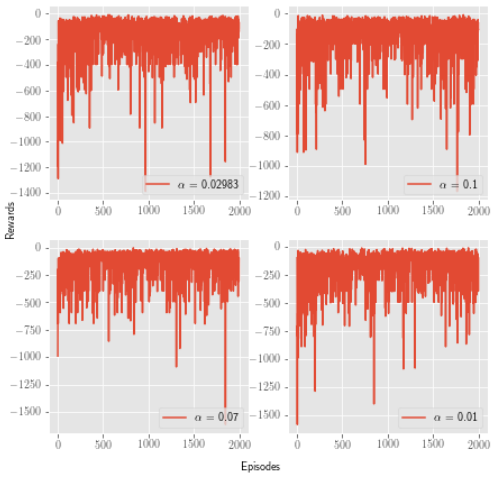
Learning Rate Variations

In this for each Algorithm and policy, policy greed and γ are fixed and Learning Rate α is varied.

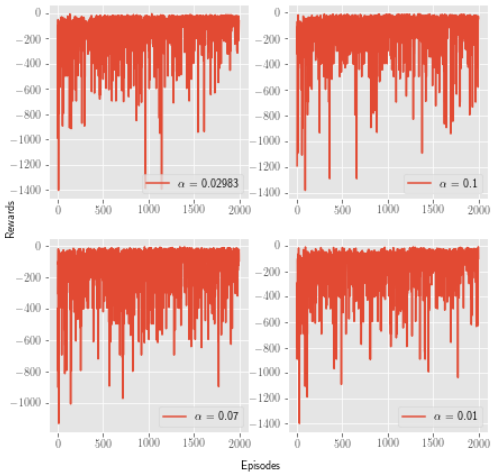
C-5-Q-learning-e-greedy $\gamma = 0.9685$ $/\epsilon = 0.05$



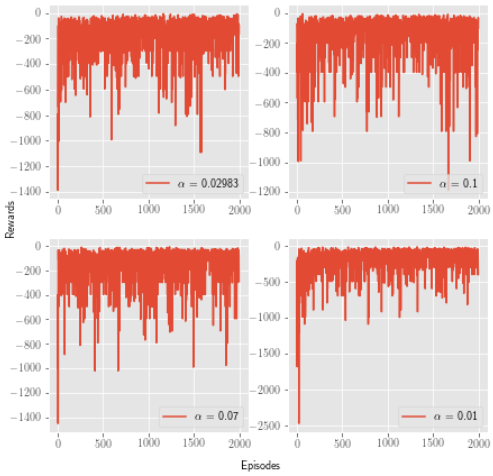
C-5-Q-learning-softmax $\gamma = 0.9685$ $\beta = 1.478$



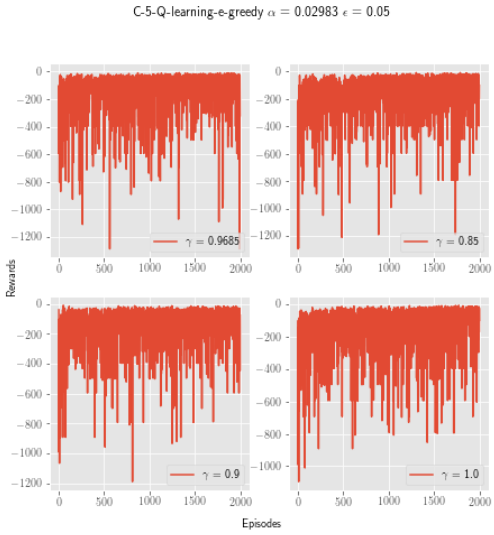
C-5-sarsa-e-greedy $\gamma = 0.9685$ $/\epsilon = 0.05$

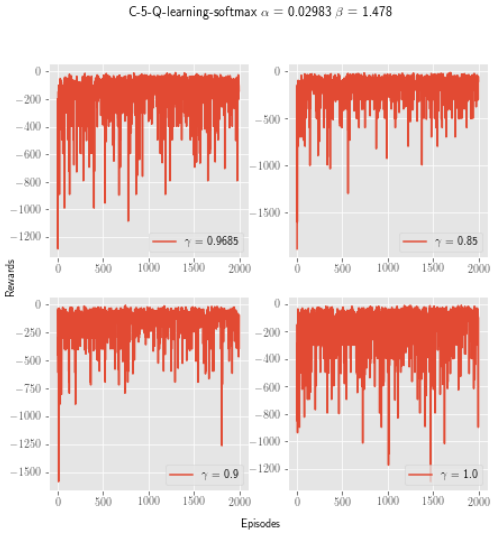


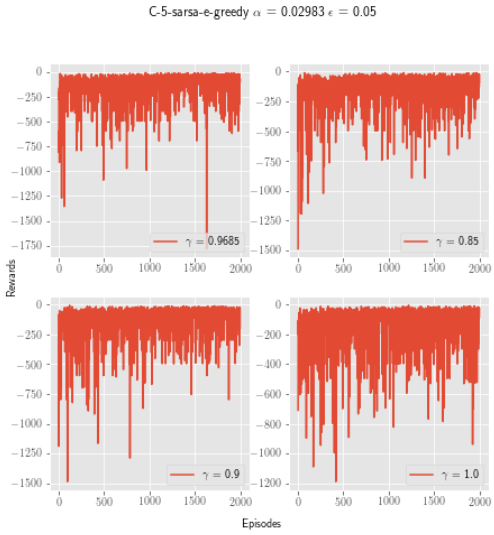
C-5-sarsa-softmax $\gamma = 0.9685$ $\beta = 1.478$

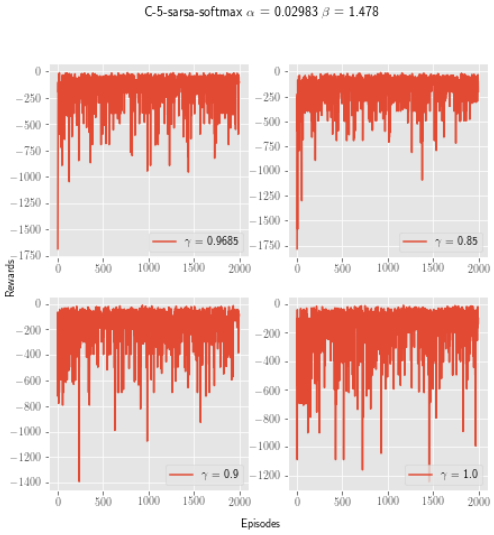


Discount Rate Variations





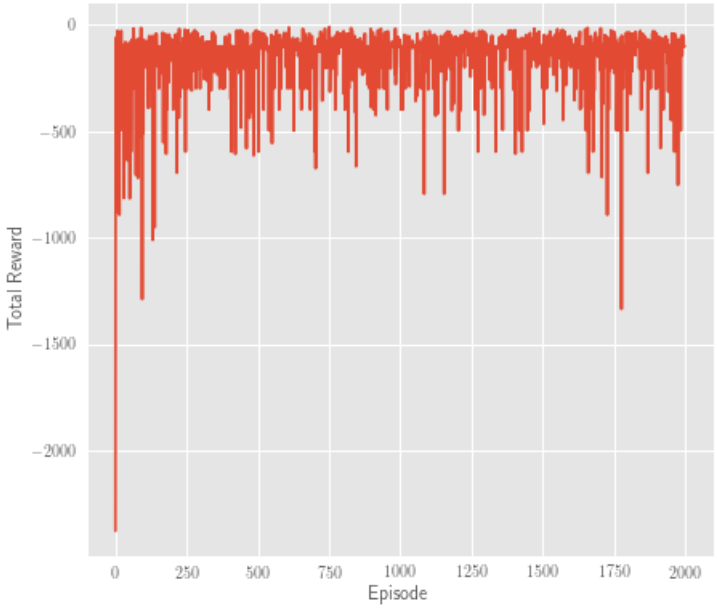


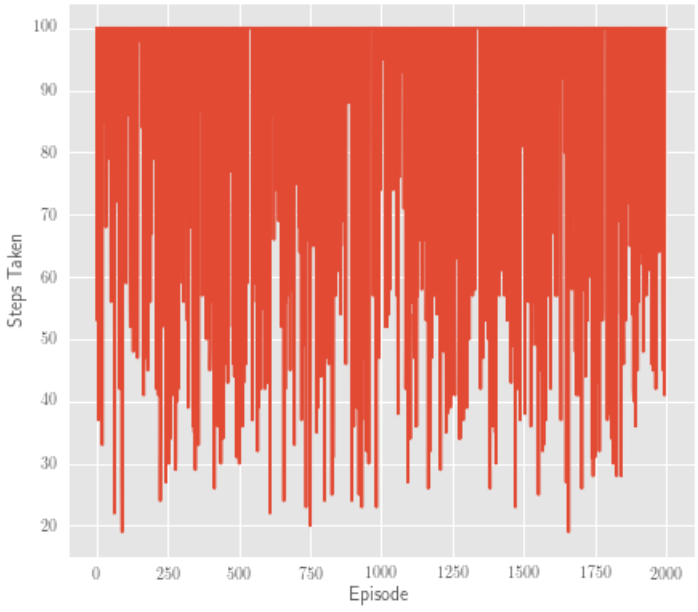


Best Plots

The plots for best set of Hyper Parameters found using Wandb analysis and also supported by variations in above section.

Reward Curve





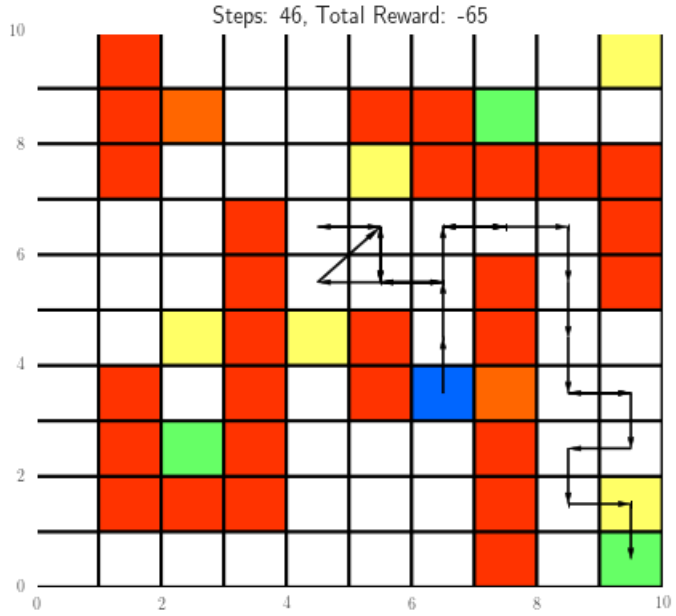
Best plots for:

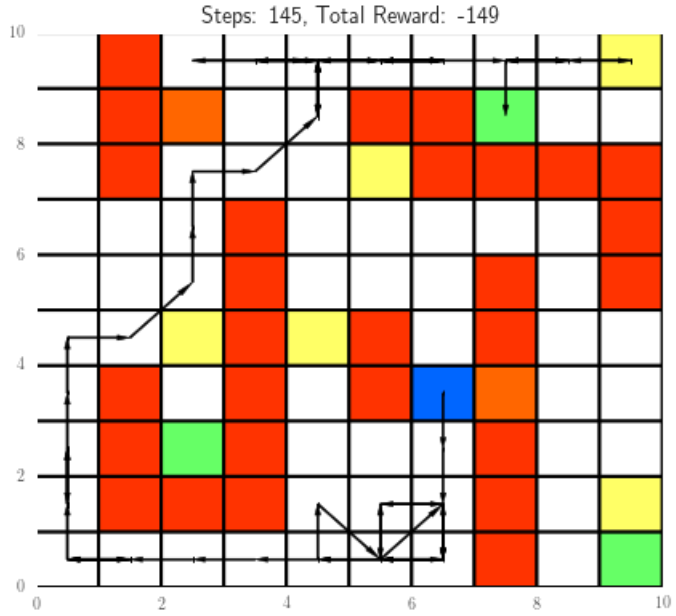
- Algorithm - q-learning
- Policy - softmax
- Beta - 1.478
- Alpha - 0.02983
- Gamma - 0.85

Final Learned Policy

Compared to the previous configuration, only the p value is further reduced to 0.35. But most of the inferences made previously apply here as well.The simulated results indicate that the agent has also learnt paths to other goal-states. The same could be inferred from increased and varied number of steps to the goal.

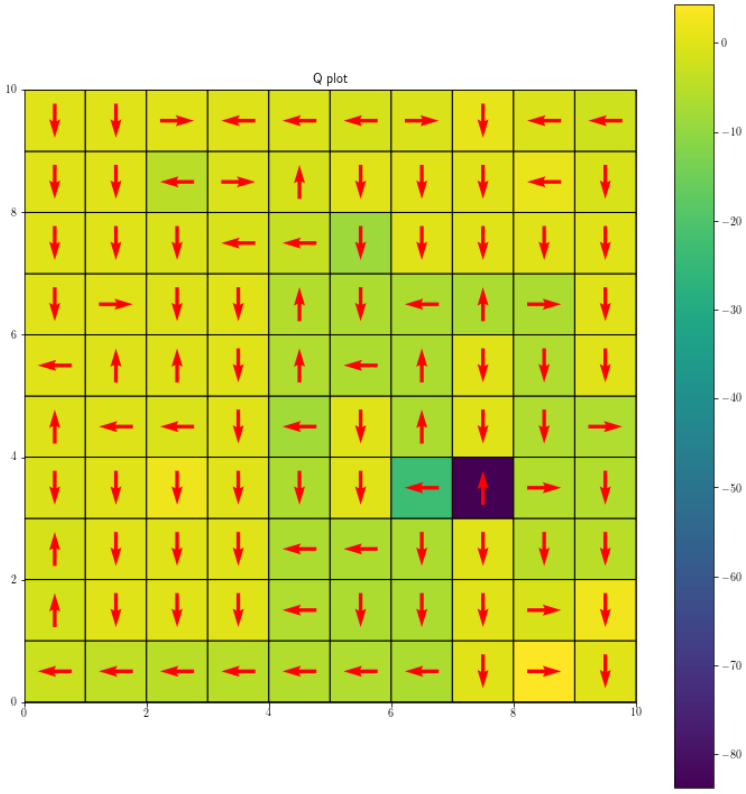
The trained agent is made to explore the environment for 2 runs. The visited states and actions are plotted


Steps: 46, Total Reward: -65
10
8
6
4
2
0
0
2
4
6
8
10


Steps: 145, Total Reward: -149
10
8
6
4
2
0
0
2
4
6
8
10

HeatMap

Heatmap of Grid World with Q values and optimal actions (of final policy learnt)

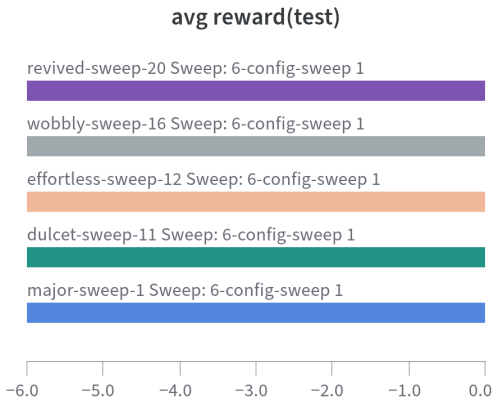


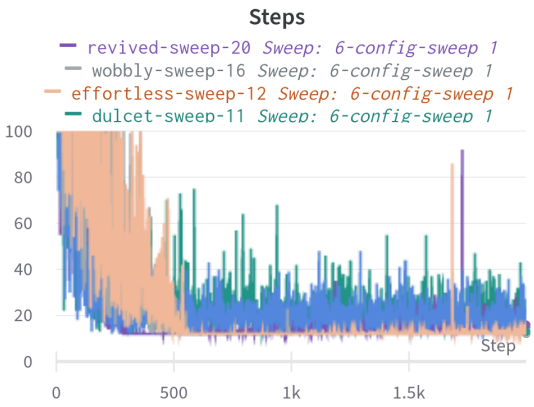
Configuration 6

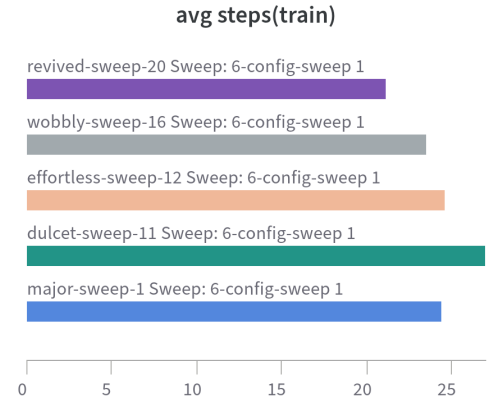
Configuration parameters

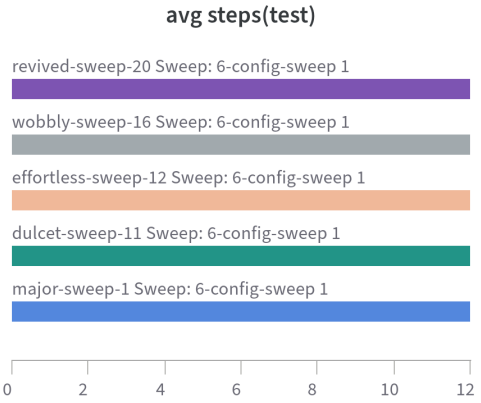
Wind = **False**, Start State = **[0,4]**, p = **1.0**

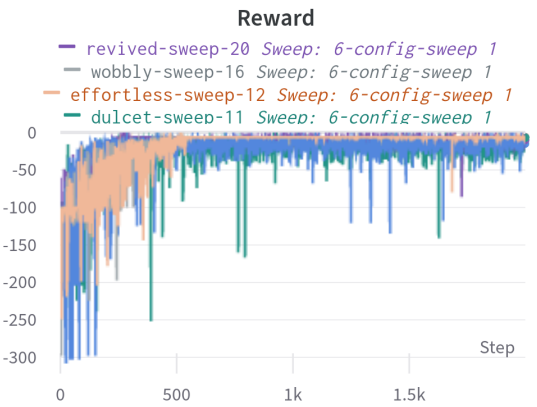
Wandb Analysis

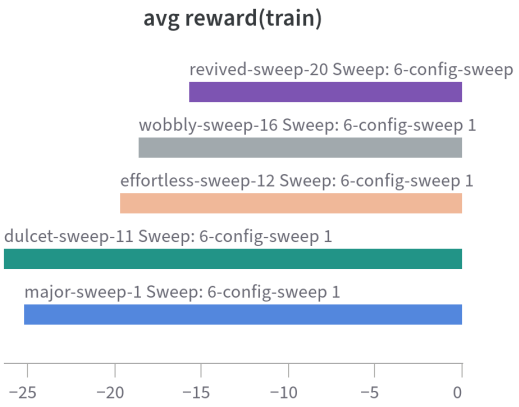


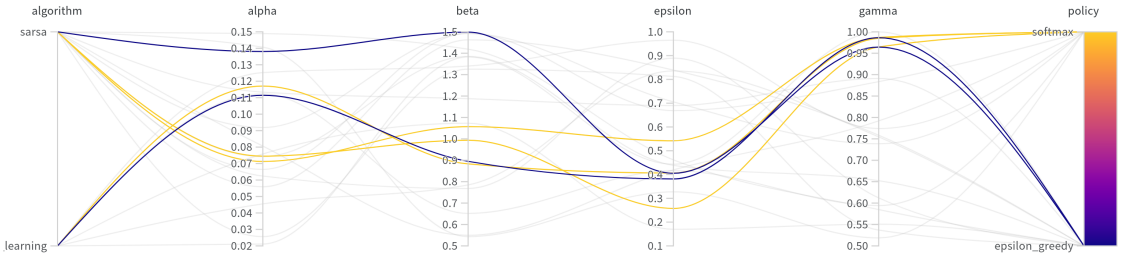










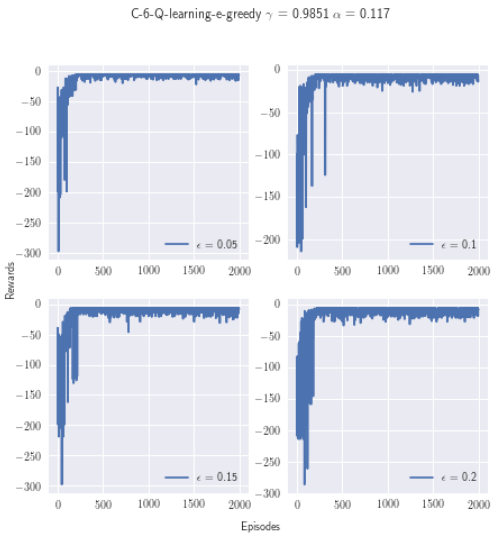


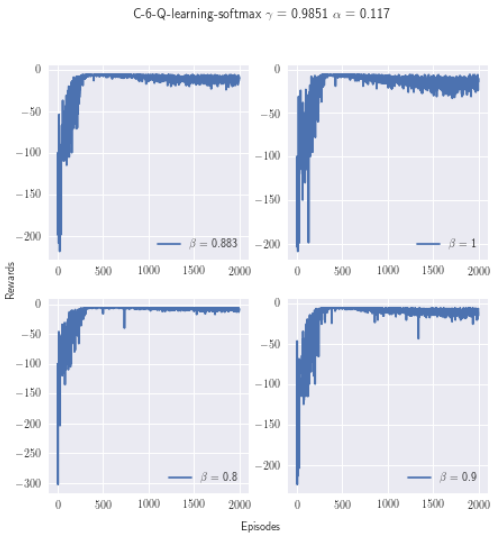
Hyper-parameter Plots

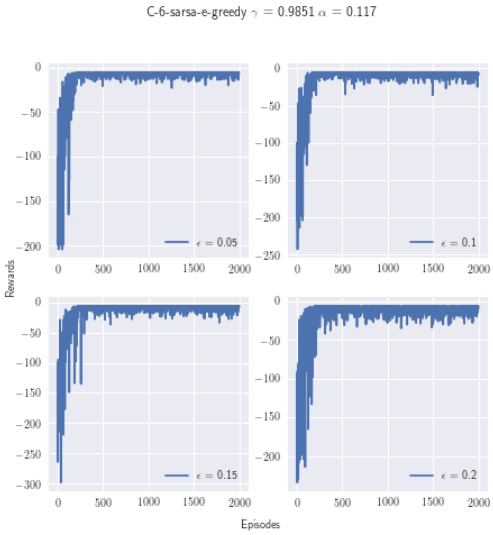
Fixing the best Hyper parameters found in the above section, other values are tried to in order to verify the observations.

