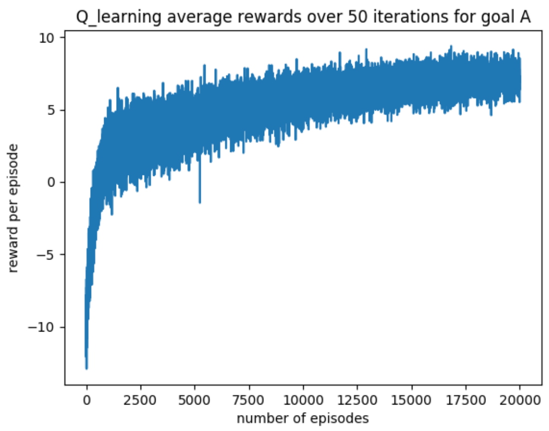
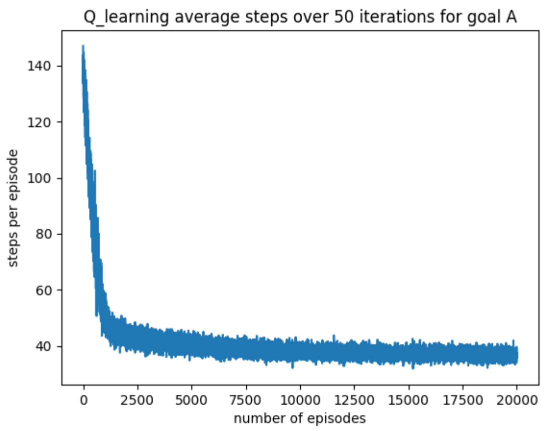
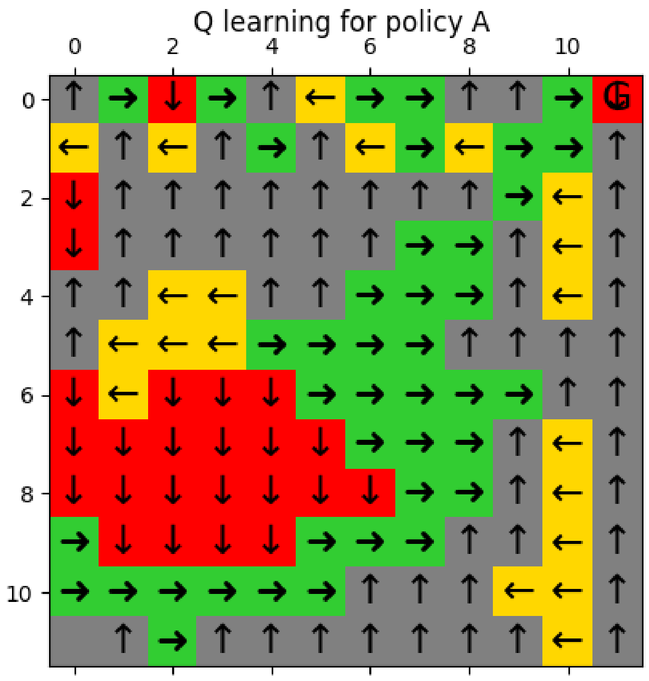
CS6700 : Reinforcement Learning
Programming Assignment #2

Name: Pranav Aurangabadkar **Roll number:** ED18S007

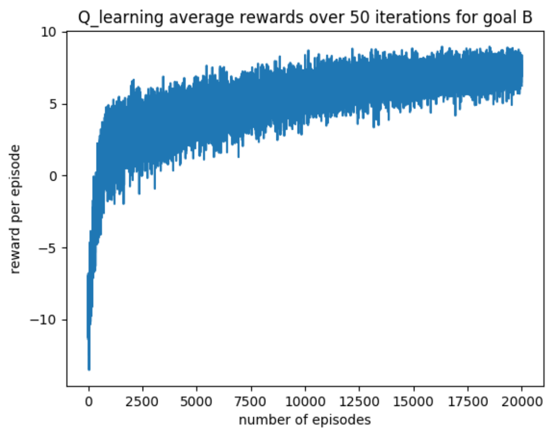
1. Implement Q-learning to solve each of the three variants of the problem. For each variant run experiments with a gamma value of 0.9. Compute the averages over 50 independent runs.

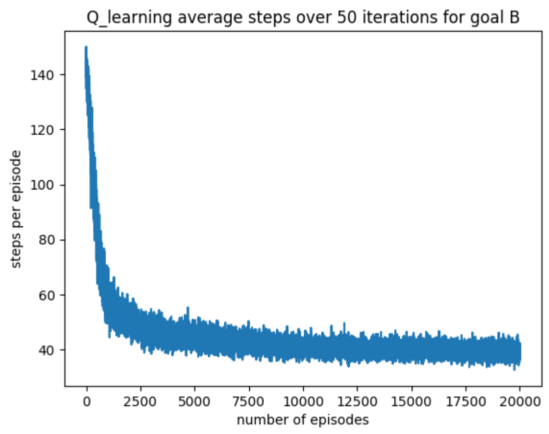


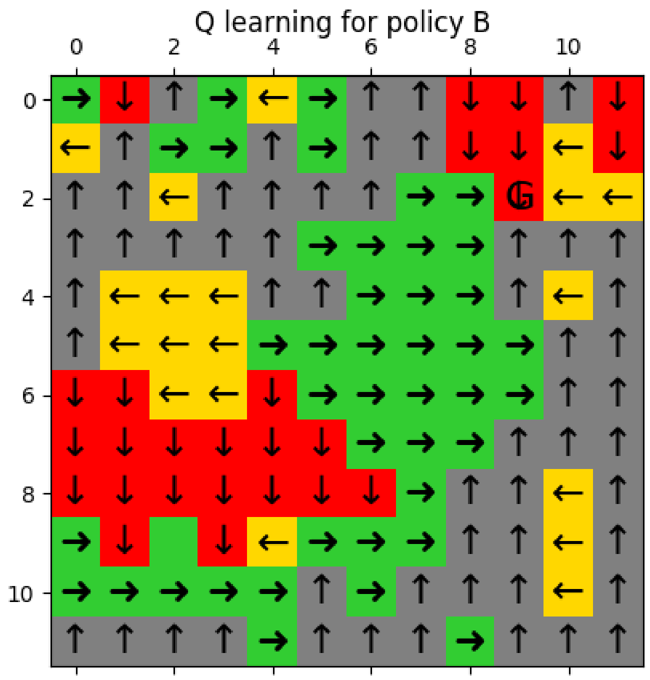




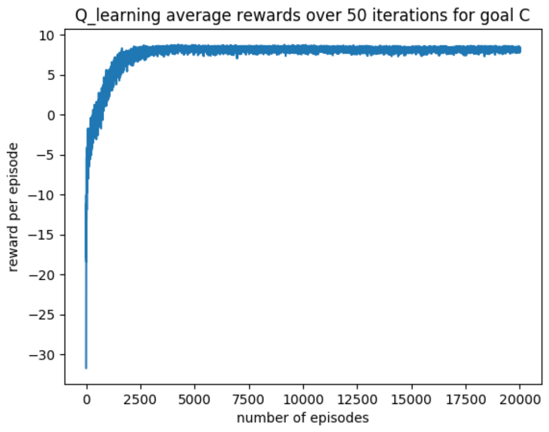
From above plots it can be seen that the average rewards is positive and continuously increases from around 2500 episodes where at this point the number of steps per episode is reduced to 50 steps. The policy plot can be traced and goal is reached within approximately 40 steps from anywhere in the puddleworld.