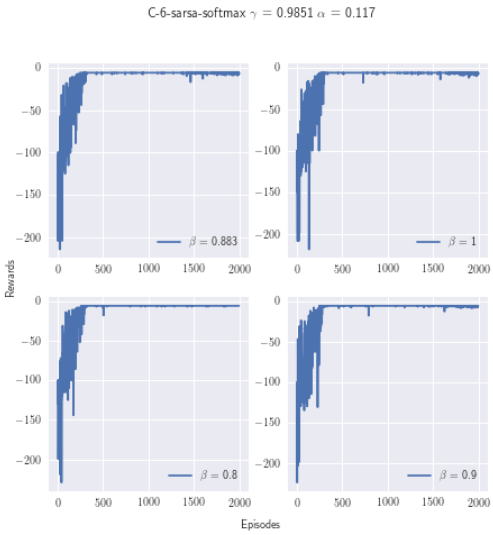
Policy Greed Variations

In this for each Algorithm and policy, α and γ are fixed and the policy greed i.e., ϵ or β , depending on the policy, is varied.





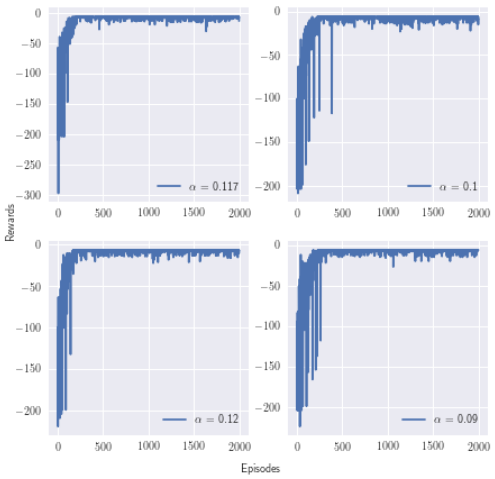




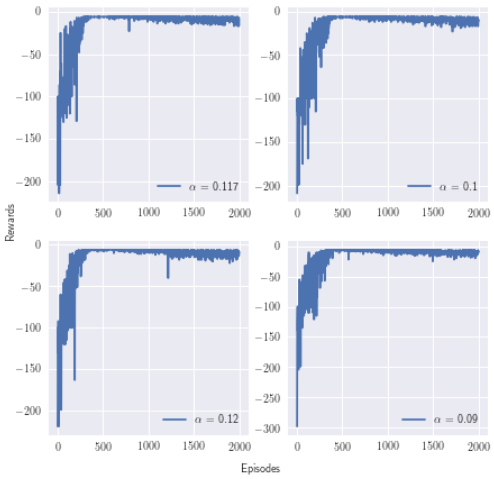
Learning Rate Variations

In this for each Algorithm and policy, policy greed and γ are fixed and Learning Rate α is varied.

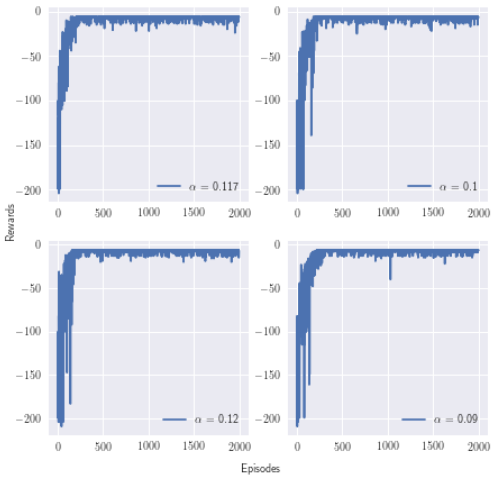
C-6-Q-learning-e-greedy $\gamma = 0.9851$ $/\epsilon = 0.05$



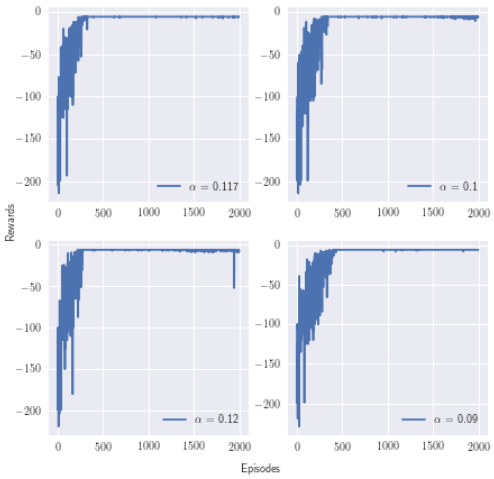
C-6-Q-learning-softmax $\gamma = 0.9851$ $\beta = 0.883$

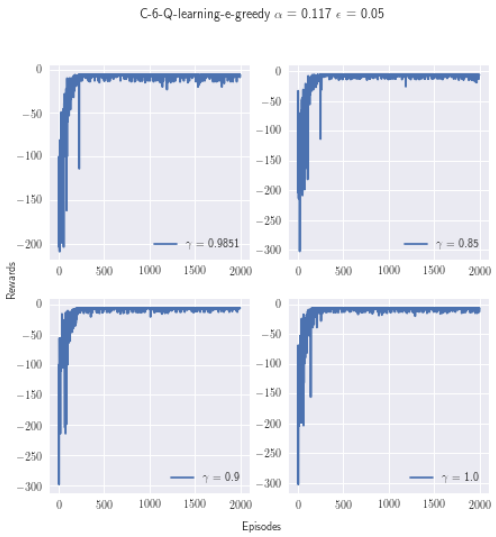


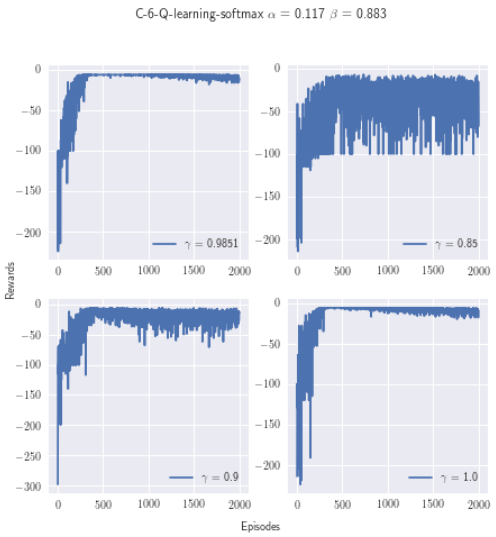
C-6-sarsa-e-greedy $\gamma = 0.9851$ $/\epsilon = 0.05$

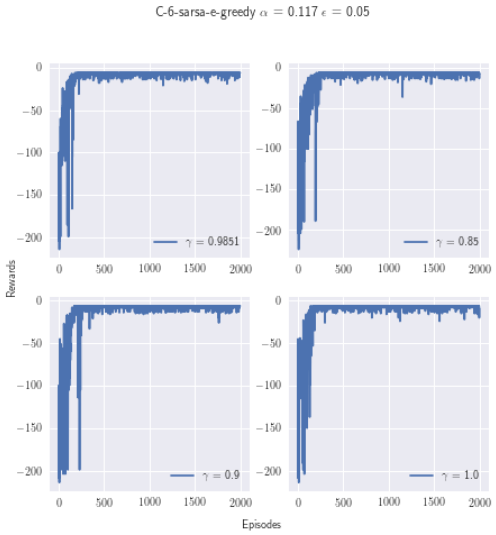


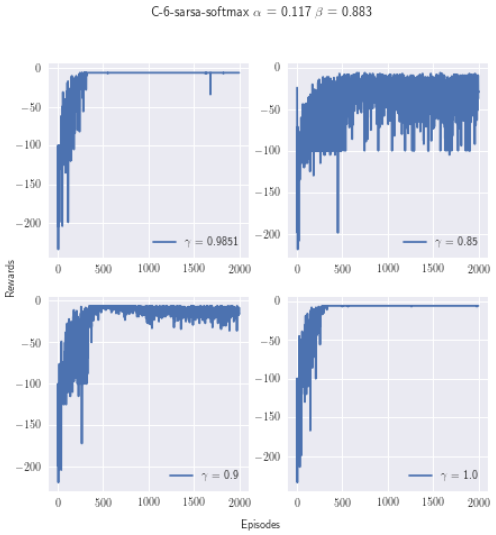
C-6-sarsa-softmax $\gamma = 0.9851$ $\beta = 0.883$







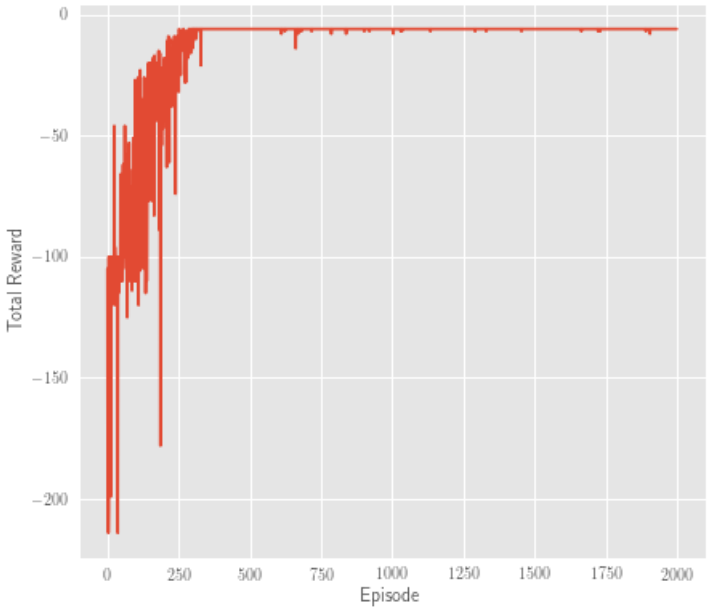


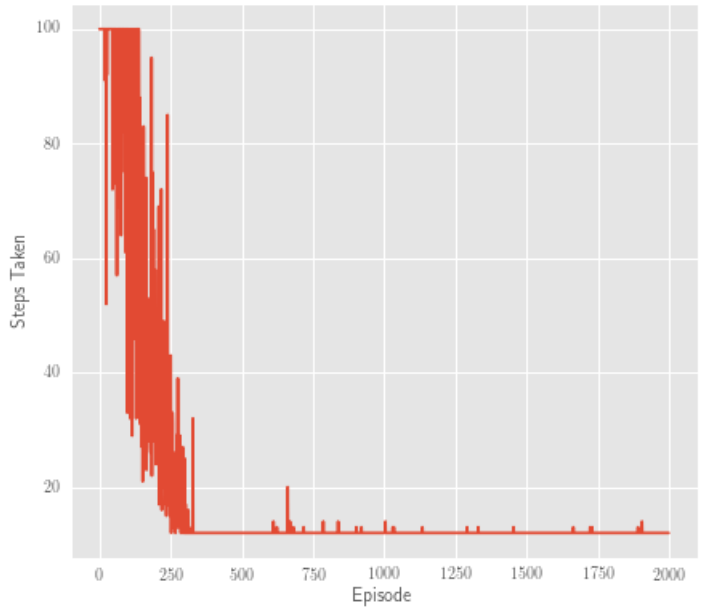


Best Plots

The plots for best set of Hyper Parameters found using Wandb analysis and also supported by variations in above section.

Reward Curve and no of steps to reach goal





Best plots for:

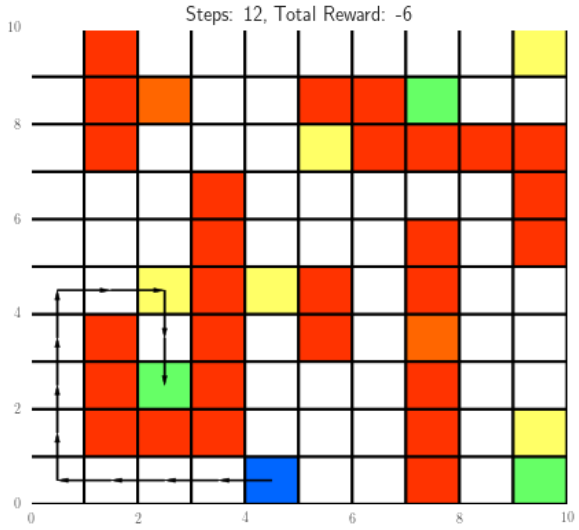
- Algorithm - Sarsa
- Policy - softmax
- Beta - 0.883
- Alpha - 0.09
- Gamma - 0.9851

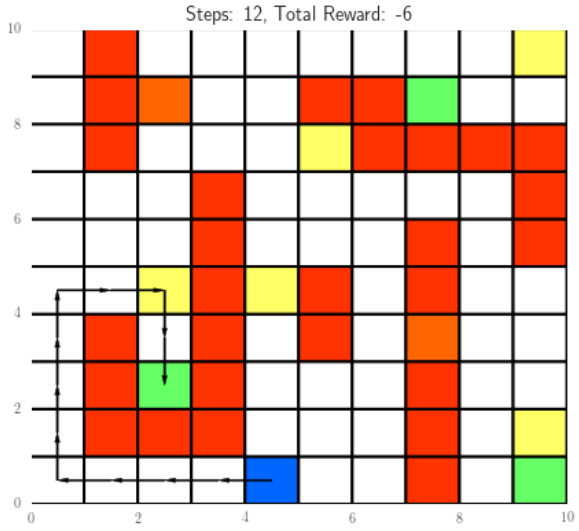
In fact there are multiple sets of hyper-parameters which are optimal, this is just one of them.

Final Learned Policy

In this case, the agent eventually follows the shortest path maximising reward from the start state to goal state(even evident in the below two runs) in 12 steps with reward=-6, because there isn't any stochasticity with respect to transitions from one state to another i.e. there is no wind and p=1.

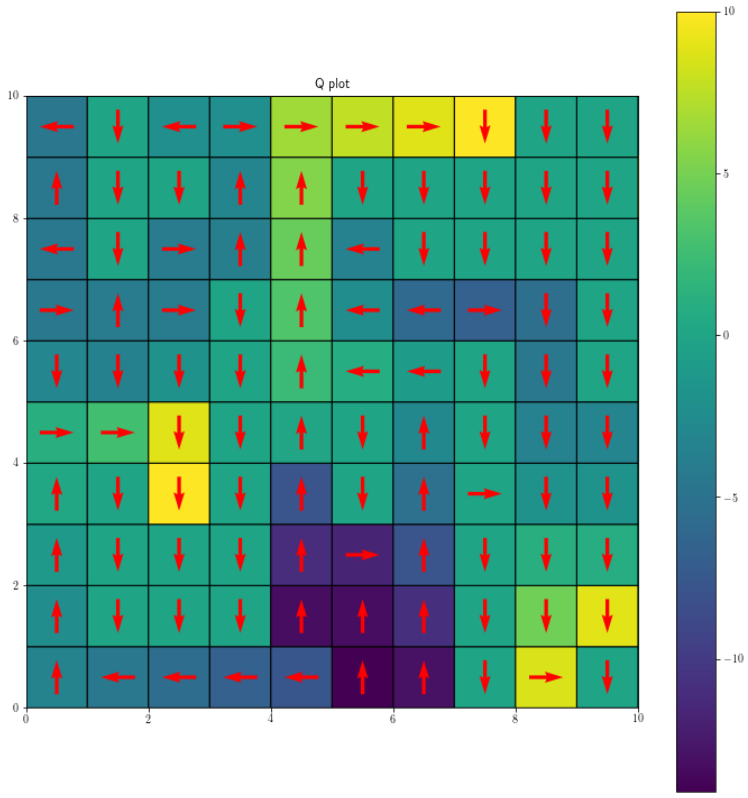
The trained agent is made to explore the environment for 2 runs. The visited states and actions are plotted





HeatMap

Heatmap of Grid World with Q values and optimal actions (of final policy learnt)

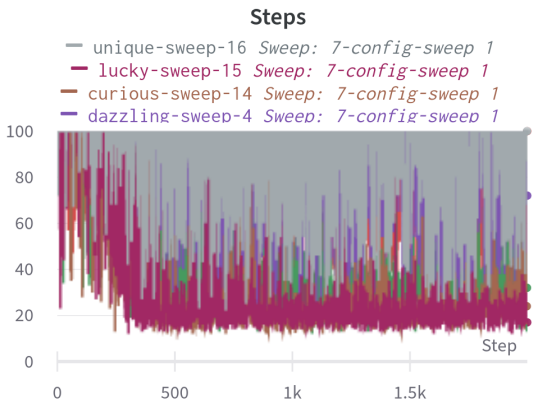

Q plot
0
2
4
6
8
10
10
5
0
−5
−10

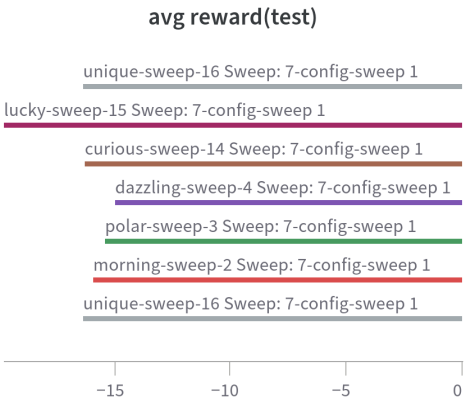
Configuration 7

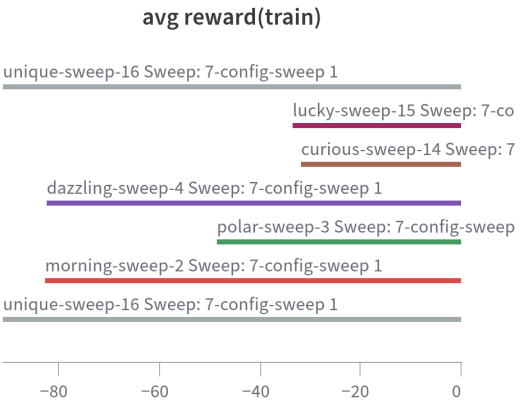
Configuration parameters

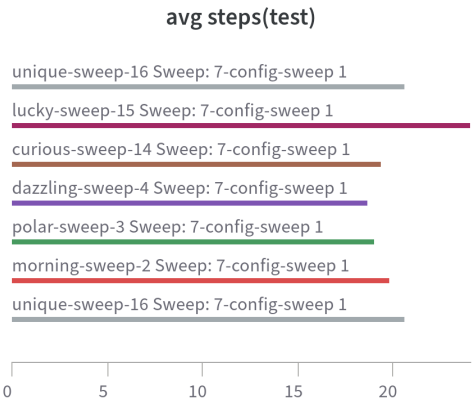
Wind = **False**, Start State = **[0,4]**, p = **0.7**

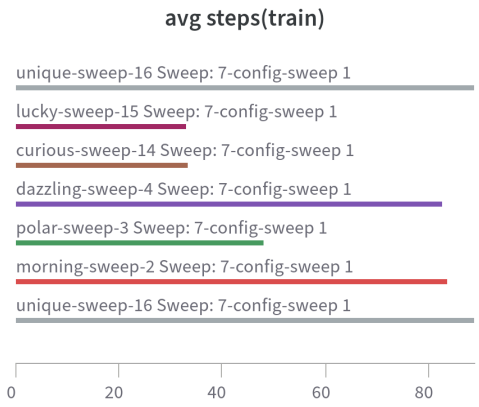
Wandb Analysis

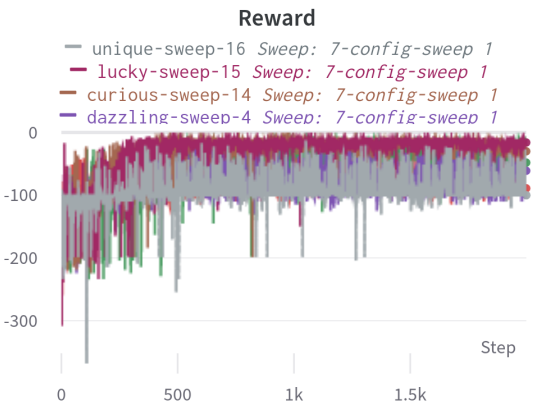


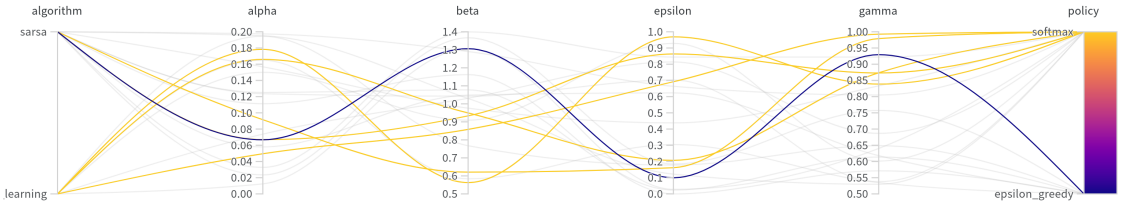










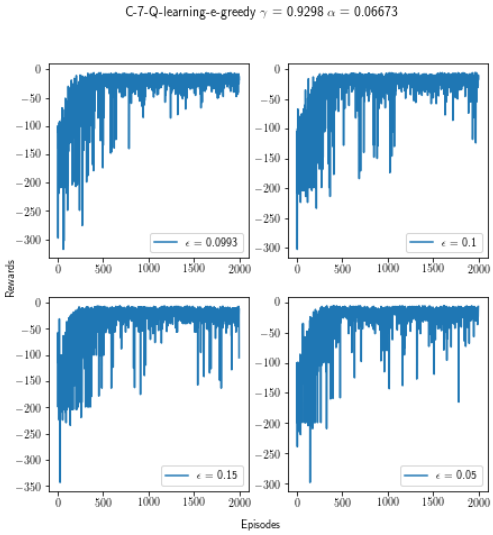


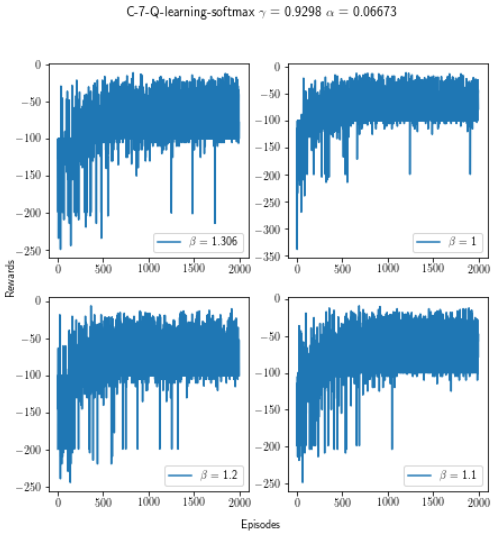
Hyper-parameter Plots

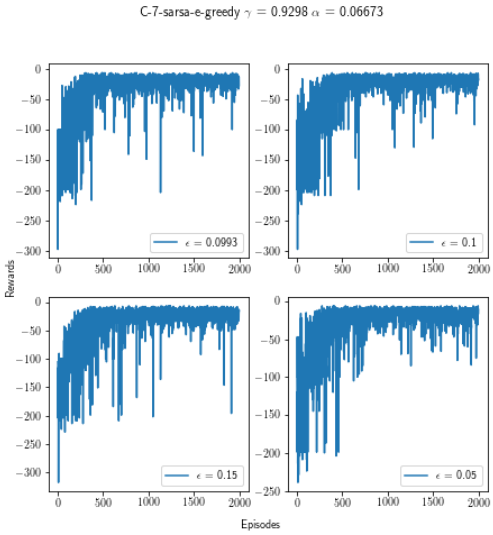
Fixing the best Hyper parameters found in the above section, other values are tried to in order to verify the observations.

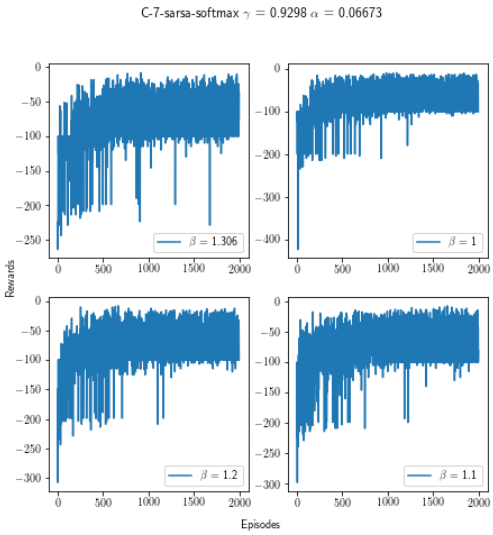
Policy Greed Variations

In this for each Algorithm and policy, α and γ are fixed and the policy greed i.e., ϵ or β , depending on the policy, is varied.





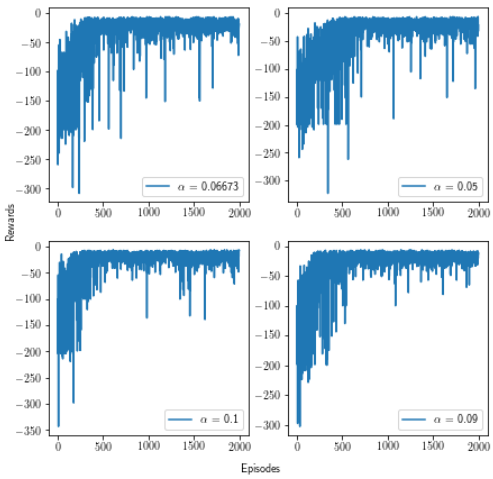




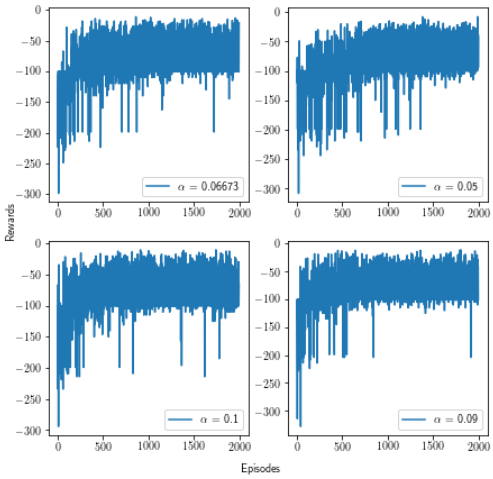
Learning Rate Variations

In this for each Algorithm and policy, policy greed and γ are fixed and Learning Rate α is varied.

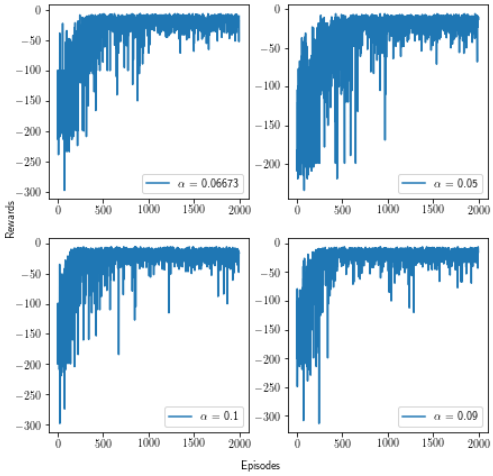
C-7-Q-learning-e-greedy $\gamma = 0.9298$ /$\epsilon = 0.0993$



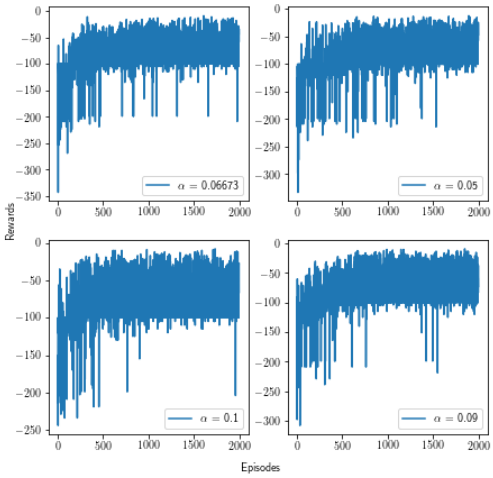
C-7-Q-learning-softmax $\gamma = 0.9298$ $\beta = 1.306$



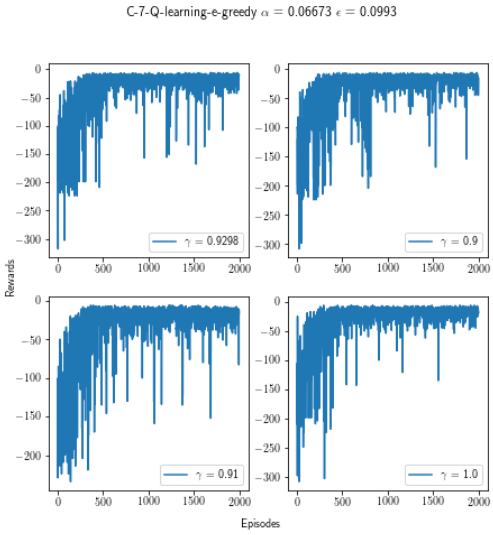
C-7-sarsa-e-greedy $\gamma = 0.9298$ /$\epsilon = 0.0993$

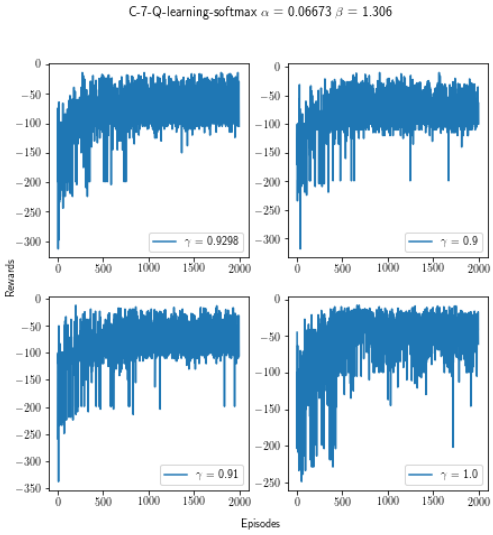


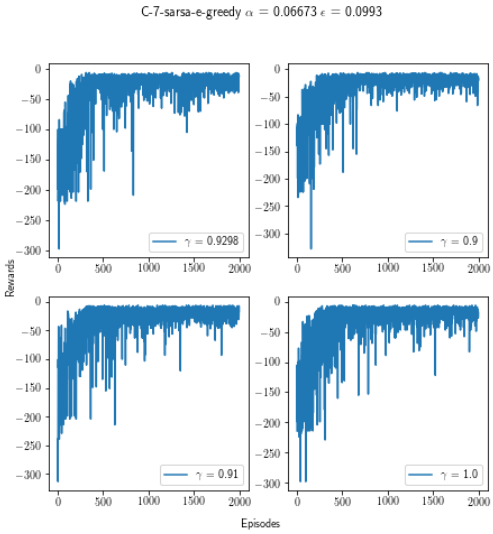
C-7-sarsa-softmax $\gamma = 0.9298$ $\beta = 1.306$

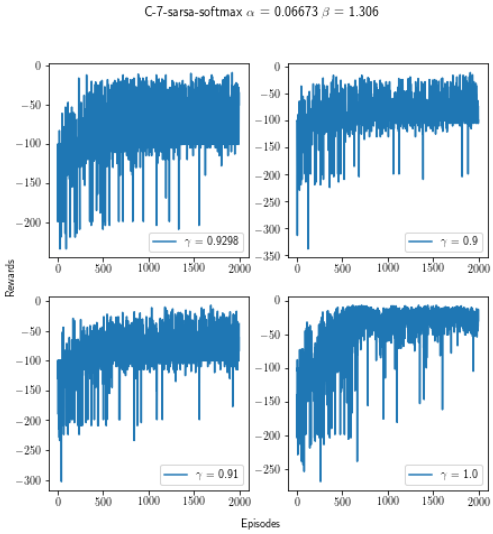


Discount Rate Variations





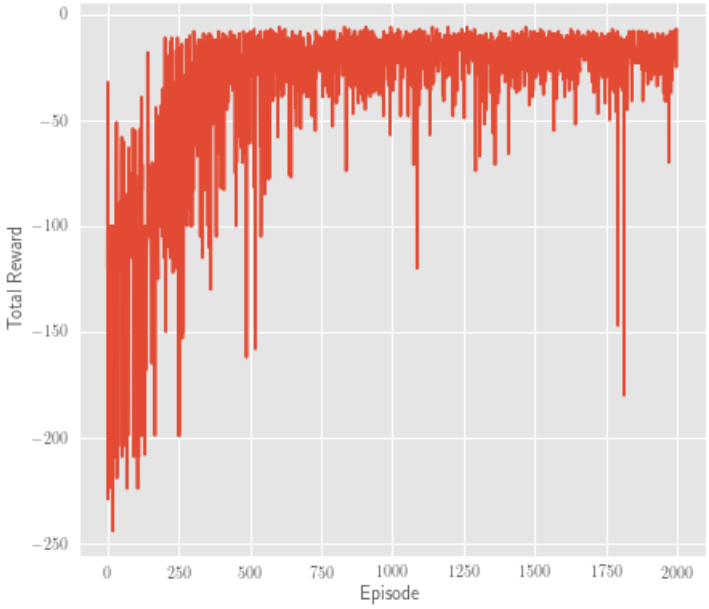


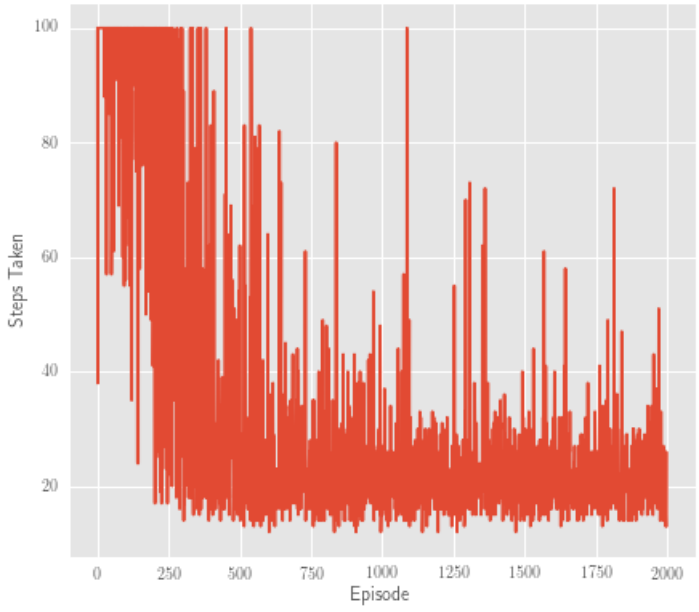


Best Plots

The plots for best set of Hyper Parameters found using Wandb analysis and also supported by variations in above section.

Reward Curve



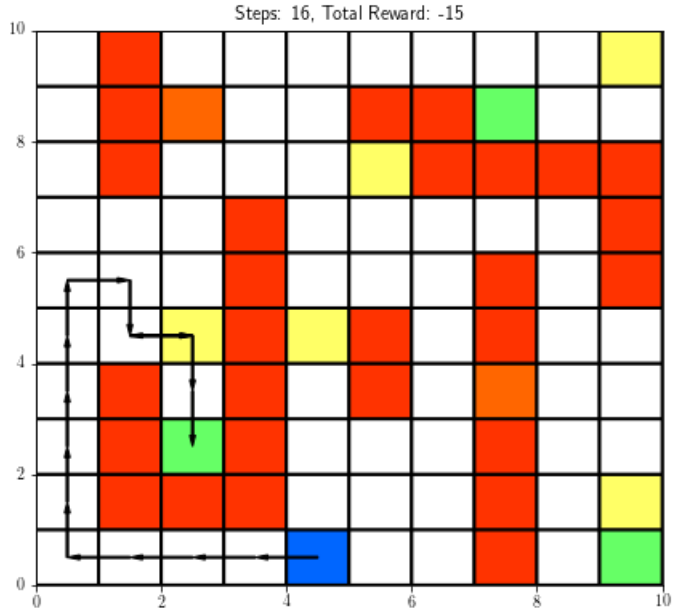


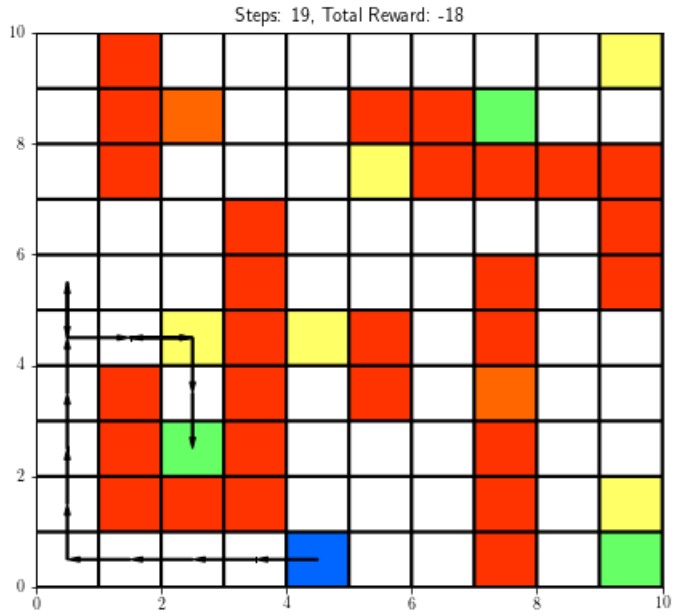
Best plots for:

- Algorithm - Sarsa
- Policy - e-greedy
- Epsilon - 0.0993
- Alpha - 0.06673
- Gamma - 0.9615