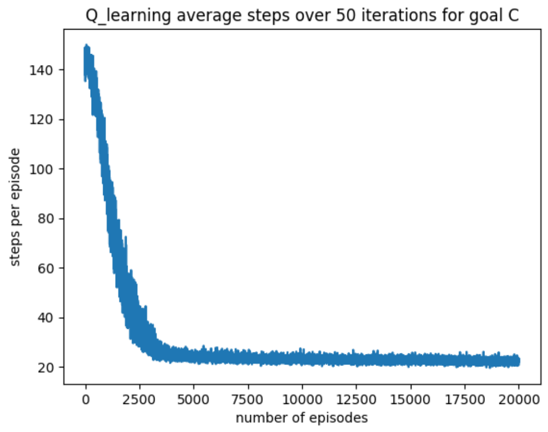


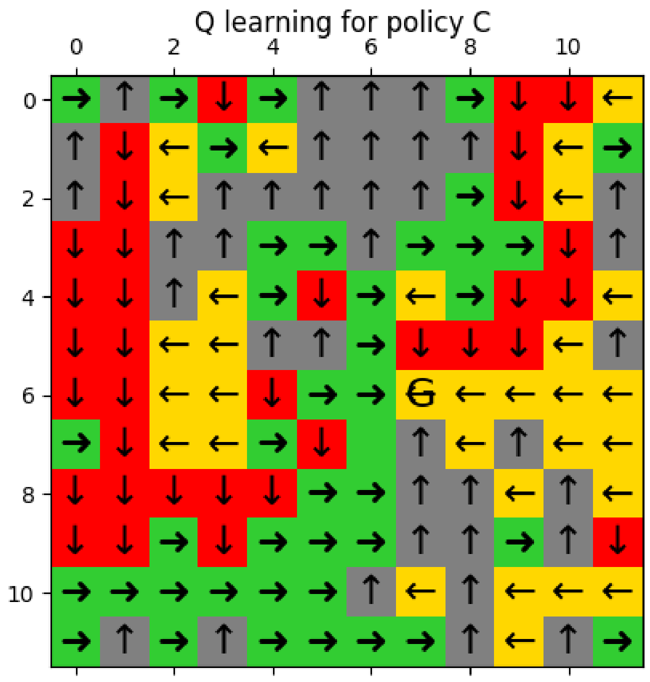




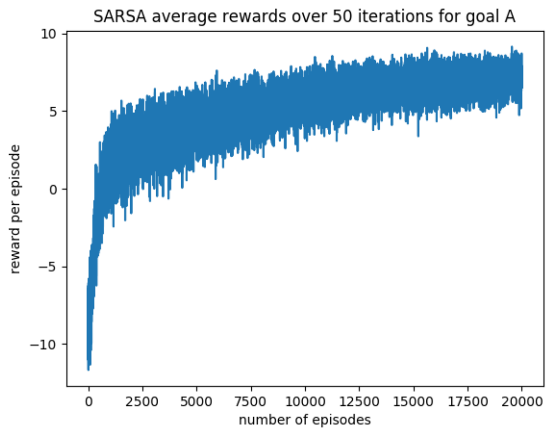
The plots for Goal B is similar to those as above for Goal A but the average rewards increases slightly above that of A the reason is B is near to that of A from each starting point similarly average steps to reach the goal is in the range 30 to 40.

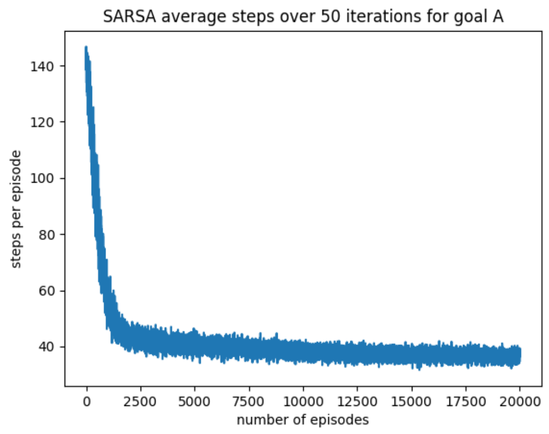


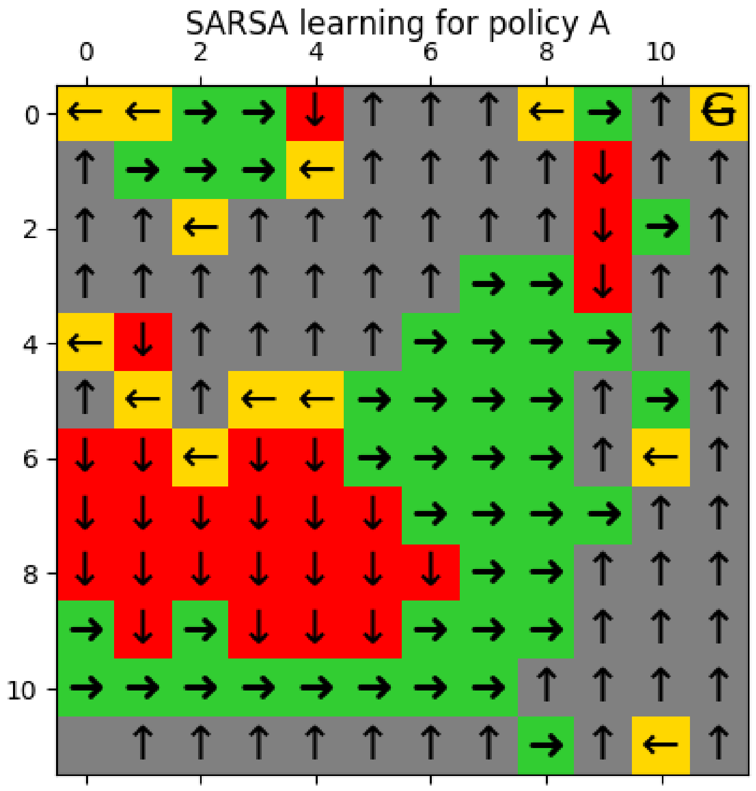




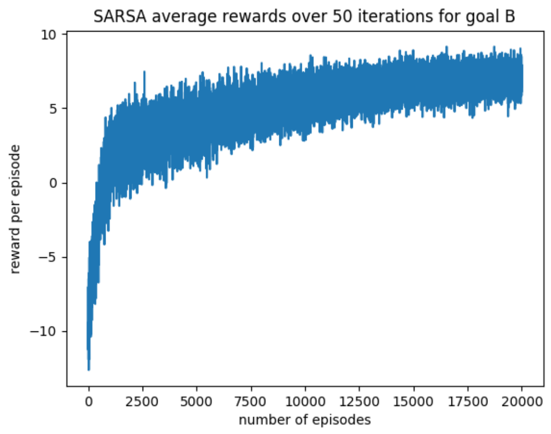
Here for goal C the average rewards increases to approximately 9 and steps to reach goal decreases to 20 over 20000 episodes.