Final Learned Policy

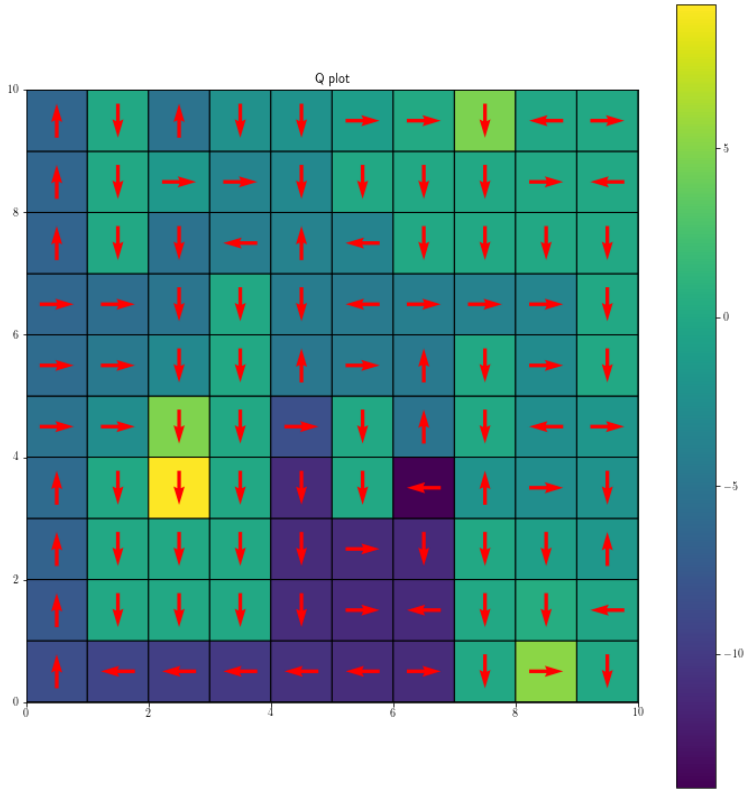
The scenario is similar to config-1 except that there is no wind, but we would still have variance in rewards, number of steps because p=0.7; it still adds stochasticity to the transitions, leading to slightly off-beat paths.
The trained agent is made to explore the environment for 2 runs. The visited states and actions are plotted


Steps: 16, Total Reward: -15
10
8
6
4
2
0
0
2
4
6
8
10


Steps: 19, Total Reward: -18
10
8
6
4
2
0
0
2
4
6
8
10

HeatMap

Heatmap of Grid World with Q values and optimal actions (of final policy learnt)

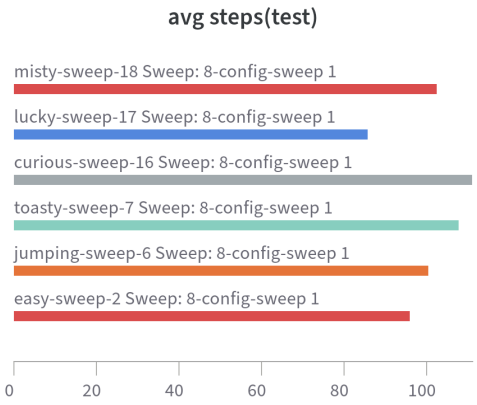


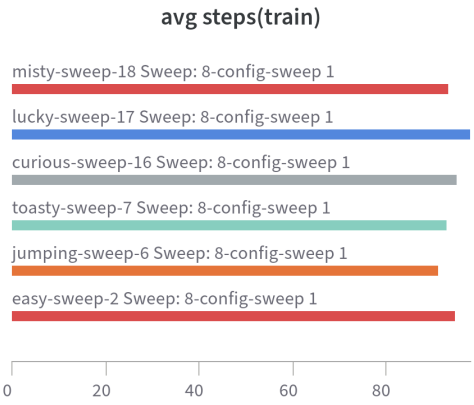
Configuration 8

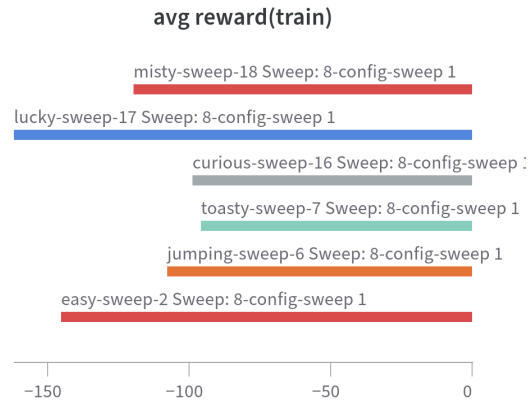
Configuration parameters

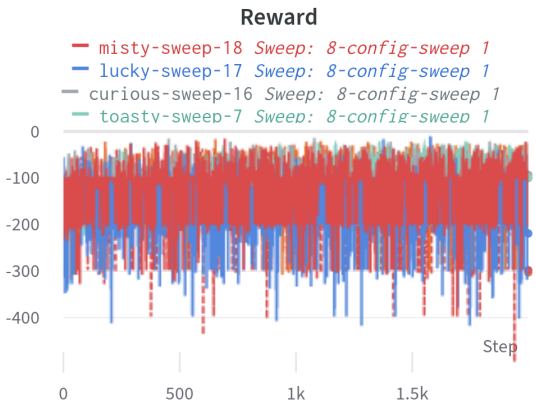
Wind = **False**, Start State = **[0,4]**, p = **0.35**

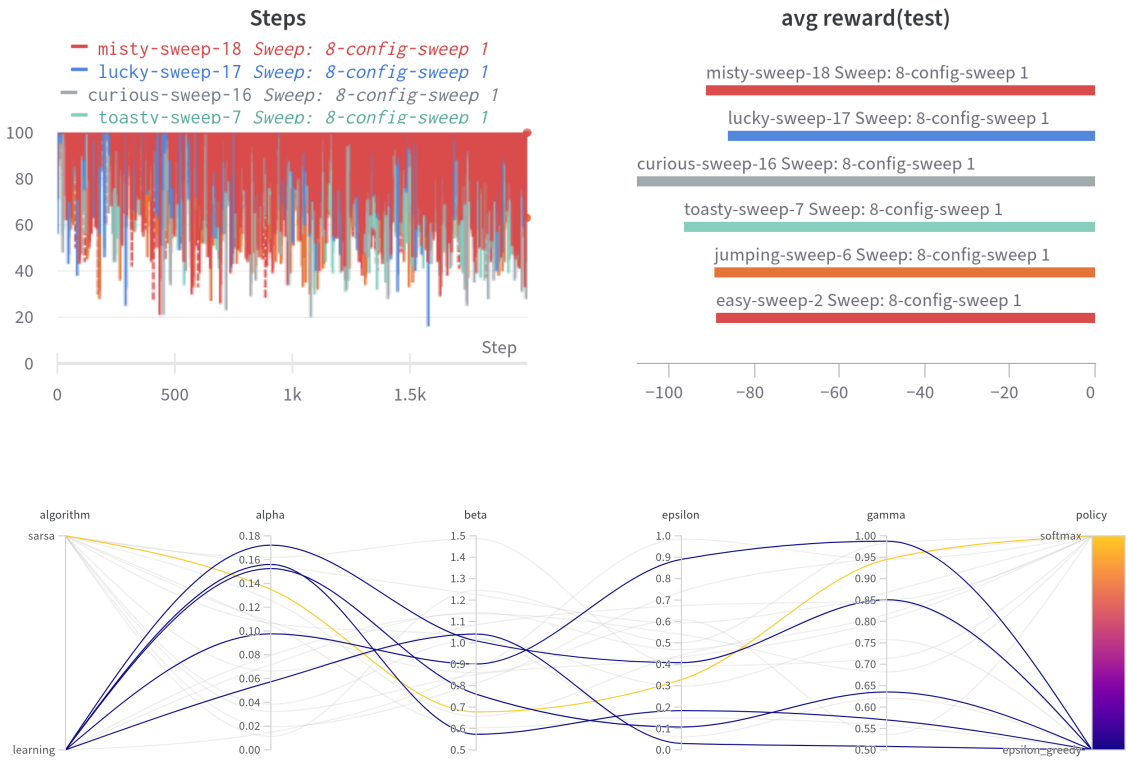
1 Wandb Analysis









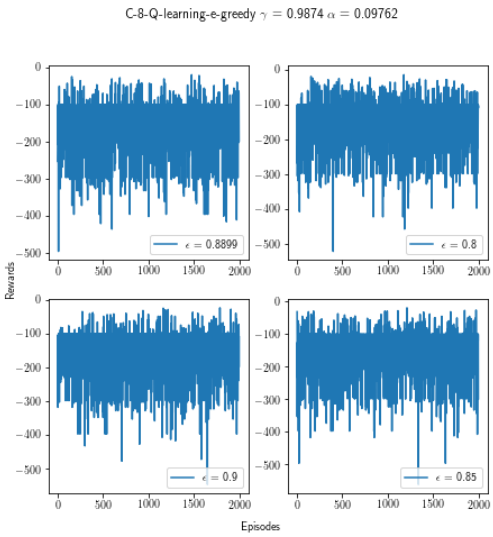


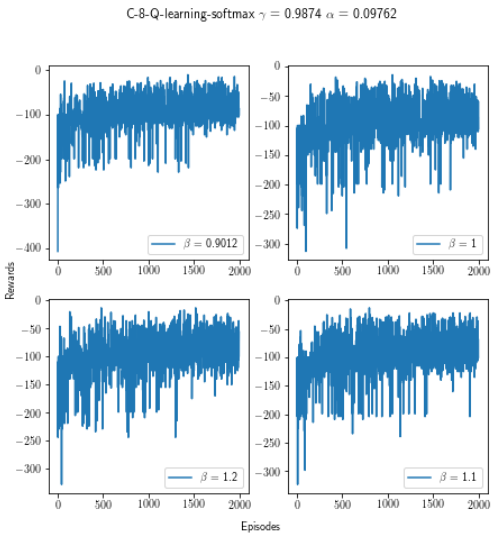
Hyper-parameter Plots

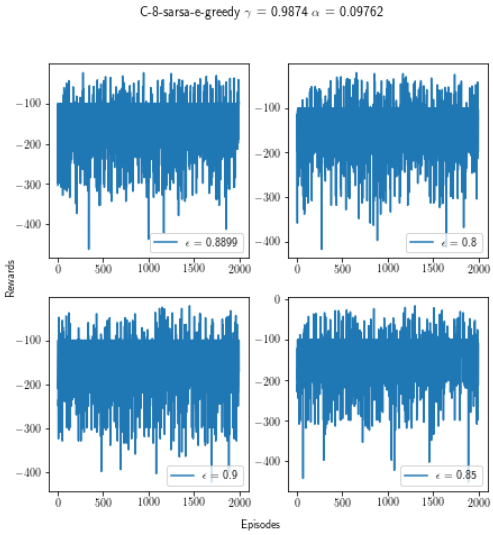
Fixing the best Hyper parameters found in the above section, other values are tried to in order to verify the observations.

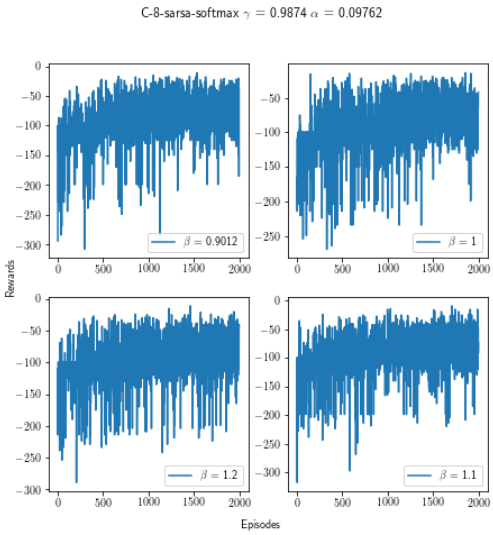
Policy Greed Variations

In this for each Algorithm and policy, α and γ are fixed and the policy greed i.e., ϵ or β , depending on the policy, is varied.





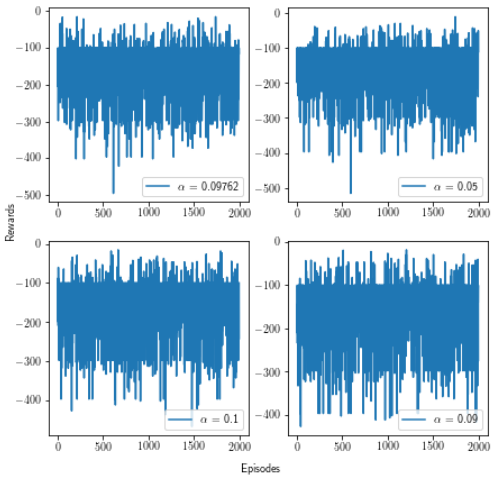




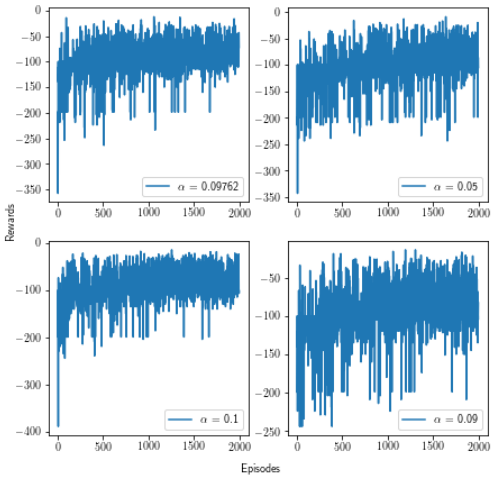
Learning Rate Variations

In this for each Algorithm and policy, policy greed and γ are fixed and Learning Rate α is varied.

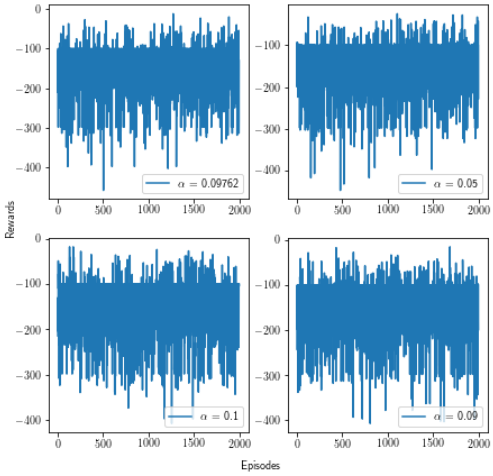
C-8-Q-learning-e-greedy $\gamma = 0.9874$ /$\epsilon = 0.8899$



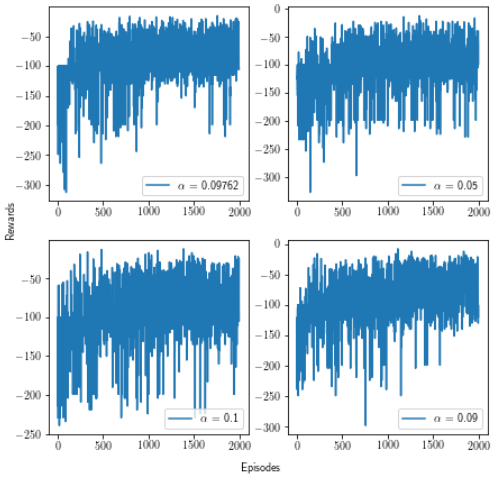
C-8-Q-learning-softmax $\gamma = 0.9874$ $\beta = 0.9012$



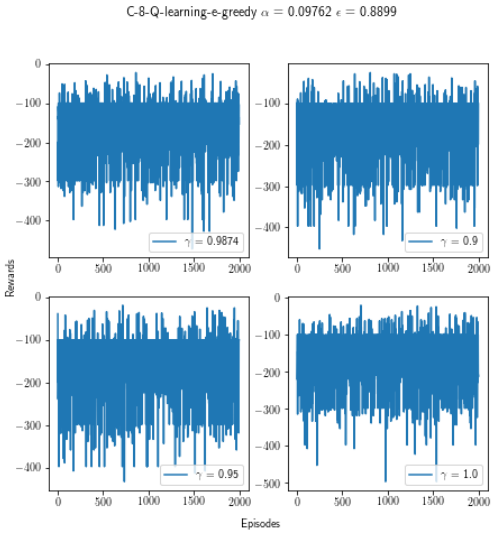
C-8-sarsa-e-greedy $\gamma = 0.9874$ /$\epsilon = 0.8899$

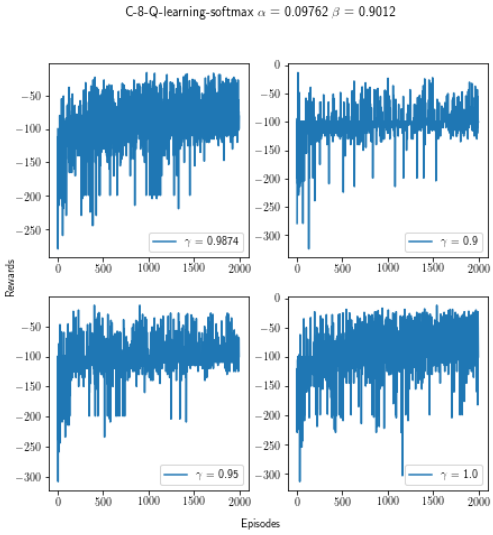


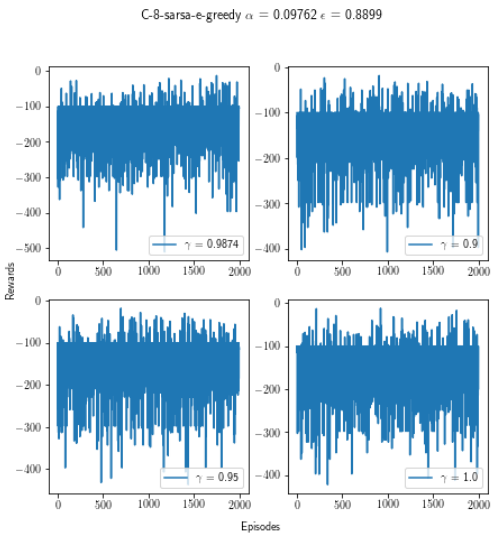
C-8-sarsa-softmax $\gamma = 0.9874$ $\beta = 0.9012$

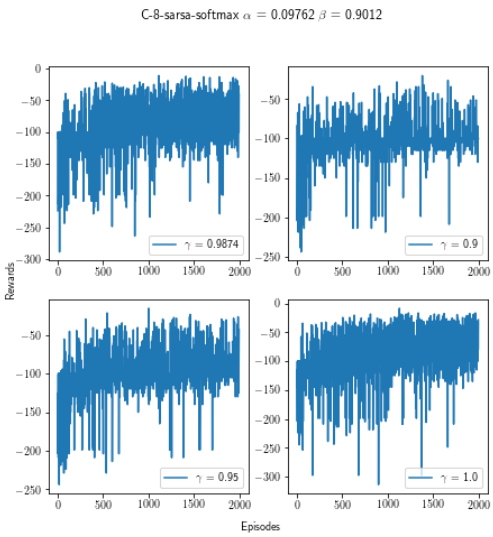


Discount Rate Variations





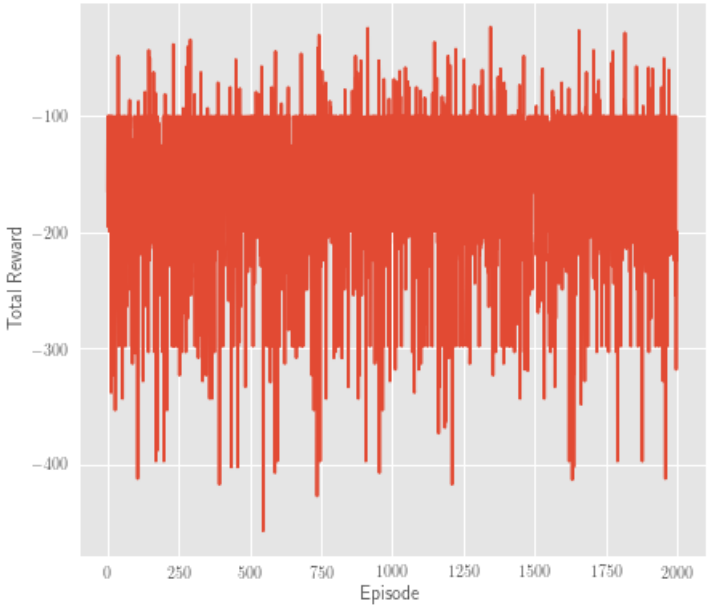


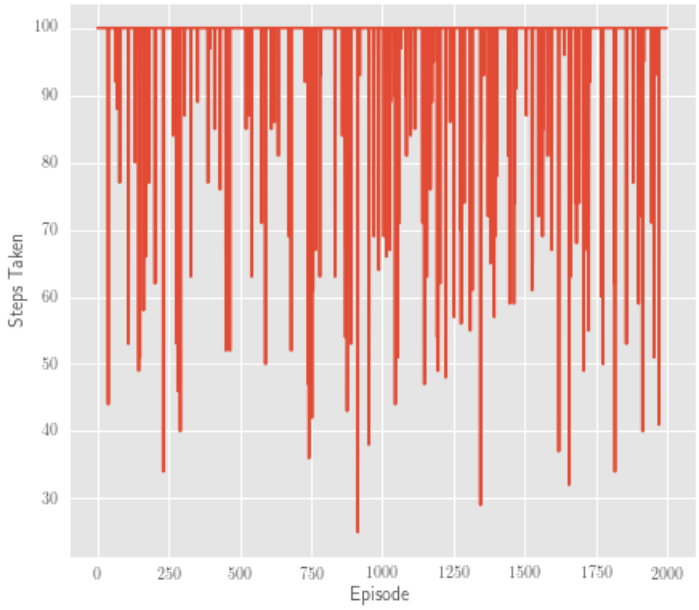


Best Plots

The plots for best set of Hyper Parameters found using Wandb analysis and also supported by variations in above section.

Reward Curve



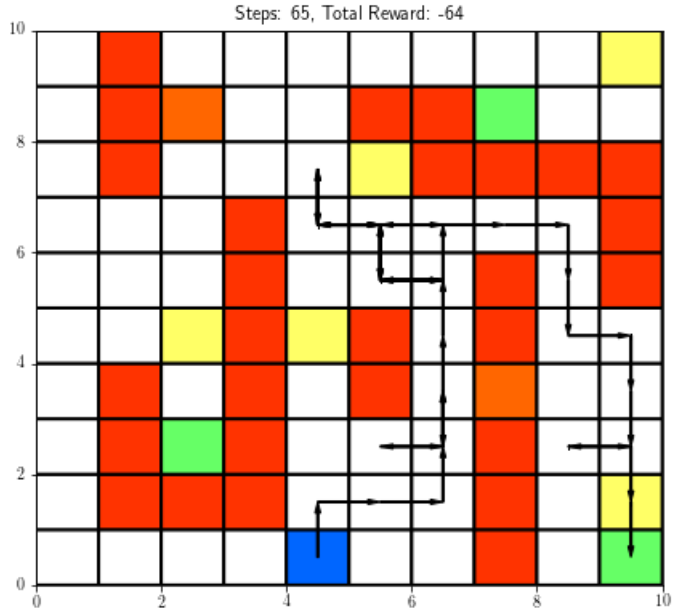


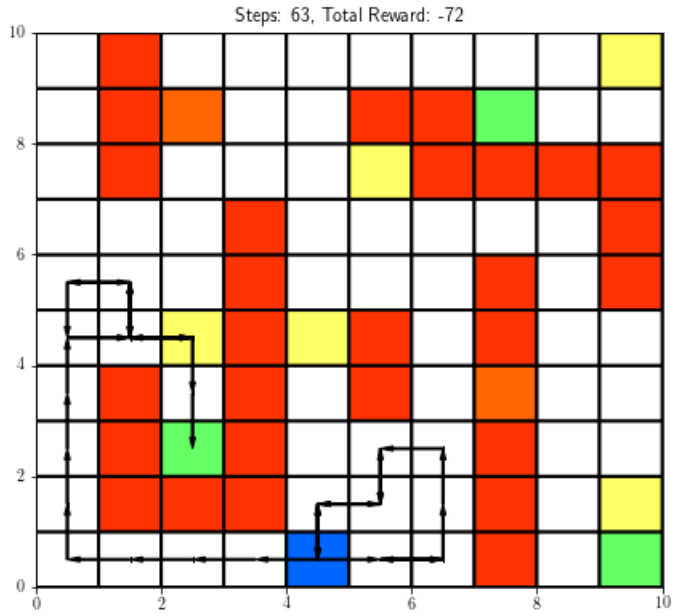
Best plots for:

- Algorithm - q-learning
- Policy - e-greedy
- Epsilon - 0.8899
- Alpha - 0.09762
- Gamma - 0.9615

Final Learned Policy

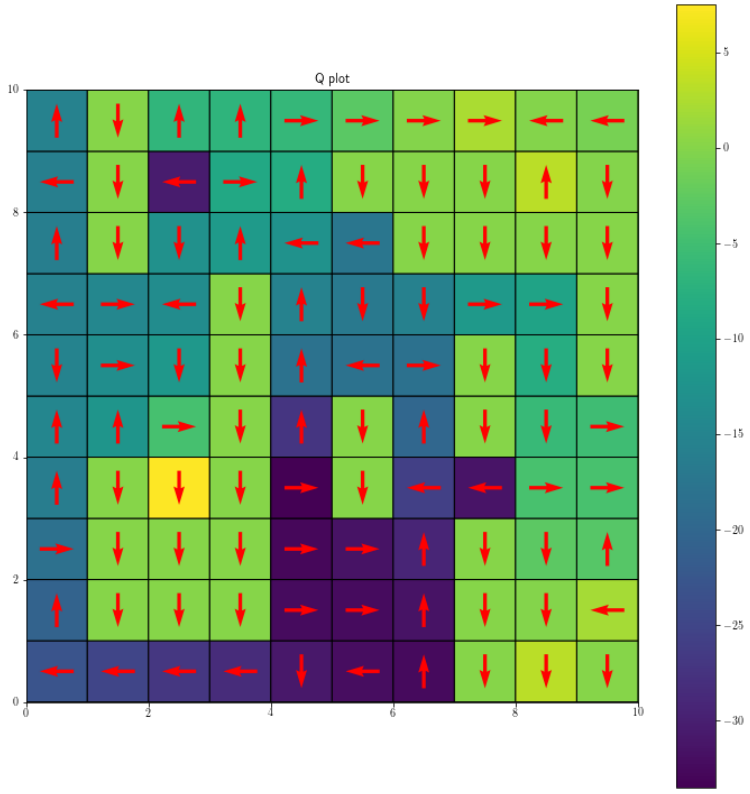
This case is pretty similar to the previous one(config-7) except that we'll have more variance in the rewards, number of steps to goal.
The trained agent is made to explore the environment for 2 runs. The visited states and actions are plotted


Steps: 65, Total Reward: -64
0
2
4
6
8
10


Steps: 63, Total Reward: -72
0
2
4
6
8
10

HeatMap

Heatmap of Grid World with Q values and optimal actions (of final policy learnt)



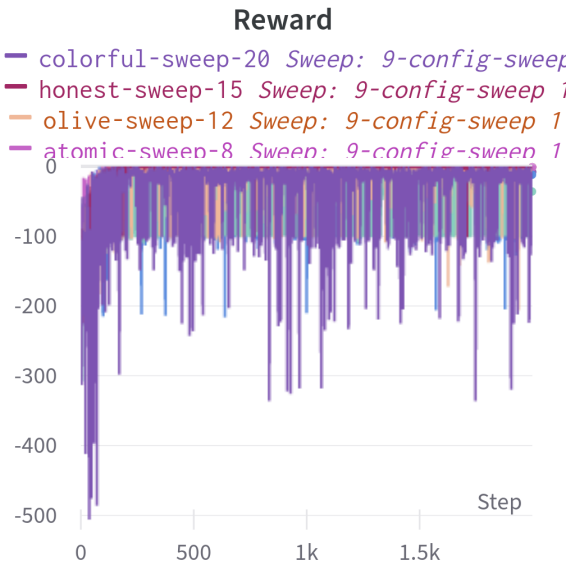
The previous three configurations(config-6,7,8) are very similar to config-(1,2,3), now that we don't even have wind. This has a consequence in config-6 where the agent always follows the optimal path.

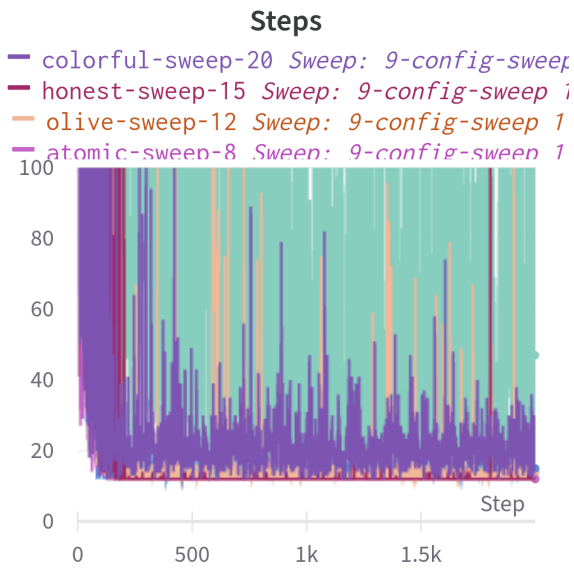
Configuration 9

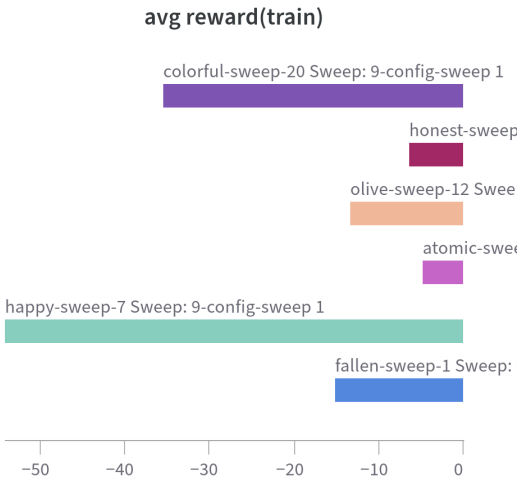
Configuration parameters

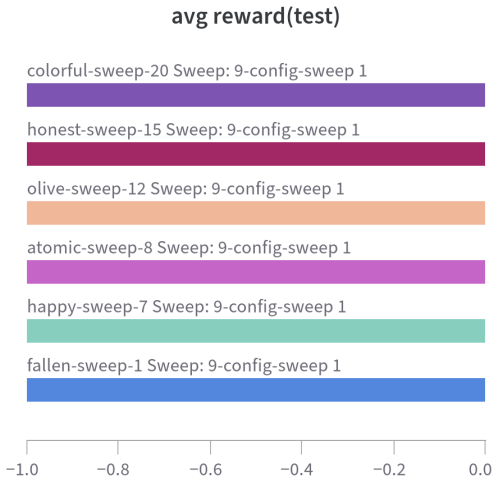
Wind = **False**, Start State = **[3,6]**, p = **1.0**

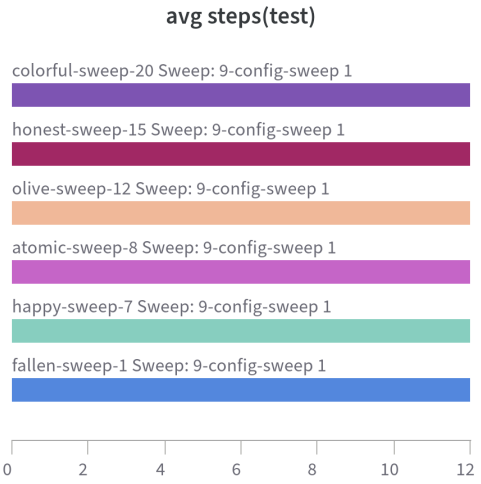
Wandb Analysis

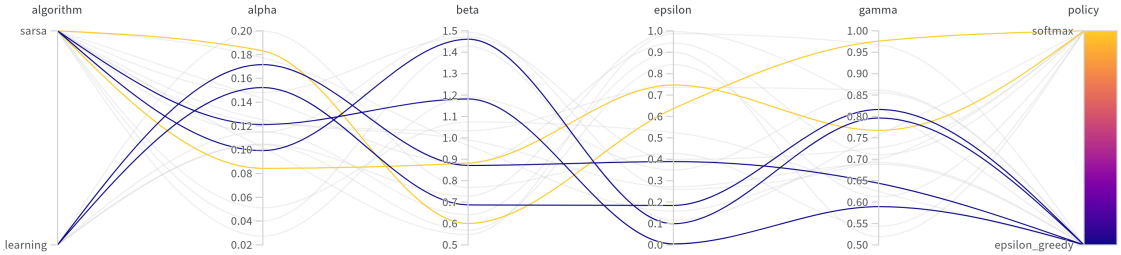



Steps
colorful-sweep-20 Sweep: 9-config-sweep
honest-sweep-15 Sweep: 9-config-sweep
olive-sweep-12 Sweep: 9-config-sweep 1
atomic-sweep-8 Sweep: 9-config-sweep 1
100
80
60
40
20
0
Step
0
500
1k
1.5k


avg reward(train)
colorful-sweep-20 Sweep: 9-config-sweep 1
honest-sweep
olive-sweep-12 Swee
atomic-swe
happy-sweep-7 Sweep: 9-config-sweep 1
fallen-sweep-1 Sweep:
−50
−40
−30
−20
−10
0


avg reward(test)
colorful-sweep-20 Sweep: 9-config-sweep 1
honest-sweep-15 Sweep: 9-config-sweep 1
olive-sweep-12 Sweep: 9-config-sweep 1
atomic-sweep-8 Sweep: 9-config-sweep 1
happy-sweep-7 Sweep: 9-config-sweep 1
fallen-sweep-1 Sweep: 9-config-sweep 1
−1.0
−0.8
−0.6
−0.4
−0.2
0.0


avg steps(test)
colorful-sweep-20 Sweep: 9-config-sweep 1
honest-sweep-15 Sweep: 9-config-sweep 1
olive-sweep-12 Sweep: 9-config-sweep 1
atomic-sweep-8 Sweep: 9-config-sweep 1
happy-sweep-7 Sweep: 9-config-sweep 1
fallen-sweep-1 Sweep: 9-config-sweep 1
0
2
4
6
8
10
12

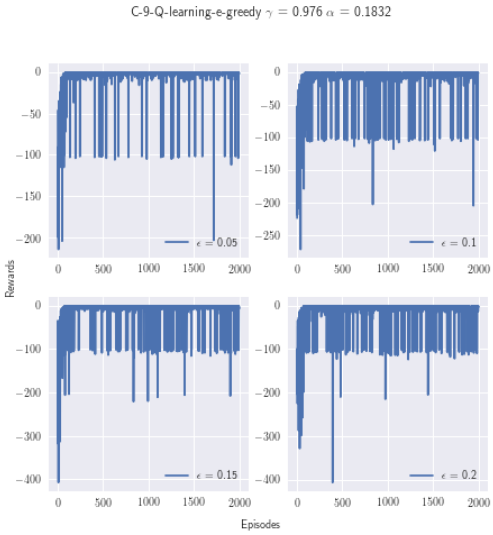

algorithm
sarsa
learning
alpha
beta
epsilon
gamma
policy
softmax
epsilon_greedy

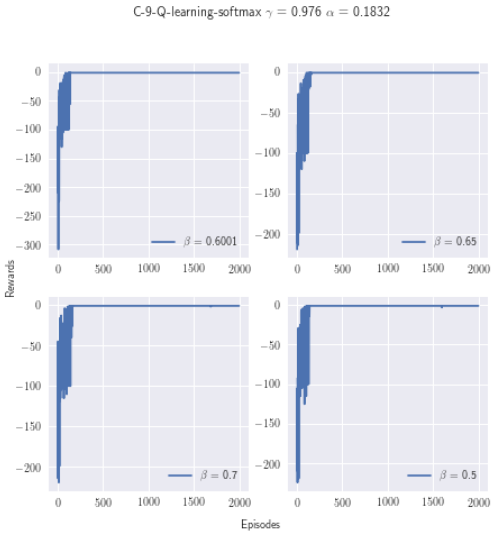
Hyper-parameter Plots

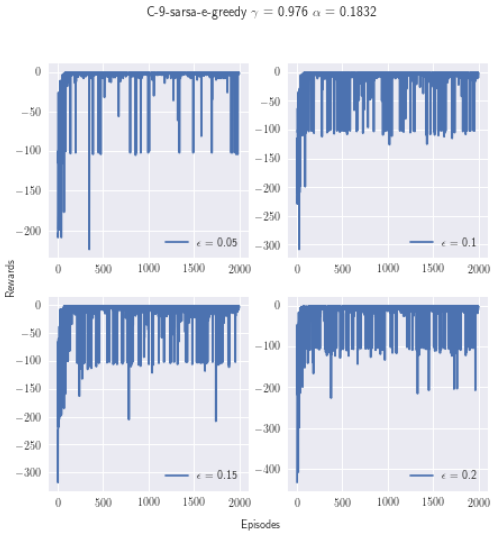
Fixing the best Hyper parameters found in the above section, other values are tried to in order to verify the observations.

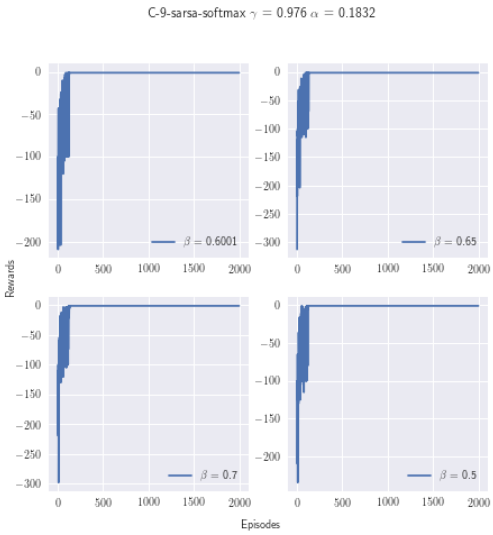
Policy Greed Variations

In this for each Algorithm and policy, α and γ are fixed and the policy greed i.e., ϵ or β , depending on the policy, is varied.



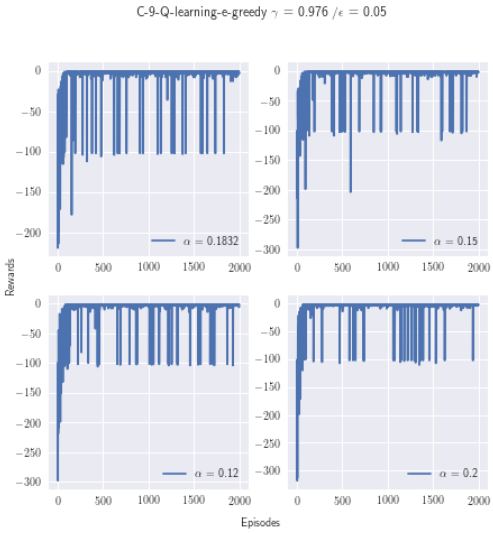


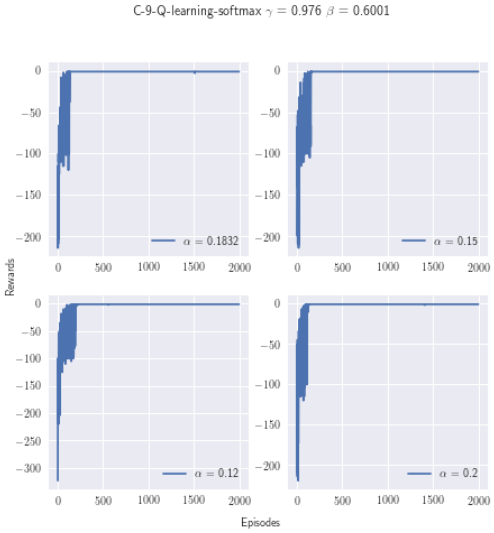


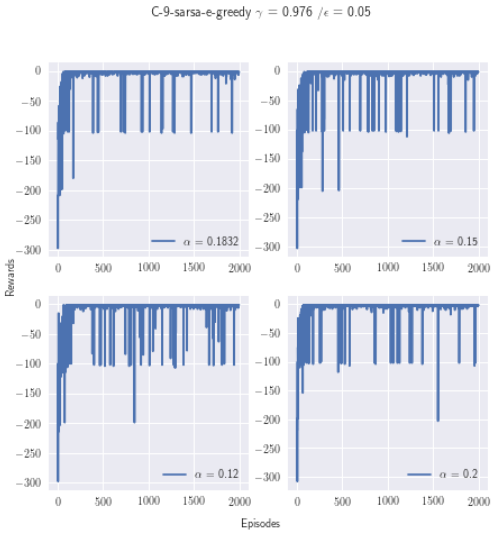


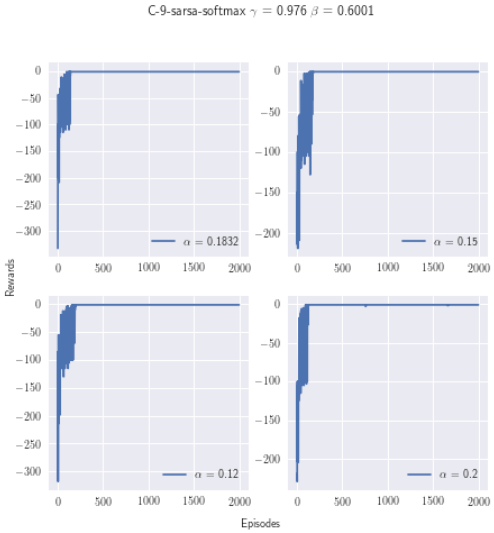
Learning Rate Variations

In this for each Algorithm and policy, policy greed and γ are fixed and Learning Rate α is varied.