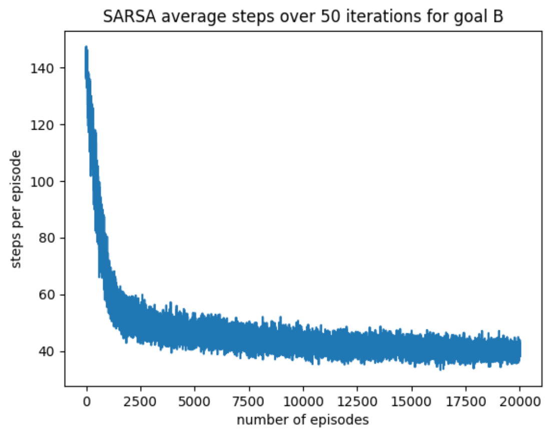


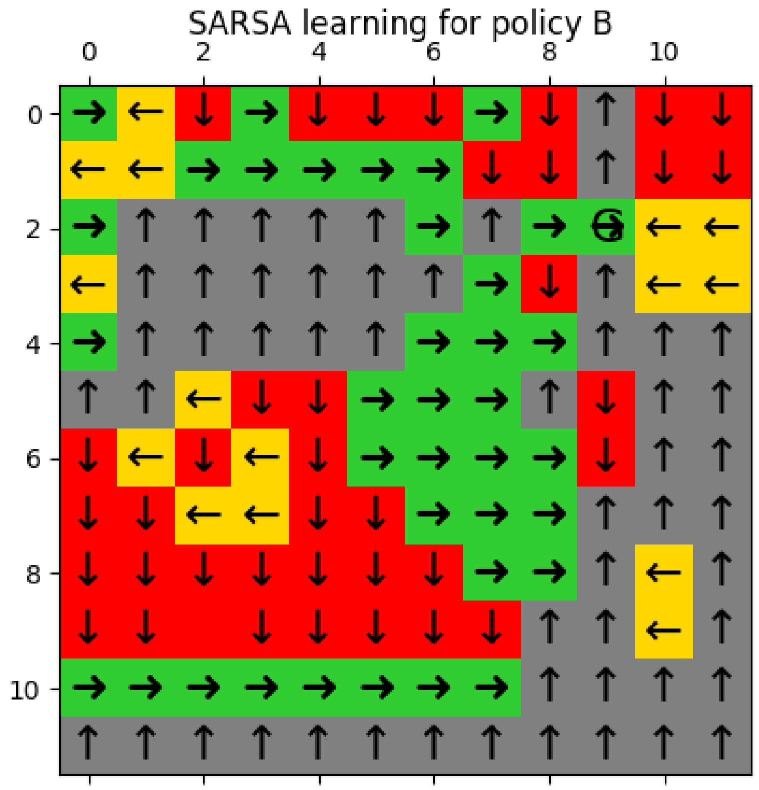




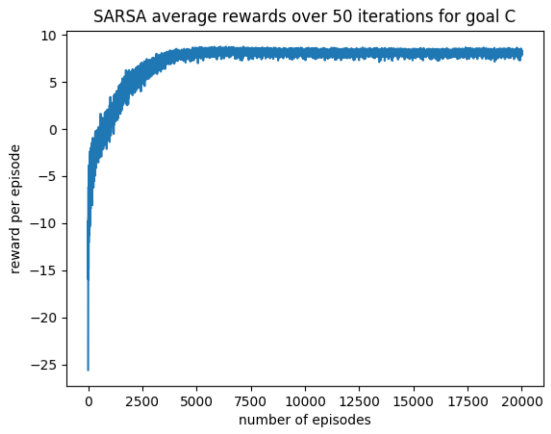
As compared to Q learning SARSA is very much similar for goal A and B. We can observe this by looking at the policy plots. For most of the states the policy remains same.

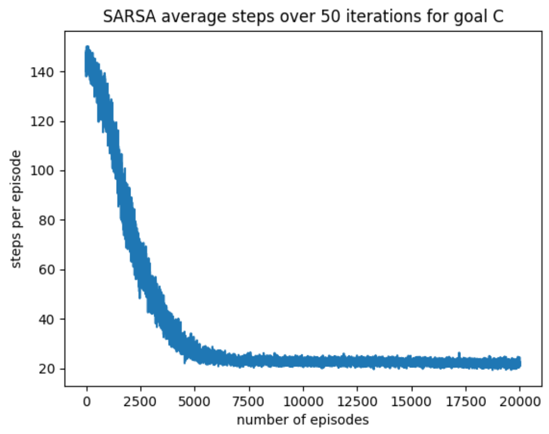


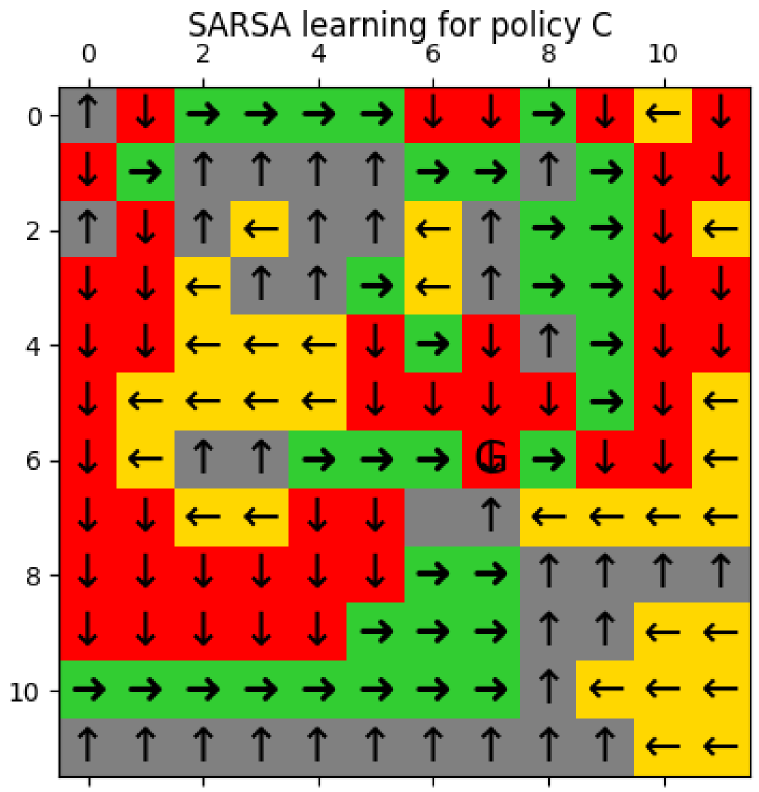




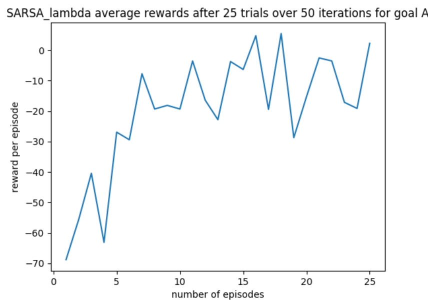
The average rewards and number of steps are also nearly same for goal A and B.

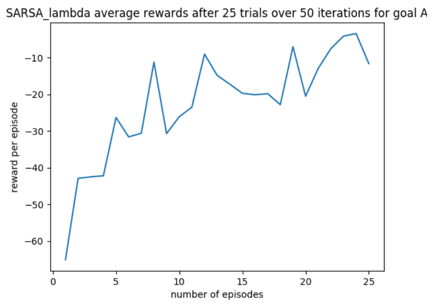


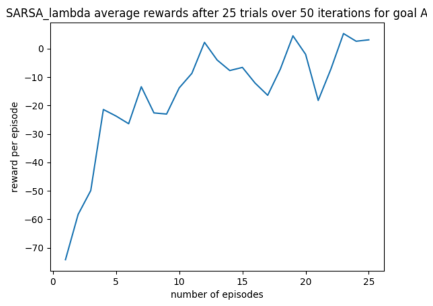


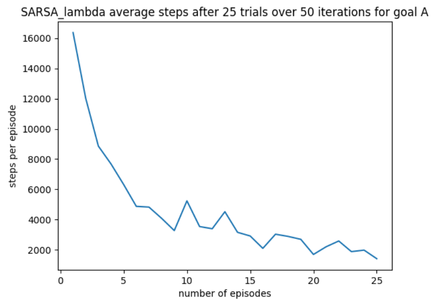


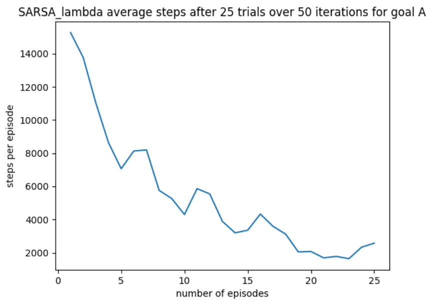
As compared to Q learning SARSA policy is largely different but both take nearly 20 steps to reach the goal with approximately reward of 9 per episode.