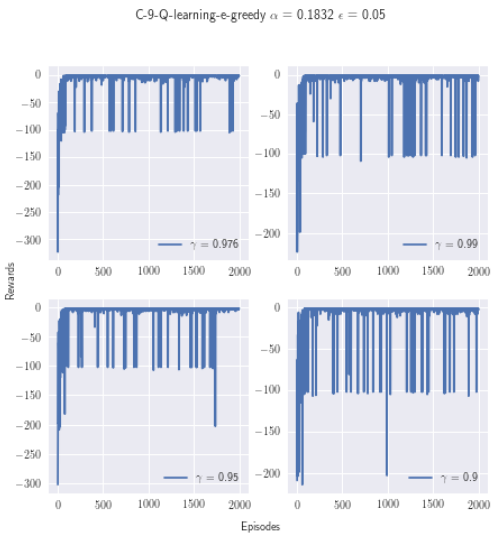


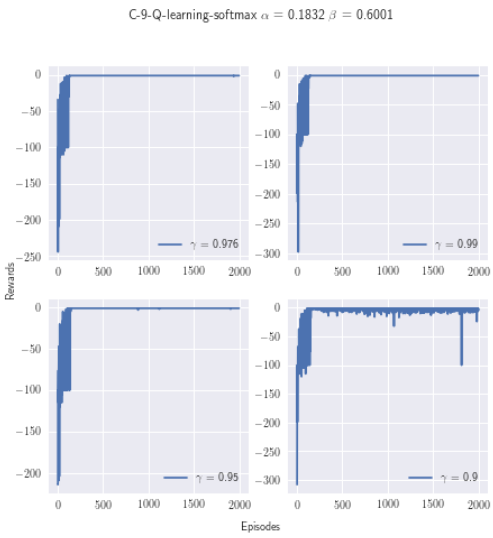


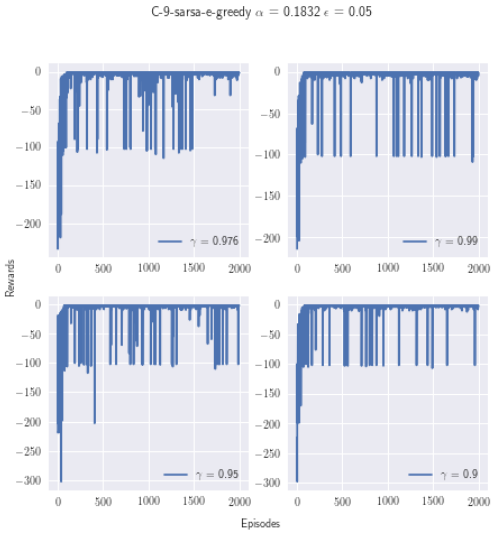


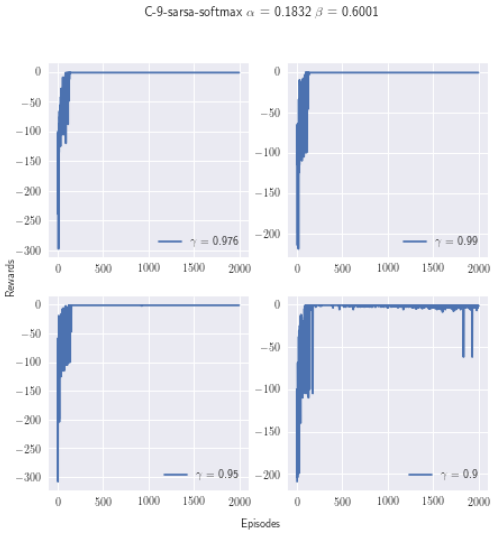


Discount Rate Variations





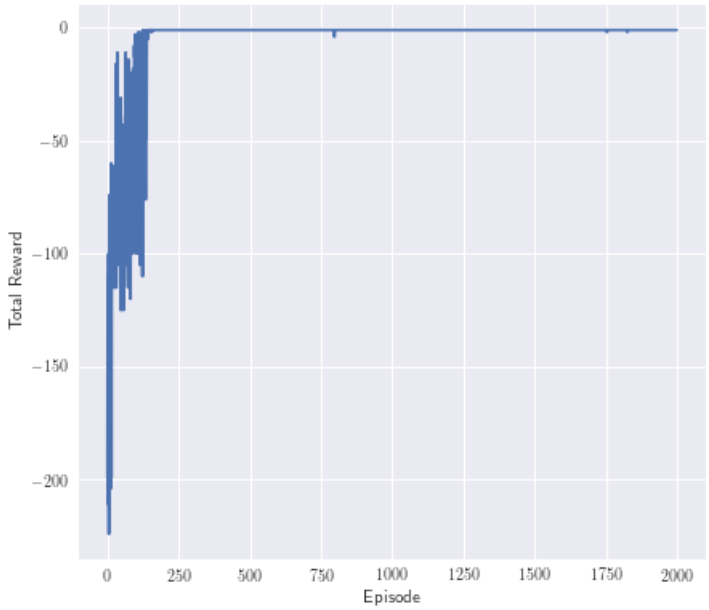


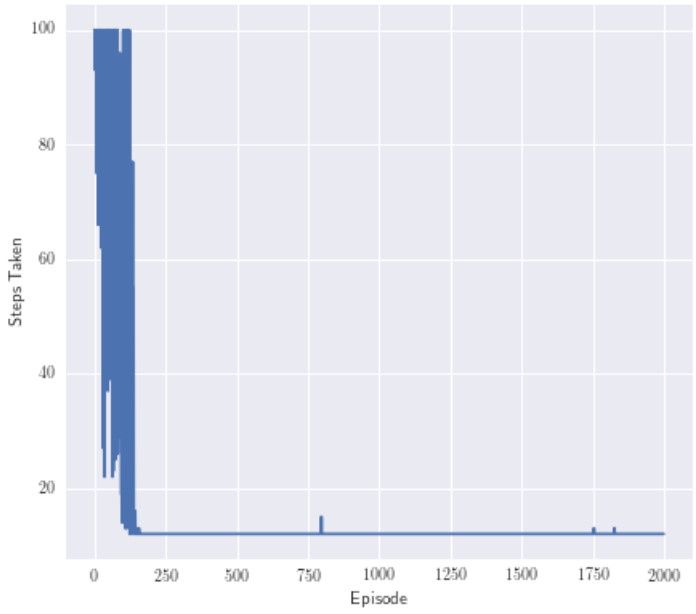


Best Plots

The plots for best set of Hyper Parameters found using Wandb analysis and also supported by variations in above section.

Reward Curve





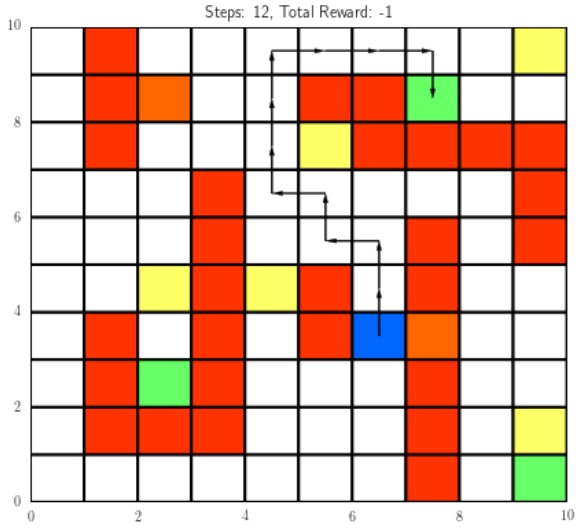
Best plots for:

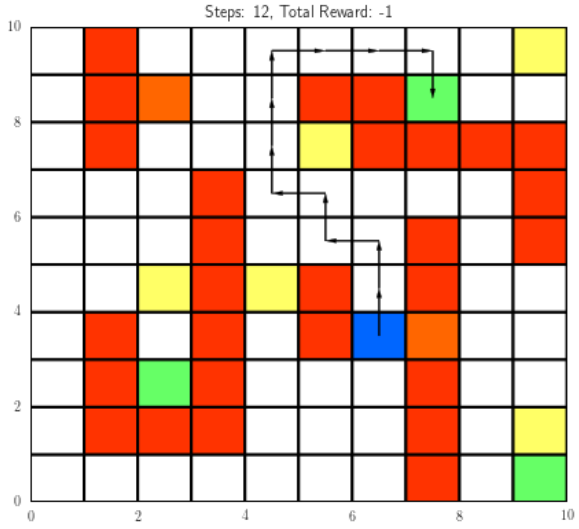
- Algorithm - Sarsa
- Policy - Softmax
- Beta - 0.6001
- Alpha - 0.1832
- Gamma - 0.9615

Final Learned Policy

We have a similar situation as in config-6 wherein there is no stochasticity involved in the transitions w.r.t wind or deviations(as p=1). The agent finally learns to follow shortest path maximising reward in 12 steps with reward=-1.
There is an important difference from config-6(since we have a different start point) that we can see in the plots for e-greedy policy. Along the optimal path while still trying to explore, the agent falls into the restart states or bad states and incurs heavy negative rewards at some points.

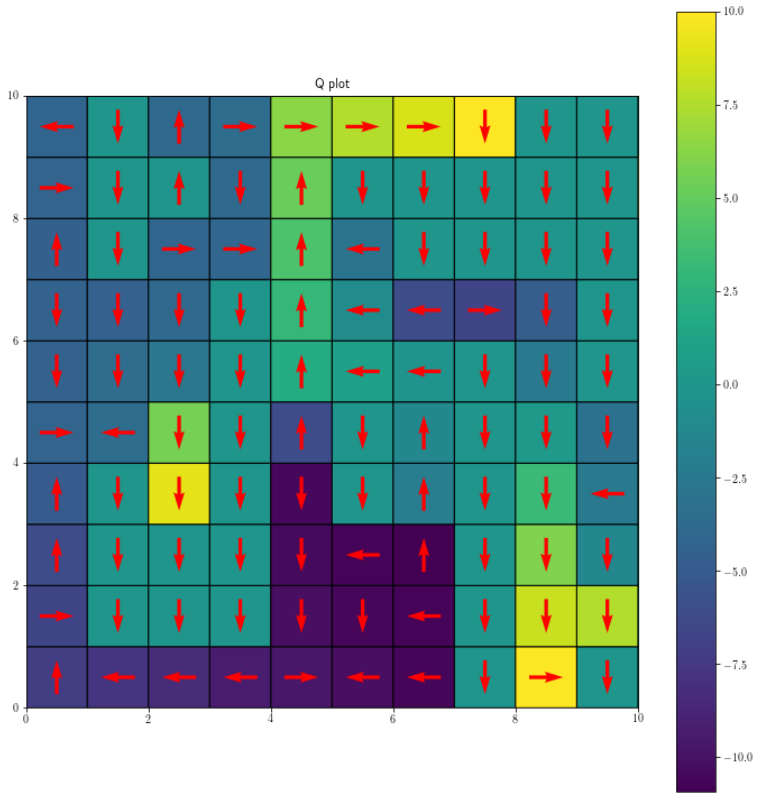
The trained agent is made to explore the environment for 2 runs. The visited states and actions are plotted





HeatMap

Heatmap of Grid World with Q values and optimal actions (of final policy learnt)

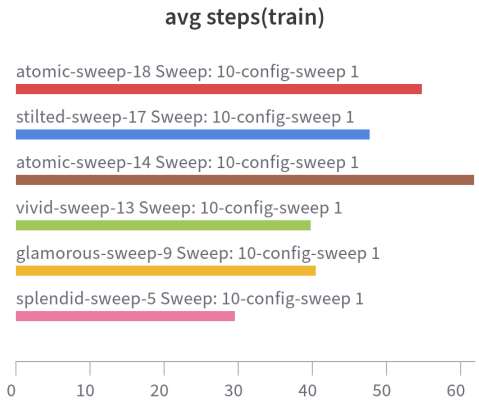

Q plot
0
2
4
6
8
10
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0

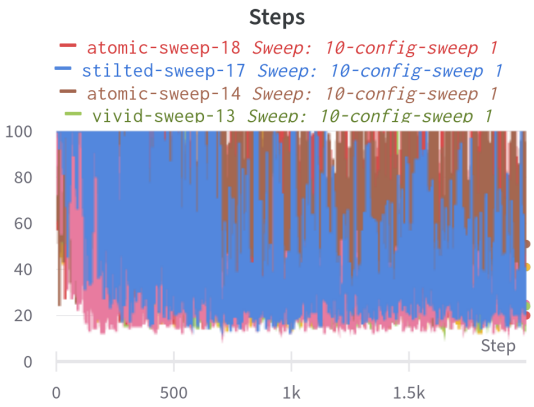
Configuration 10

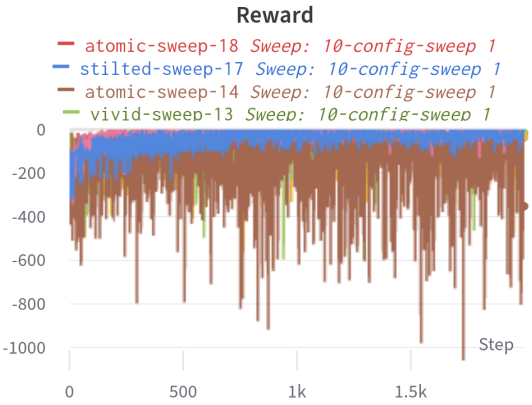
Configuration parameters

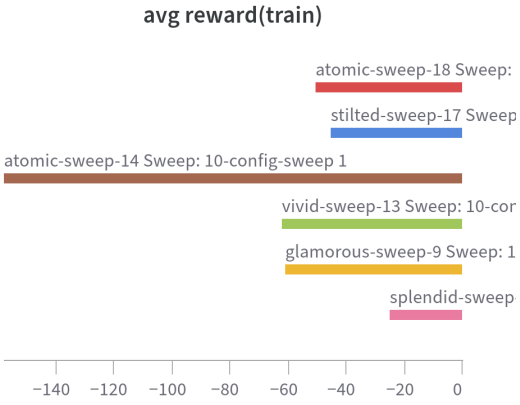
Wind = **False**, Start State = **[3,6]**, p = **0.7**

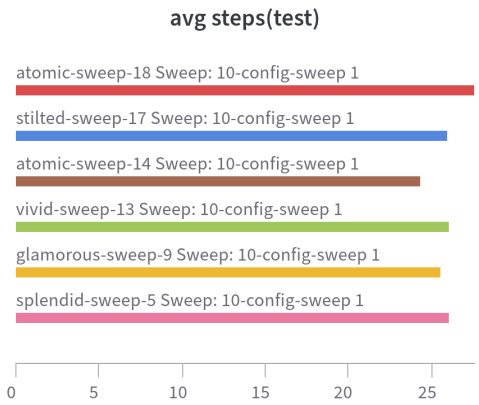
Wandb Analysis

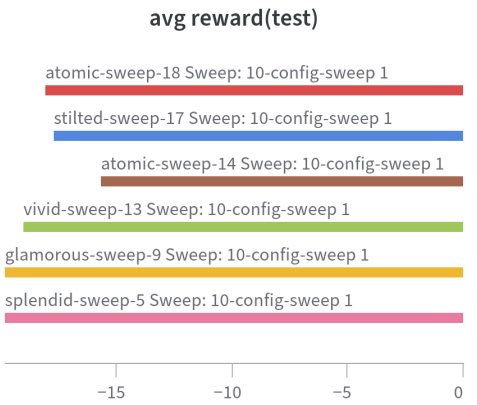


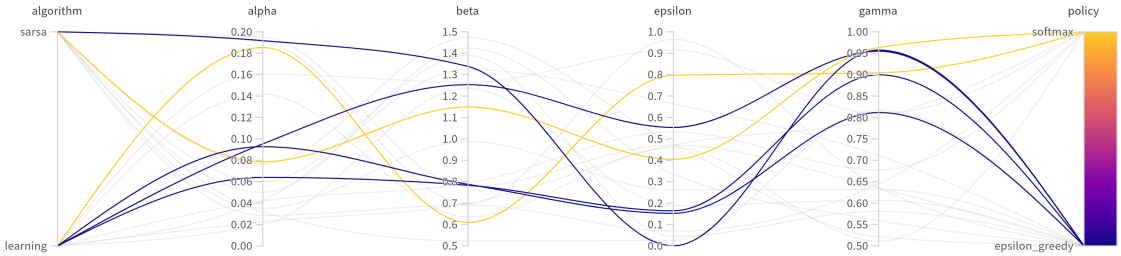










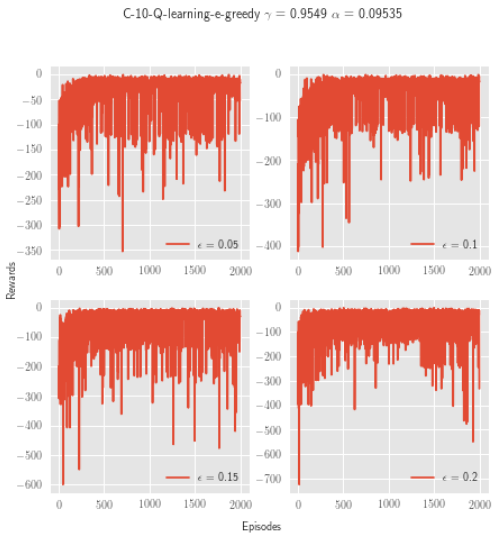


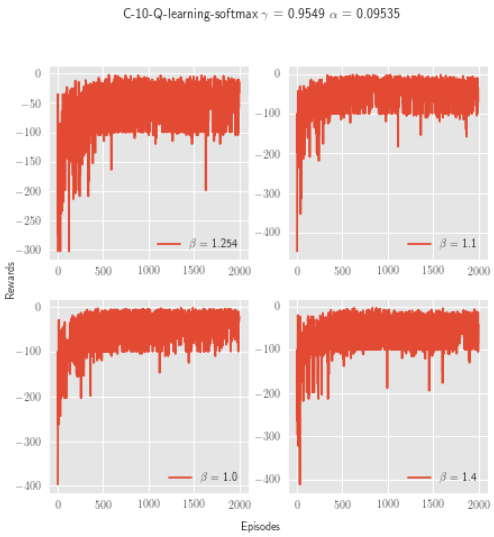
Hyper-parameter Plots

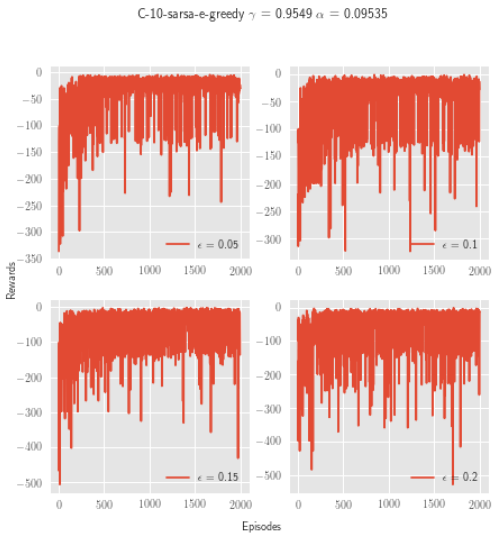
Fixing the best Hyper parameters found in the above section, other values are tried to in order to verify the observations.

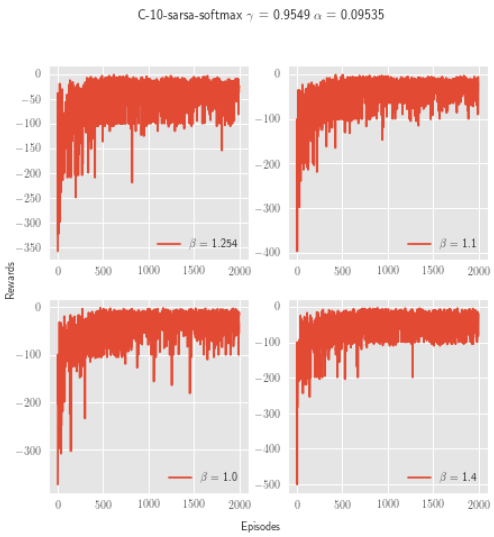
Policy Greed Variations

In this for each Algorithm and policy, α and γ are fixed and the policy greed i.e., ϵ or β , depending on the policy, is varied.