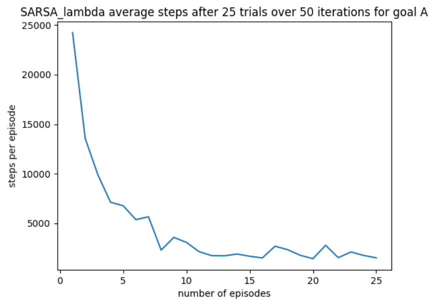

SARSA_lambda average rewards after 25 trials over 50 iterations for goal A
reward per episode
number of episodes

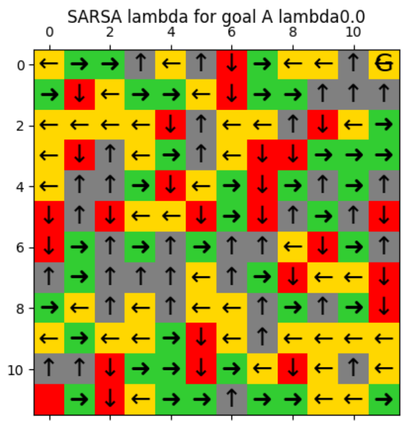

SARSA_lambda average rewards after 25 trials over 50 iterations for goal A
reward per episode
number of episodes

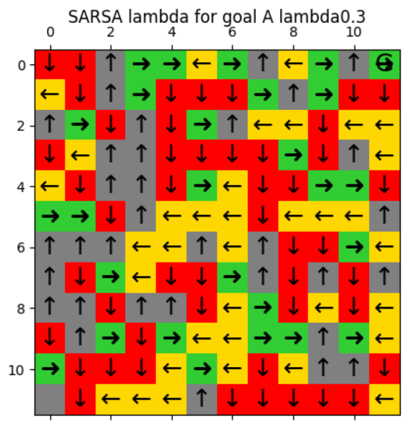

SARSA_lambda average rewards after 25 trials over 50 iterations for goal A
reward per episode
number of episodes

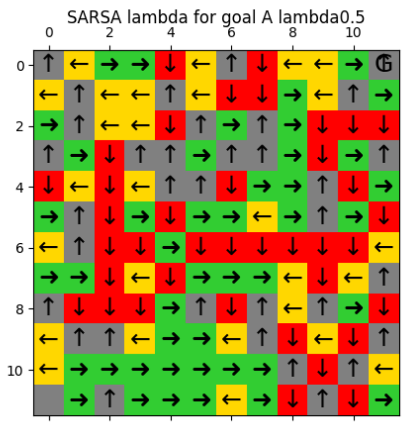

SARSA_lambda average steps after 25 trials over 50 iterations for goal A
steps per episode
number of episodes

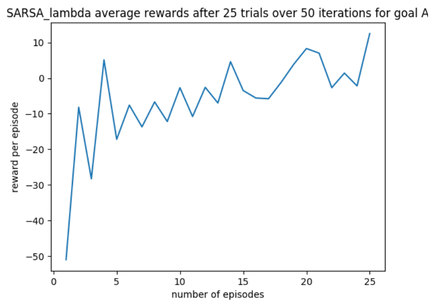

SARSA_lambda average steps after 25 trials over 50 iterations for goal A
steps per episode
number of episodes

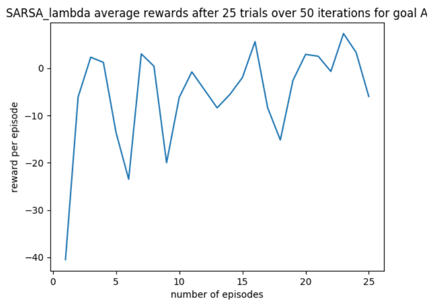

SARSA_lambda average steps after 25 trials over 50 iterations for goal A
steps per episode
number of episodes

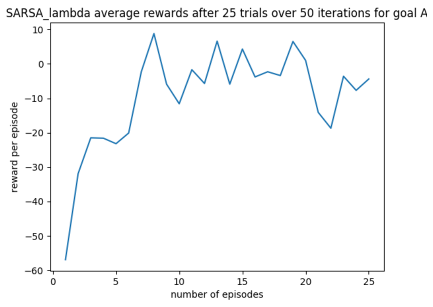

SARSA lambda for goal A lambda0.0
G

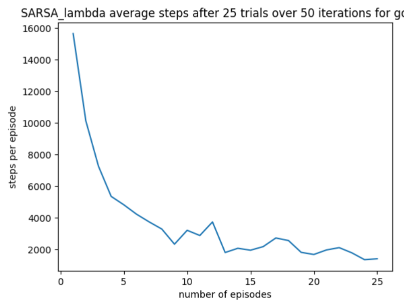

SARSA lambda for goal A lambda0.3
G

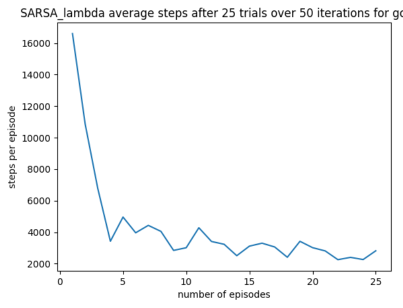

SARSA lambda for goal A lambda0.5
G

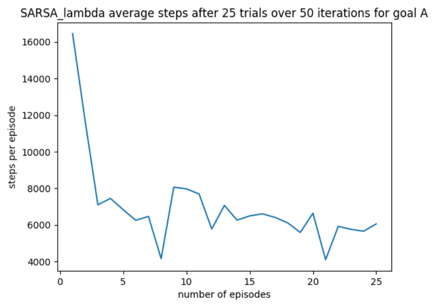

SARSA_lambda average rewards after 25 trials over 50 iterations for goal A
reward per episode
number of episodes

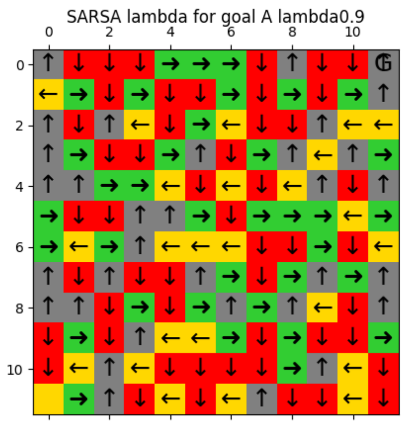

SARSA_lambda average rewards after 25 trials over 50 iterations for goal A
reward per episode
number of episodes

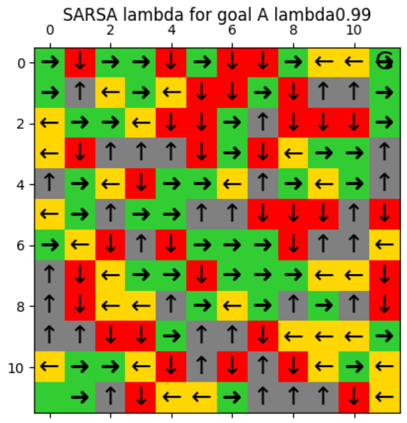

SARSA_lambda average rewards after 25 trials over 50 iterations for goal A
reward per episode
number of episodes

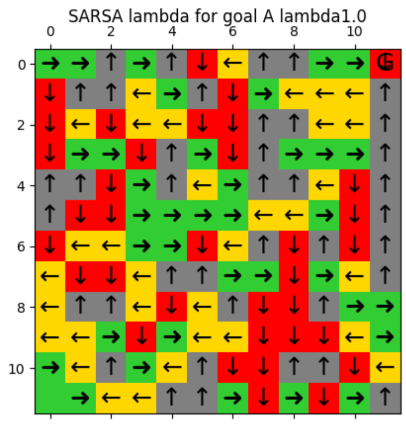




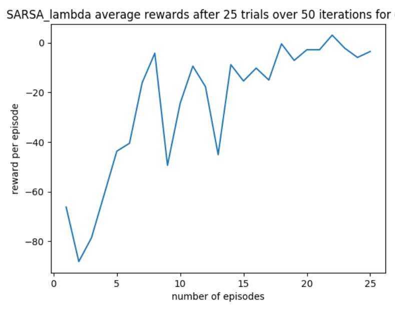


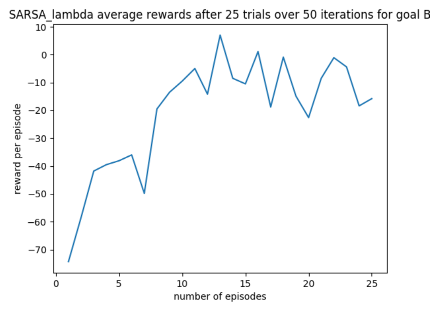


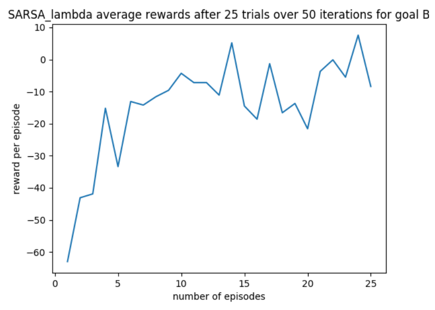


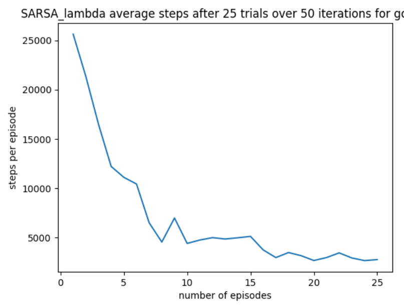


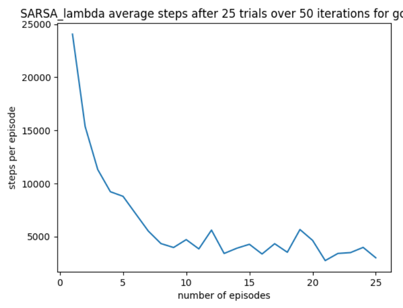
For goal A as lambda value increases from 0.0 to 1.0 the average reward over 25 trials over 50 runs increases initially it is negative then for lambda greater than 0.5 it increases. However the number of steps is observed to be lowest for lambda 0.99 value. Also the policies are different for each value of lambda.