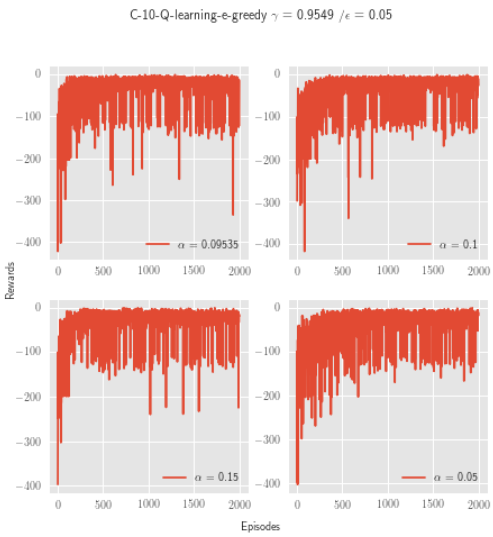


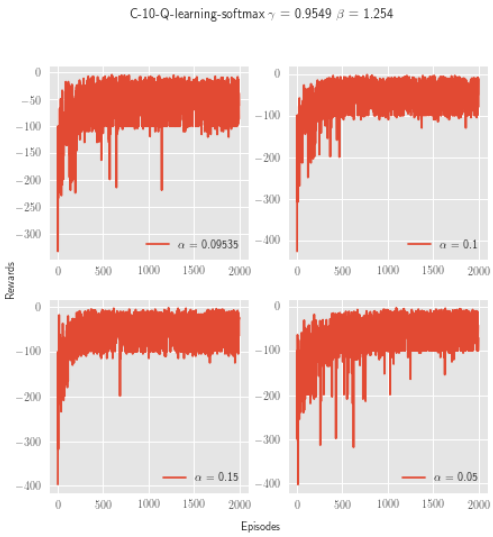


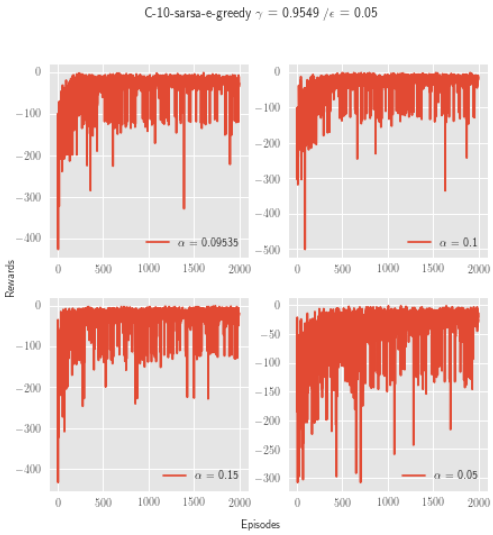
Inferences:

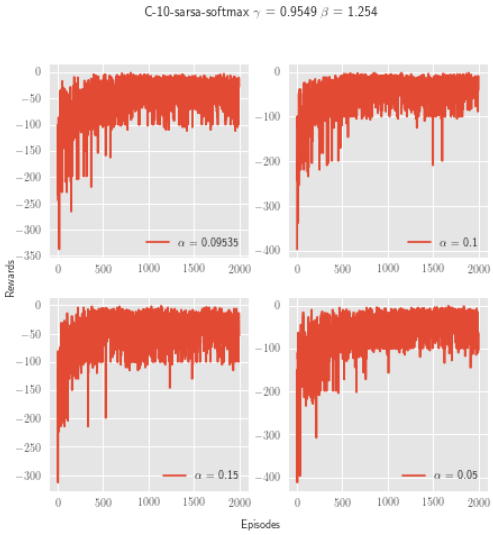
Learning Rate Variations

In this for each Algorithm and policy, policy greed and γ are fixed and Learning Rate α is varied.


C-10-Q-learning-e-greedy $\gamma = 0.9549$ $/\epsilon = 0.05$
$\alpha = 0.09535$
$\alpha = 0.1$
$\alpha = 0.15$
$\alpha = 0.05$
Rewards
Episodes

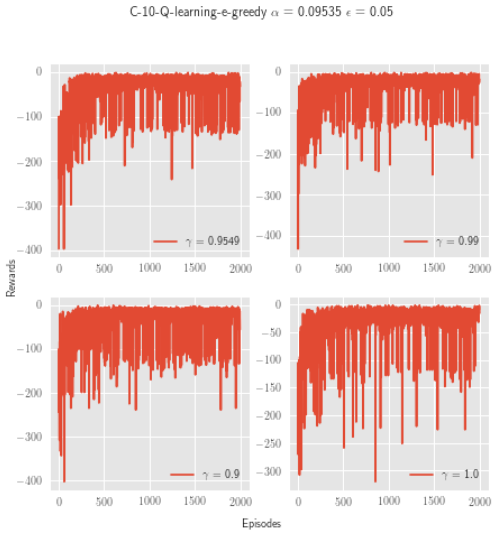

C-10-Q-learning-softmax $\gamma = 0.9549$ $\beta = 1.254$
$\alpha = 0.09535$
$\alpha = 0.1$
$\alpha = 0.15$
$\alpha = 0.05$
Rewards
Episodes

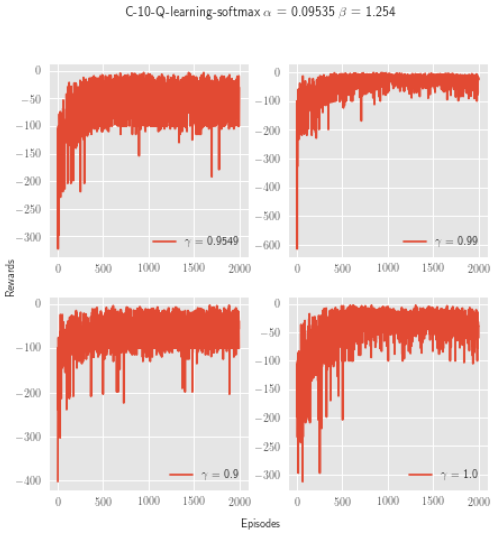

C-10-sarsa-e-greedy $\gamma = 0.9549$ $/\epsilon = 0.05$
$\alpha = 0.09535$
$\alpha = 0.1$
$\alpha = 0.15$
$\alpha = 0.05$
Rewards
Episodes

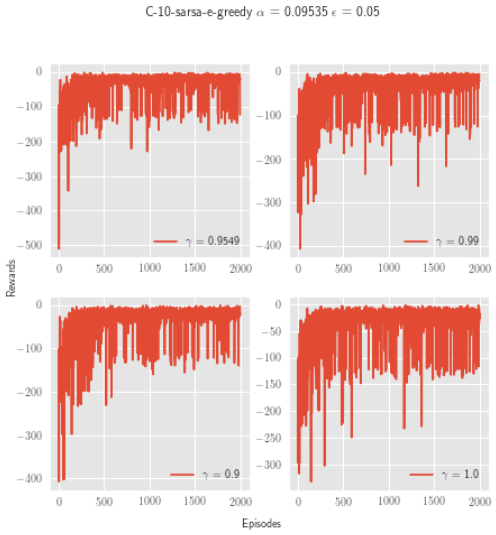

C-10-sarsa-softmax $\gamma = 0.9549$ $\beta = 1.254$
$\alpha = 0.09535$
$\alpha = 0.1$
$\alpha = 0.15$
$\alpha = 0.05$
Rewards
Episodes

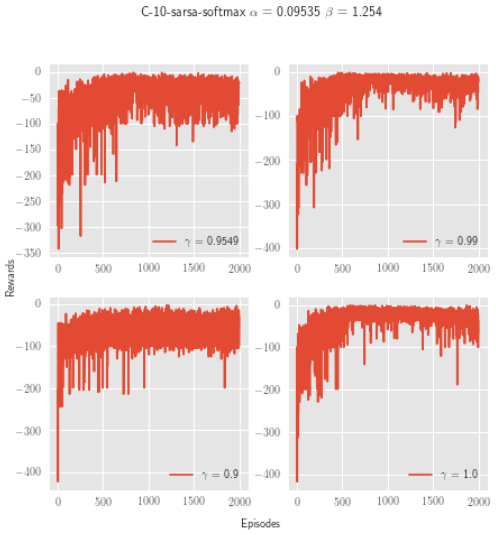
Inferences:

Discount Rate Variations







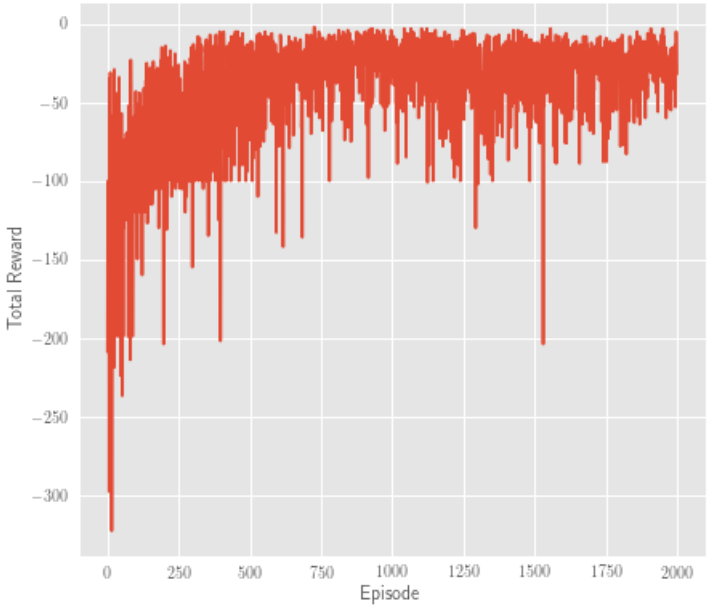


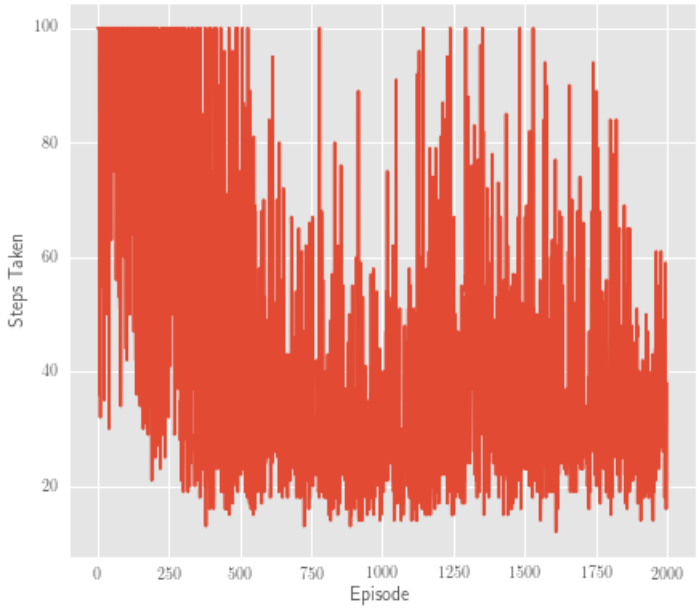
Inferences:

Best Plots

The plots for best set of Hyper Parameters found using Wandb analysis and also supported by variations in above section.

Reward Curve

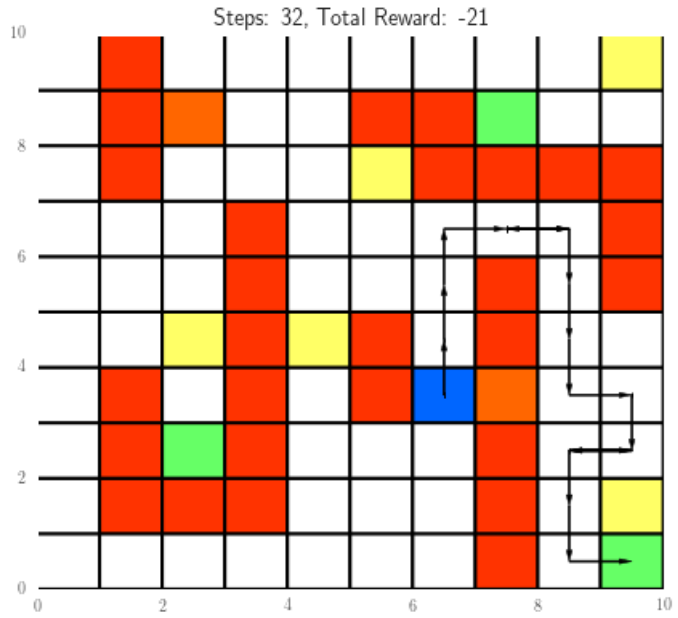


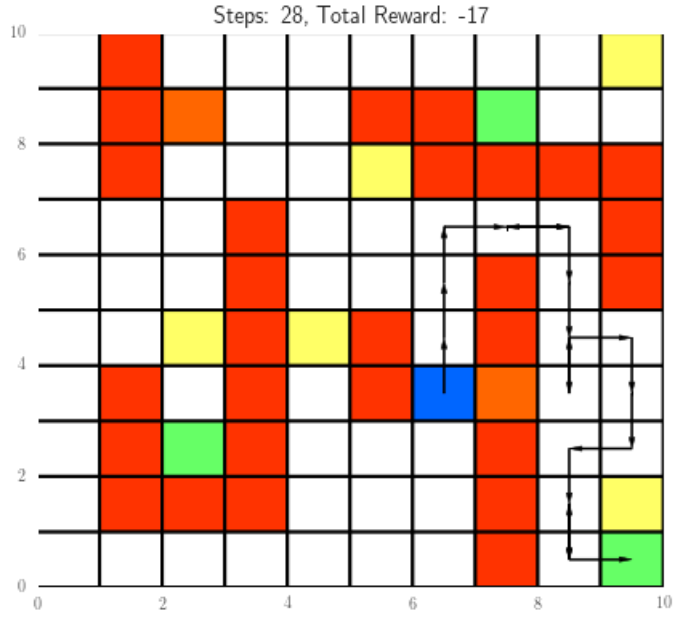


- Algorithm - q-learning
- Policy - softmax
- Beta - 1.254
- Alpha - 0.09535
- Gamma - 0.99

Final Learned Policy

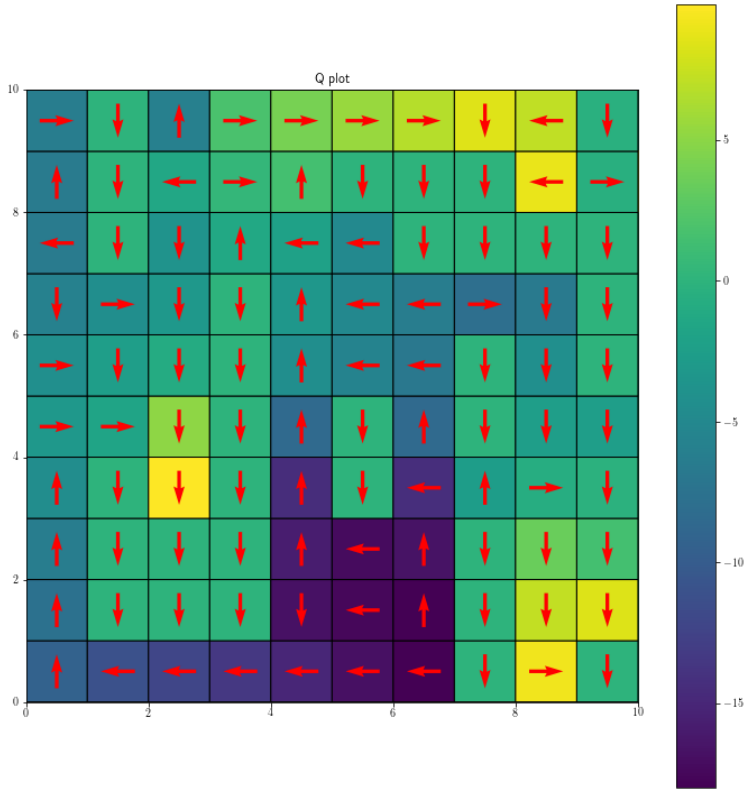
Though wind is not present, we do have stochasticity for p=0.7, leading to the same inferences as in config-4.
The trained agent is made to explore the environment for 2 runs. The visited states and actions are plotted


Steps: 32, Total Reward: -21
10
8
6
4
2
0
0
2
4
6
8
10


Steps: 28, Total Reward: -17
10
8
6
4
2
0
0
2
4
6
8
10

HeatMap

Heatmap of Grid World with Q values and optimal actions (of final policy learnt)