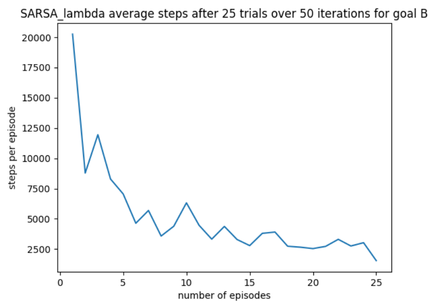


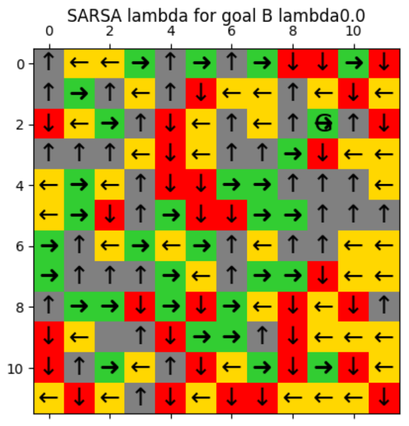


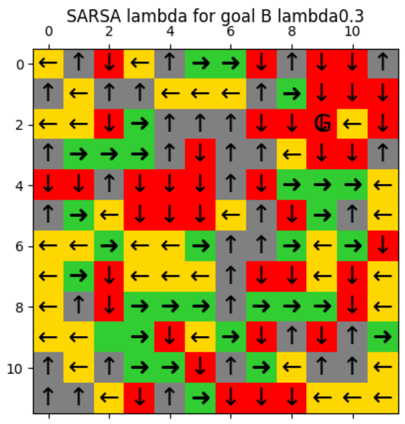


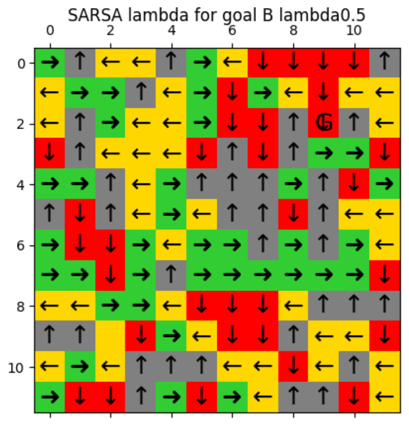


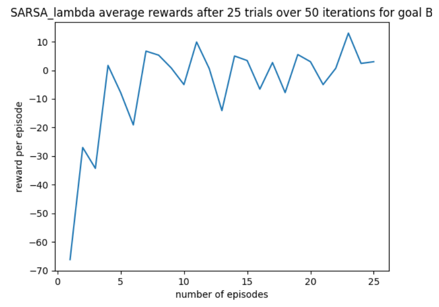


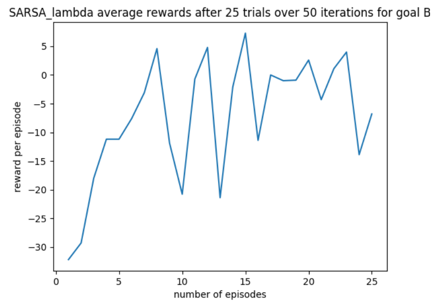


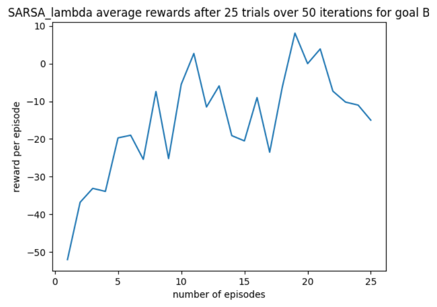


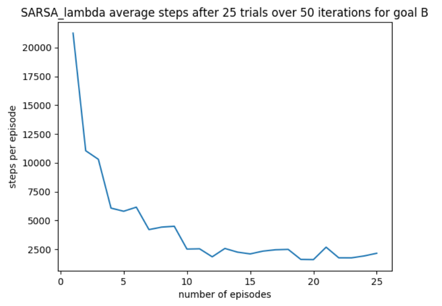


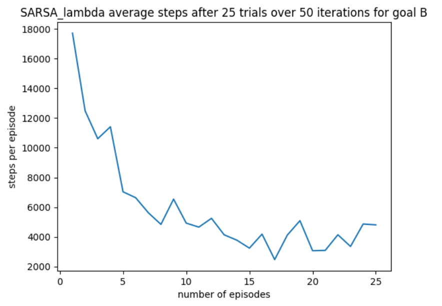


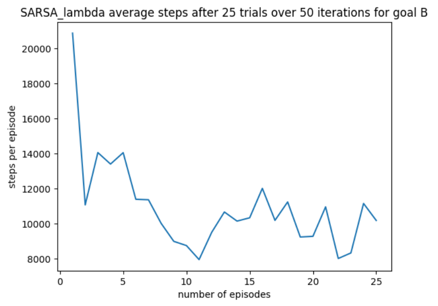


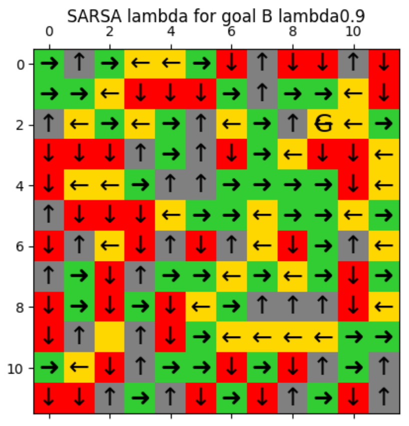


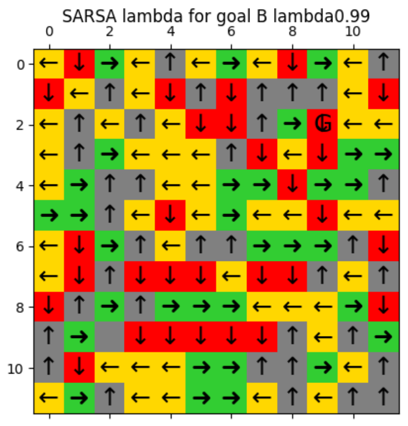


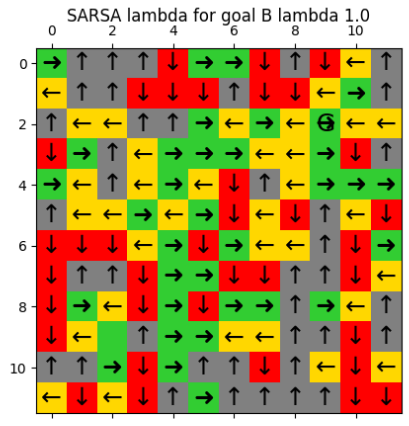




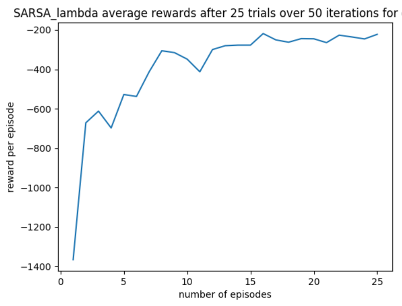
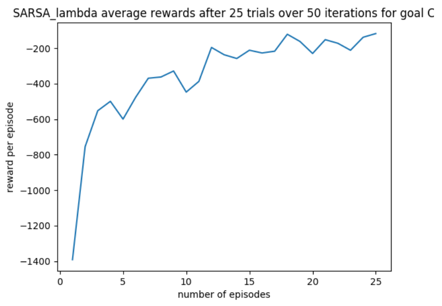
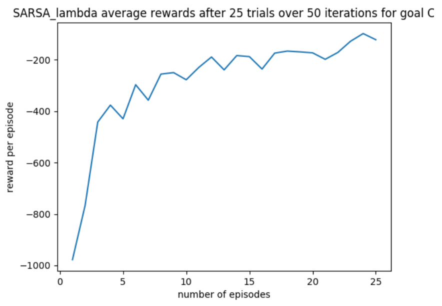
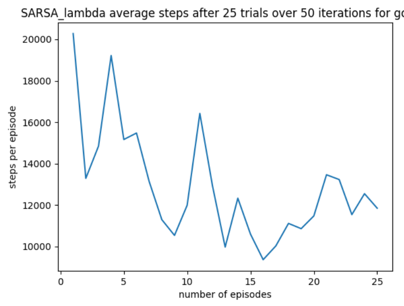
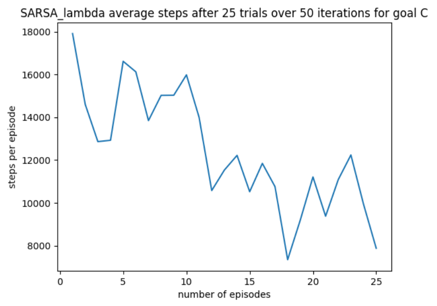
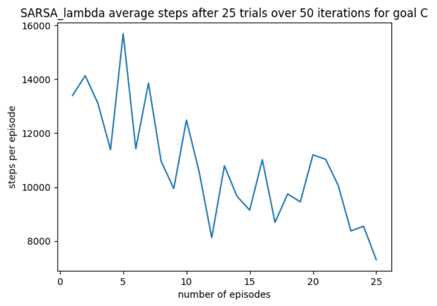
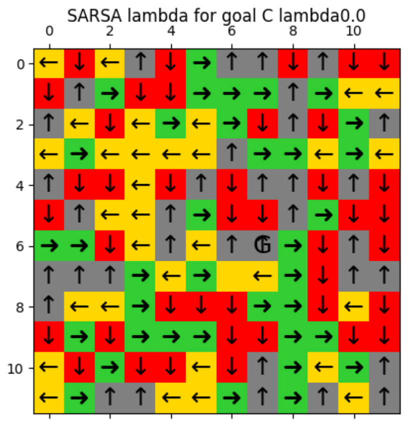
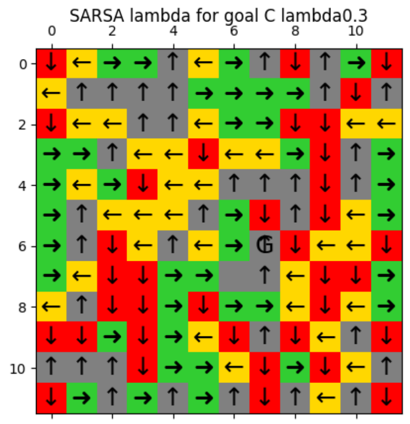
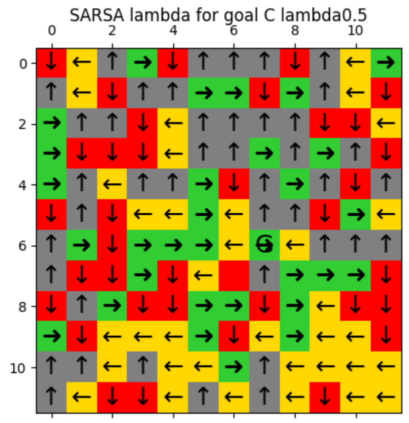
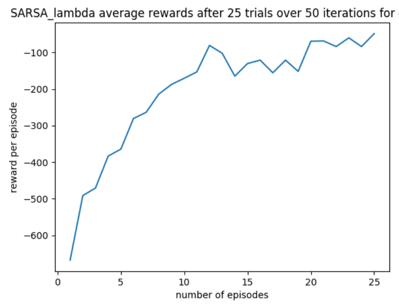
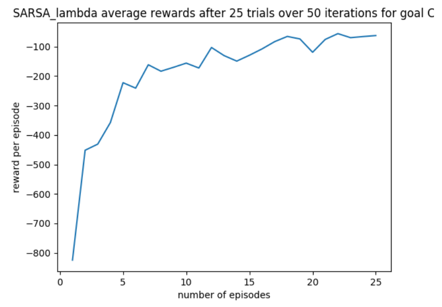
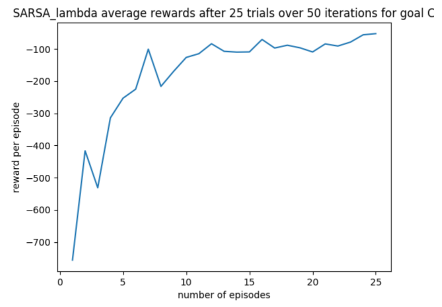


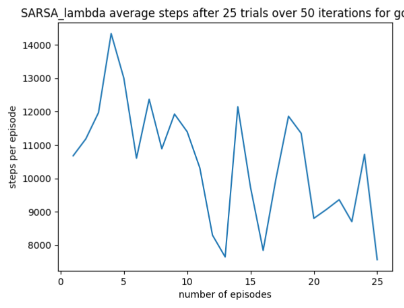


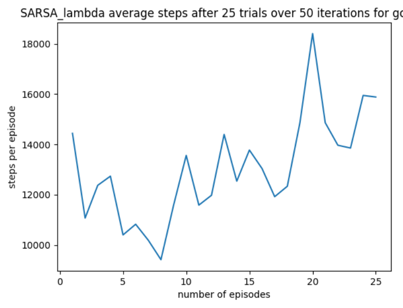


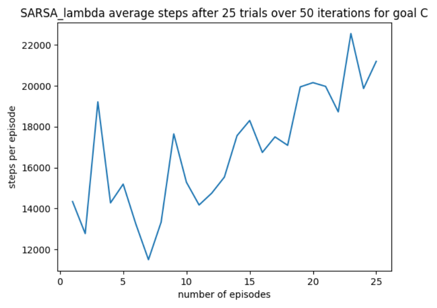


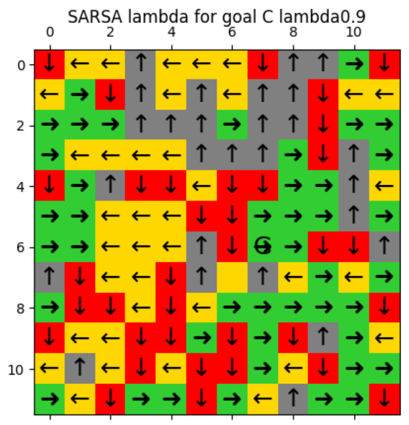
Similar to above case the reward increases over 25 trial as lambda increases but best time step is for 0.99 value and policies are also very different.

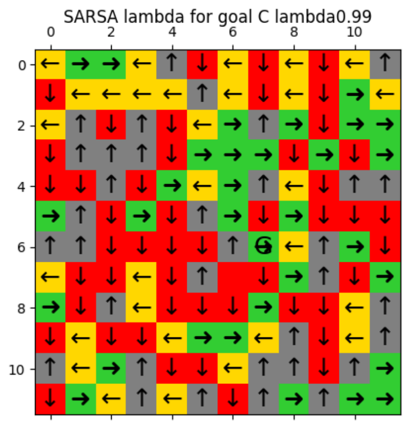

SARSA_lambda average rewards after 25 trials over 50 iterations for goal C
reward per episode
number of episodes

SARSA_lambda average rewards after 25 trials over 50 iterations for goal C
reward per episode
number of episodes

SARSA_lambda average rewards after 25 trials over 50 iterations for goal C
reward per episode
number of episodes

SARSA_lambda average steps after 25 trials over 50 iterations for goal C
steps per episode
number of episodes

SARSA_lambda average steps after 25 trials over 50 iterations for goal C
steps per episode
number of episodes

SARSA_lambda average steps after 25 trials over 50 iterations for goal C
steps per episode
number of episodes

SARSA lambda for goal C lambda0.0
G

SARSA lambda for goal C lambda0.3
G

SARSA lambda for goal C lambda0.5
G

SARSA_lambda average rewards after 25 trials over 50 iterations for goal C
reward per episode
number of episodes

SARSA_lambda average rewards after 25 trials over 50 iterations for goal C
reward per episode
number of episodes

SARSA_lambda average rewards after 25 trials over 50 iterations for goal C
reward per episode
number of episodes

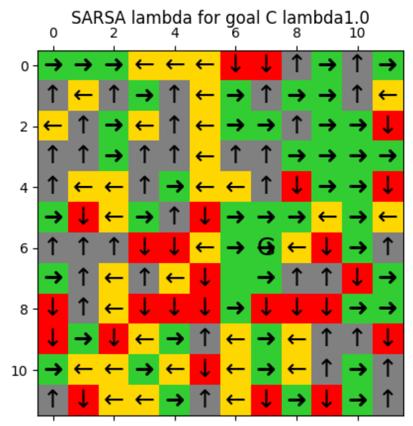




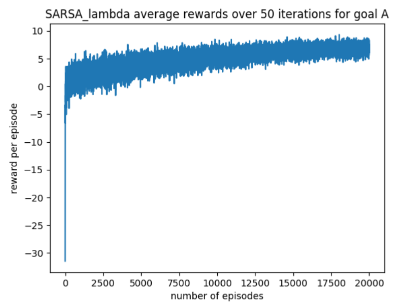


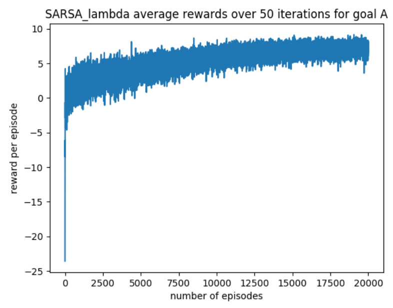


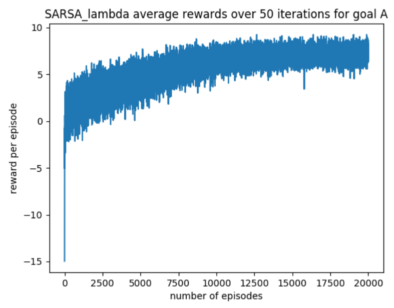


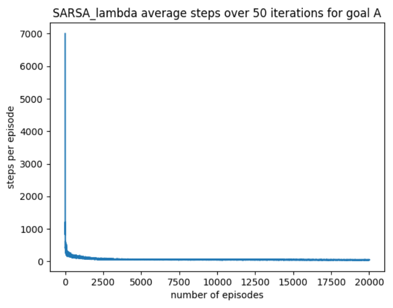


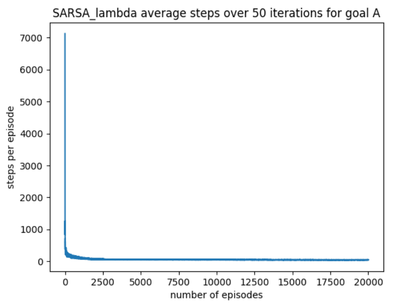
For Goal C the average reward as compared to above cases is very small also large number of steps is observed.