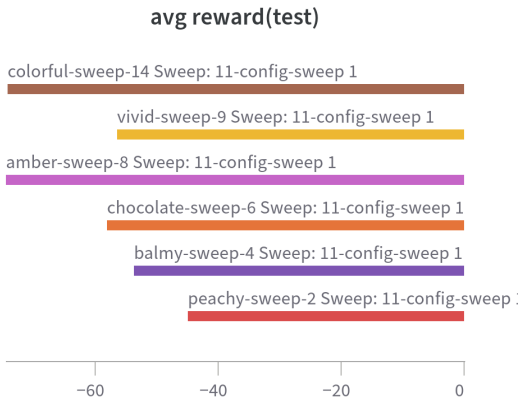


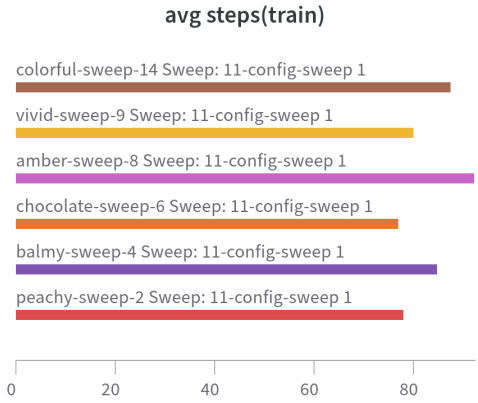
Configuration 11

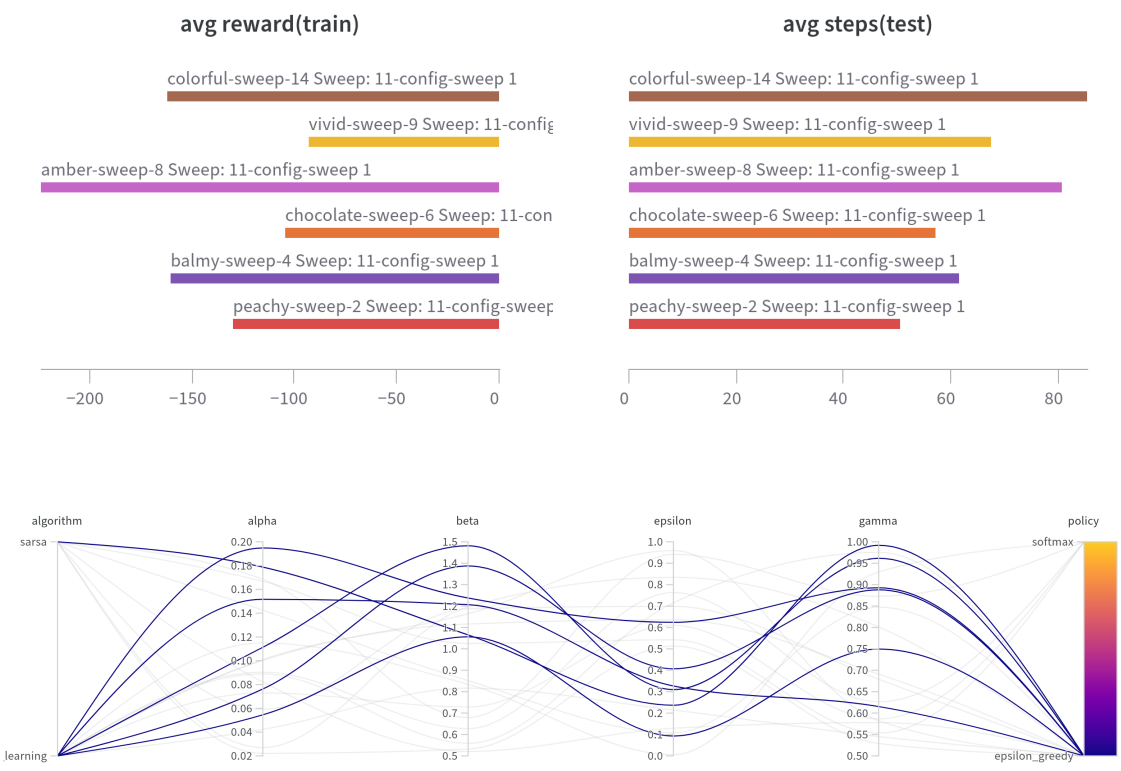
Configuration parameters

Wind = **False**, Start State = **[3,6]**, p = **0.35**

Wandb Analysis





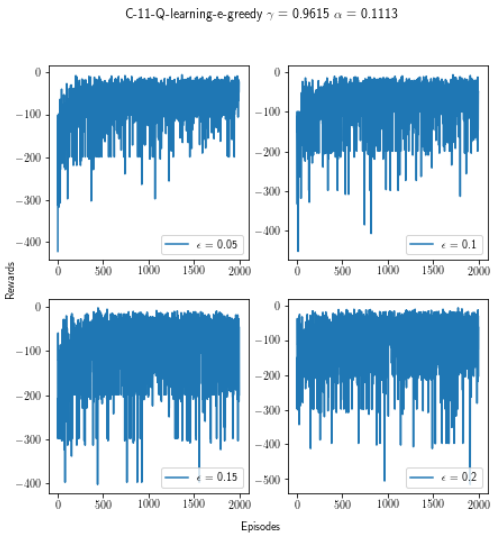


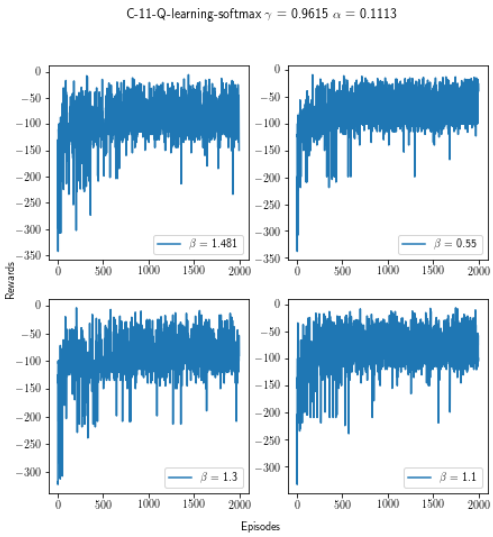
Hyper-parameter Plots

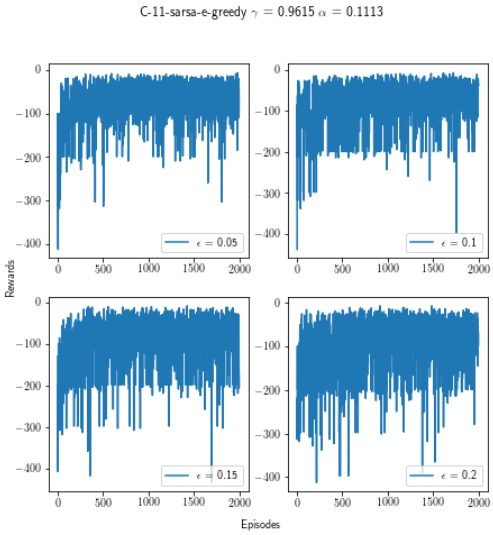
Fixing the best Hyper parameters found in the above section, other values are tried to in order to verify the observations.

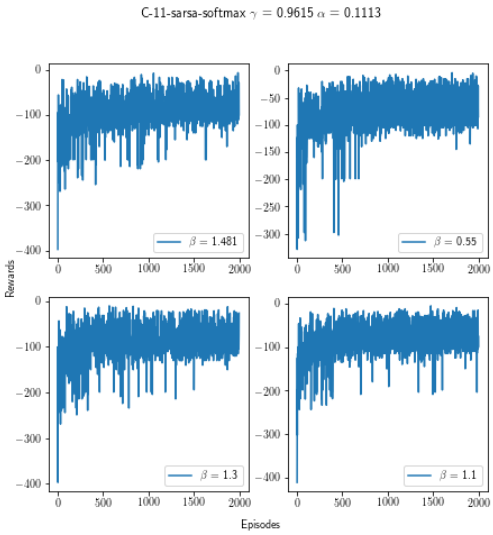
Policy Greed Variations

In this for each Algorithm and policy, α and γ are fixed and the policy greed i.e., ϵ or β , depending on the policy, is varied.





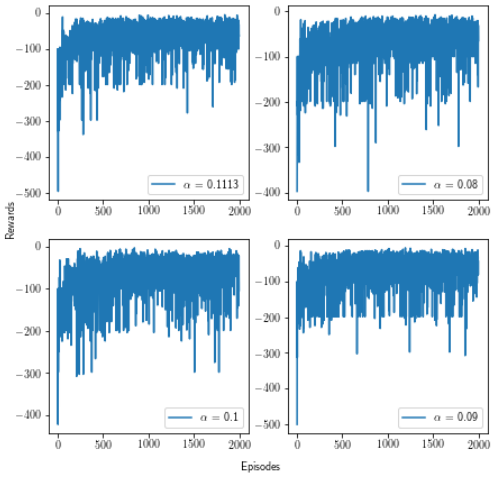




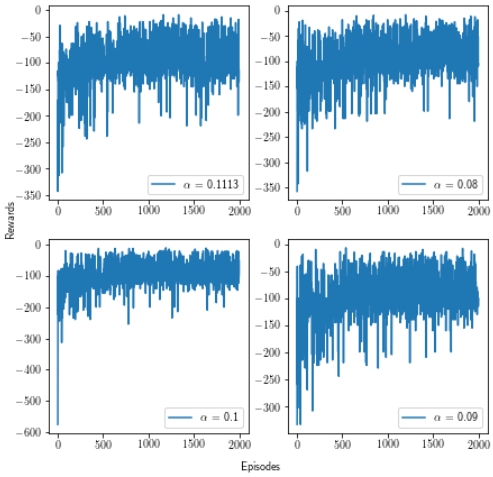
Learning Rate Variations

In this for each Algorithm and policy, policy greed and γ are fixed and Learning Rate α is varied.

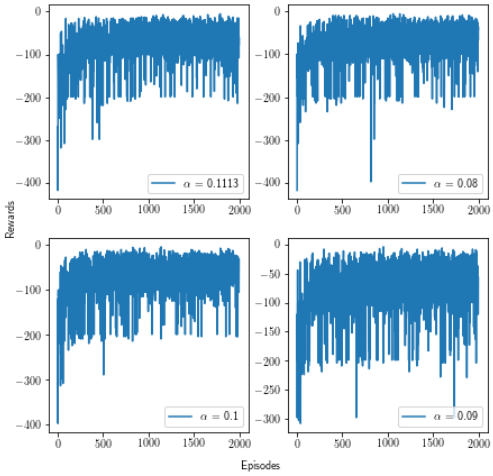
C-11-Q-learning-e-greedy $\gamma = 0.9615$ $/\epsilon = 0.05$



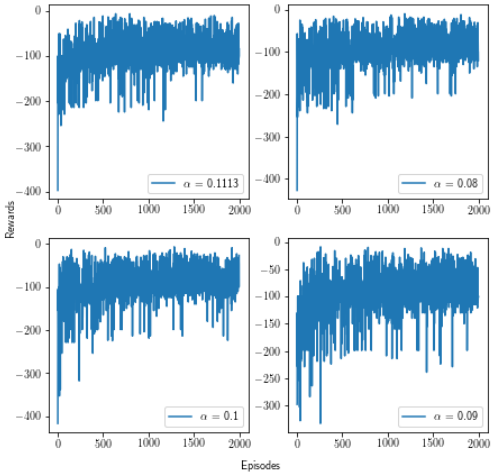
C-11-Q-learning-softmax $\gamma = 0.9615$ $\beta = 1.481$



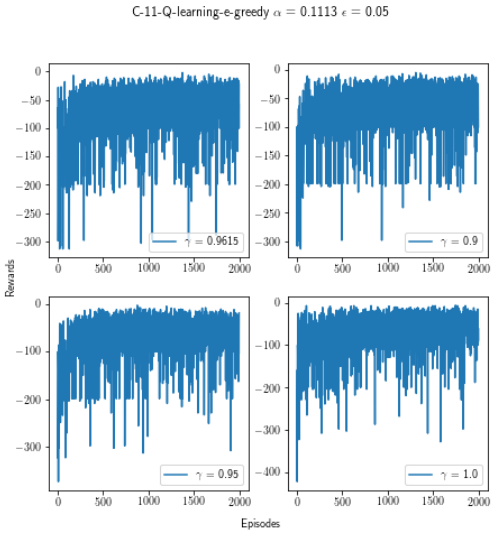
C-11-sarsa-e-greedy $\gamma = 0.9615$ $/\epsilon = 0.05$

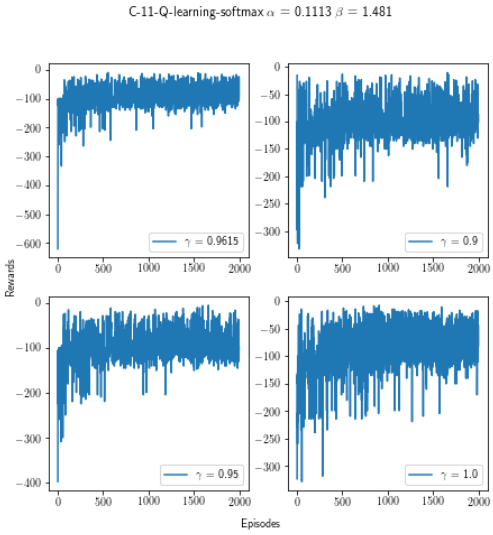


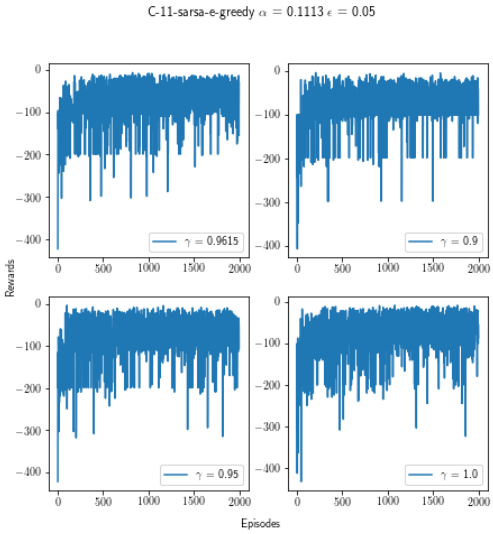
C-11-sarsa-softmax $\gamma = 0.9615$ $\beta = 1.481$

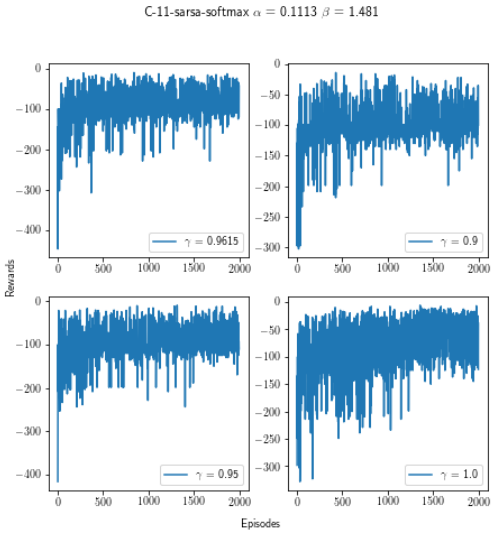


Discount Rate Variations





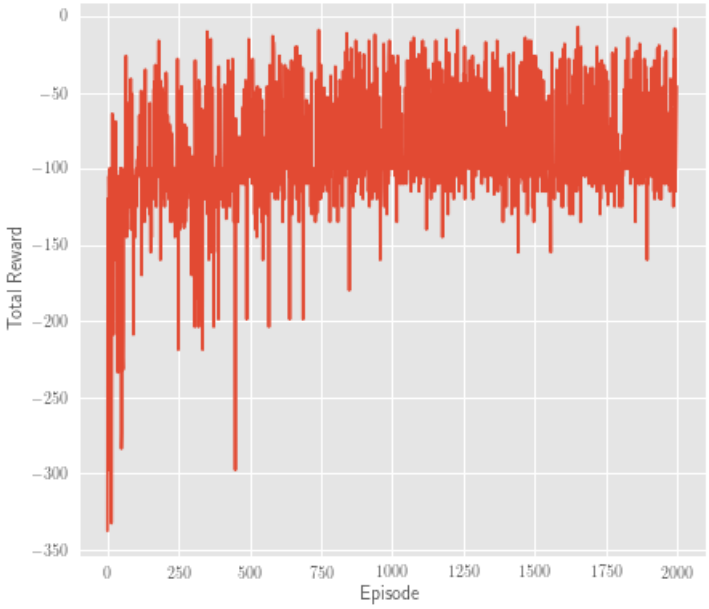


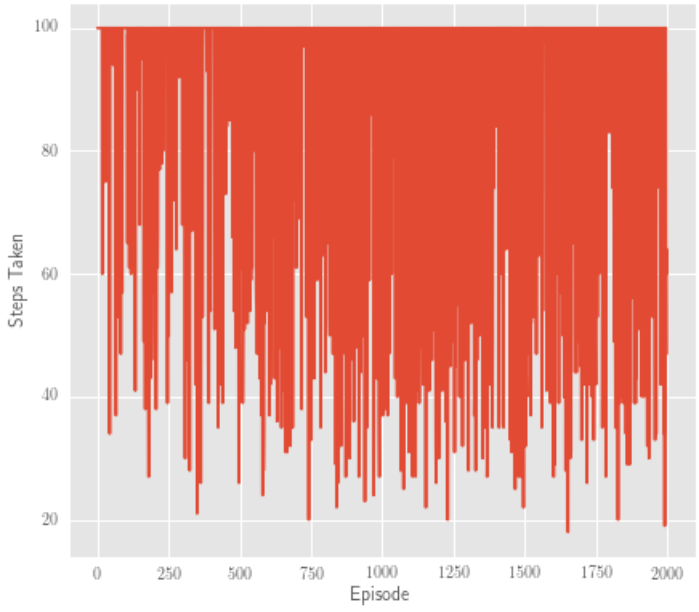


Best Plots

The plots for best set of Hyper Parameters found using Wandb analysis and also supported by variations in above section.

Reward Curve



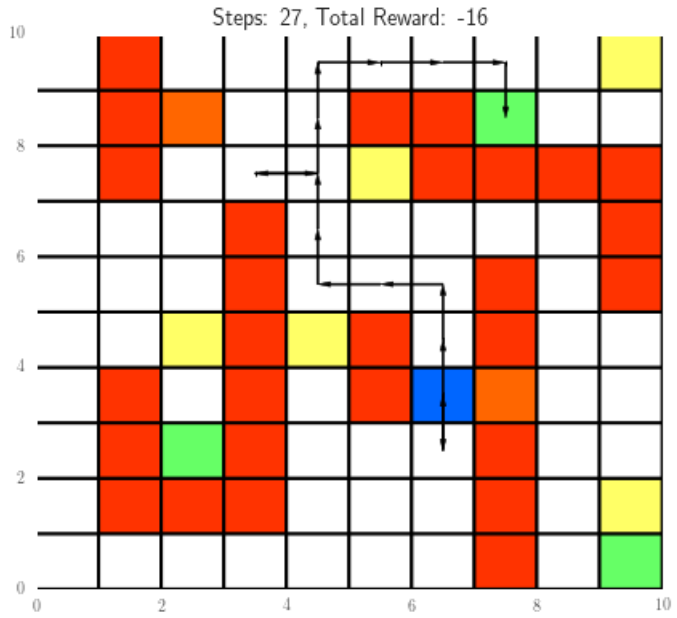


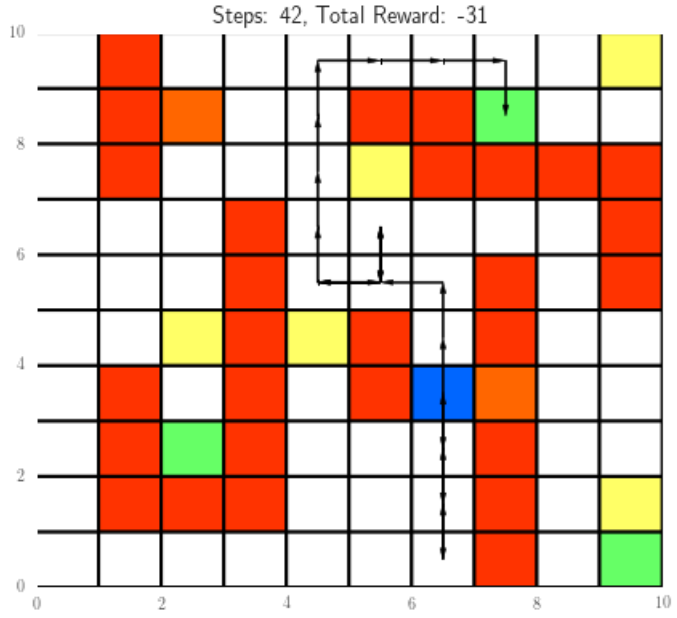
Best plots for:

- Algorithm - q-learning
- Policy - softmax
- Beta - 1.481
- Alpha - 0.09
- Gamma - 0.9615

Final Learned Policy

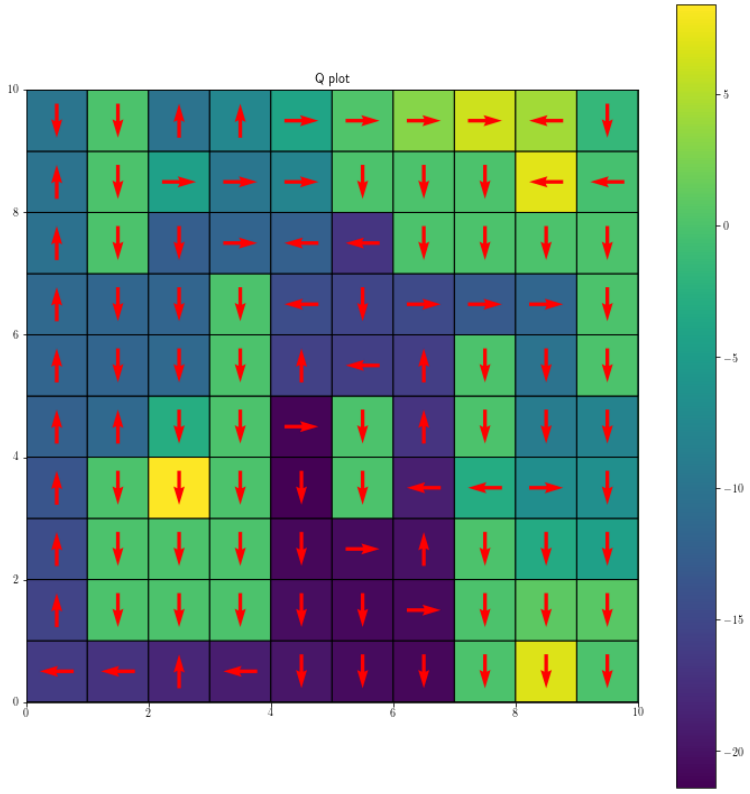
The trained agent is made to explore the environment for 2 runs. The visited states and actions are plotted


Steps: 27, Total Reward: -16
10
8
6
4
2
0
0
2
4
6
8
10


Steps: 42, Total Reward: -31
10
8
6
4
2
0
0
2
4
6
8
10

HeatMap

Heatmap of Grid World with Q values and optimal actions (of final policy learnt)



Conclusion: Throughout the twelve configurations we've tried to train the agent through different environments. The start states significantly affects the way the agent learns, its rewards, the number of steps to the goal. We've seen that with no stochasticity the agent follows a fixed path, while by adding wind and decreasing p we see more off-beat paths to the goal.We've also seen how the algorithm/ stochasticity in the environment nudges the agent to explore new paths to different goal-states(as in config-4,5).