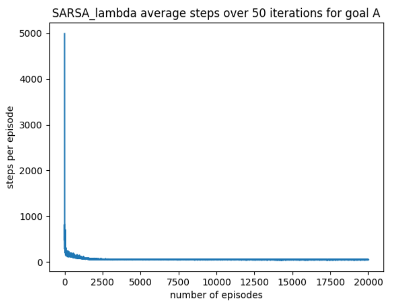


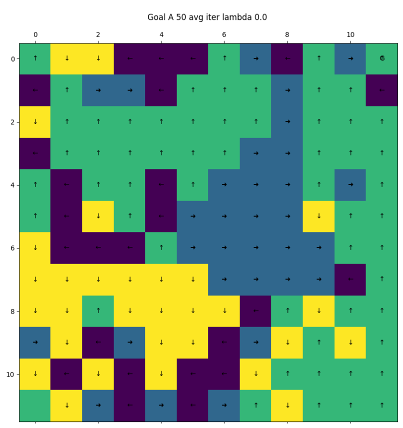


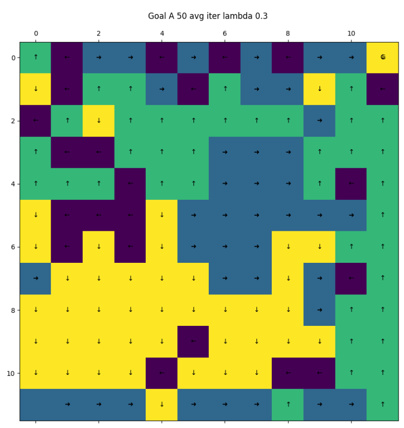


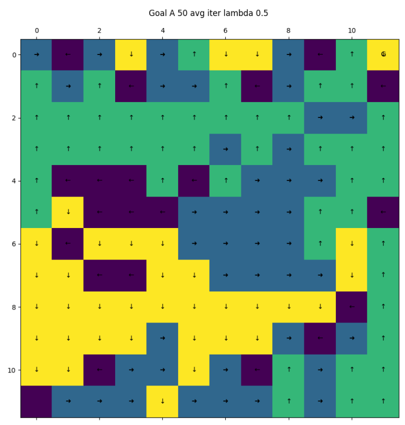




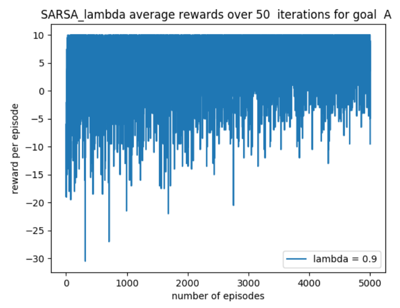


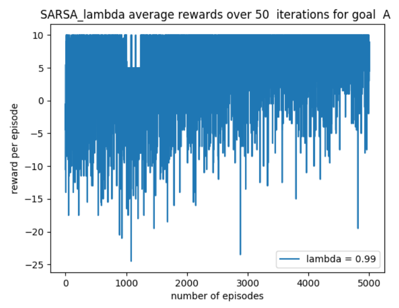


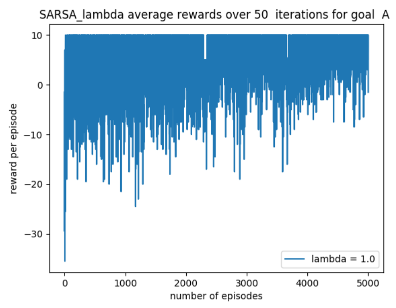


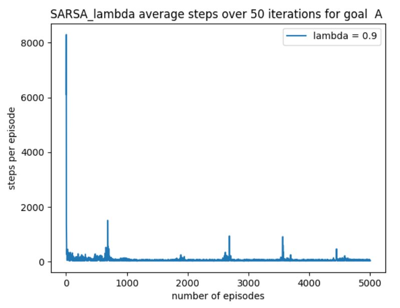


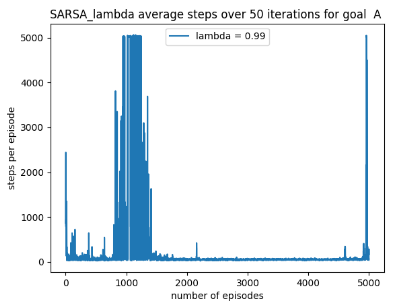
For goal A here the trials are run for 20000 episode for lambda 0,0.3 and 0.5. For all these values the average reward and number of steps is same as can be observed from the plots. The policies are different but have same average values.
As the time for each iteration of SARSA lambda is taking about 12 hours for one value of lambda in further experimentation the number of episodes are reduced to 5000. Also it was observed through experimentation while following SARSA lambda updated the policy is stuck in a loop i.e. while following a current policy when updated in a current state after some iterations the agent returns to the same state and the episode continues forever. To avoid this a step breaker is introduced as all the state here are 144 if for a current episode if number of steps are more than 5000 steps then break the loop and start with the next episode. Thus below plots are different than above for SARSA lambda case.

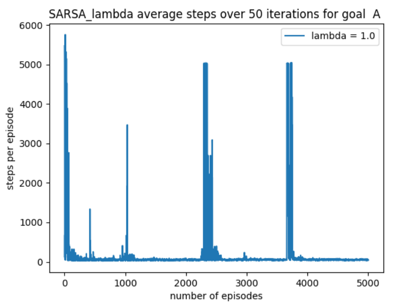


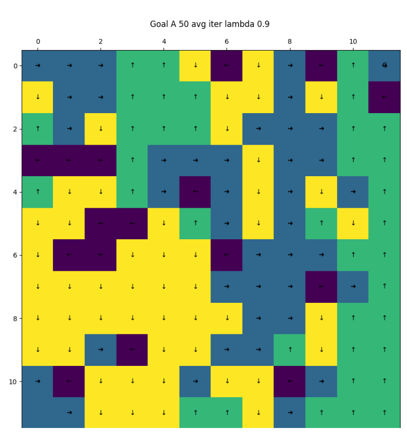


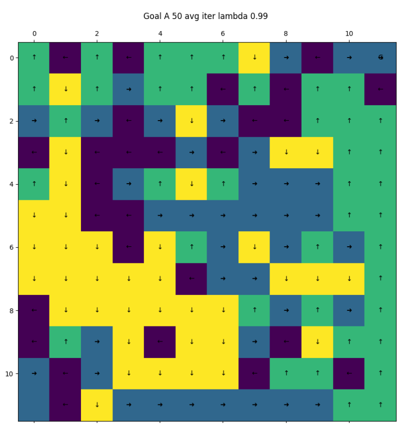


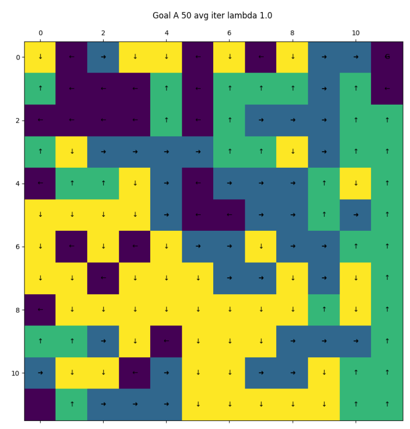




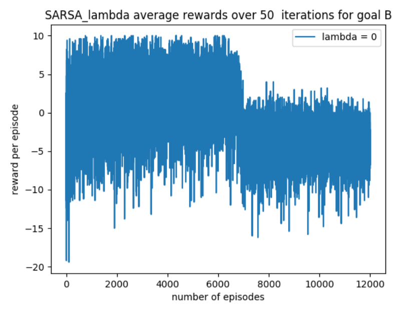


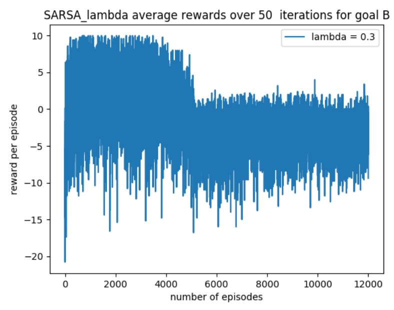


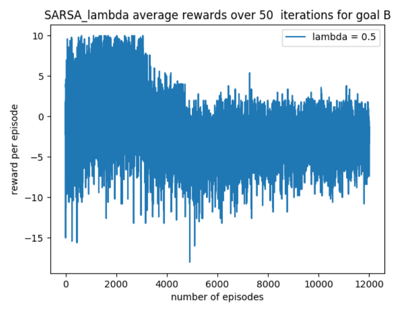


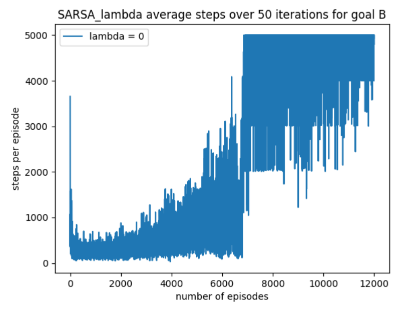


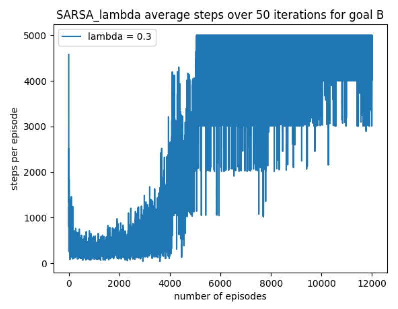
The rewards are observed to be increasing as lambda increases for above goal A.

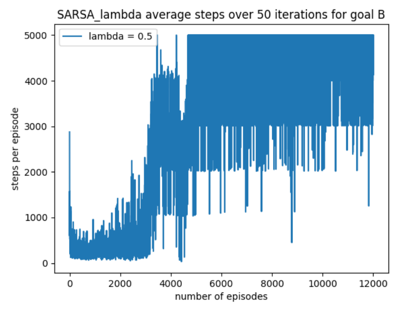


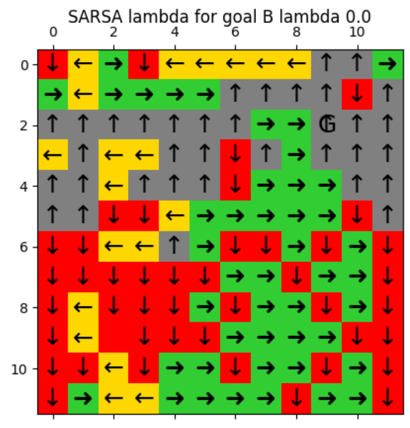


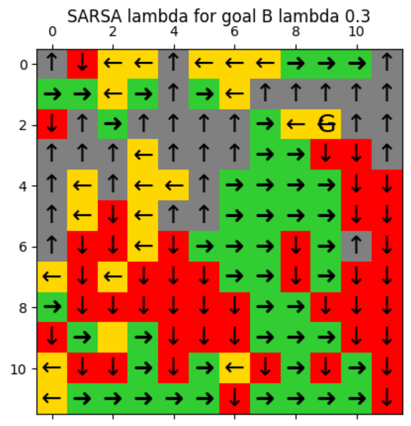


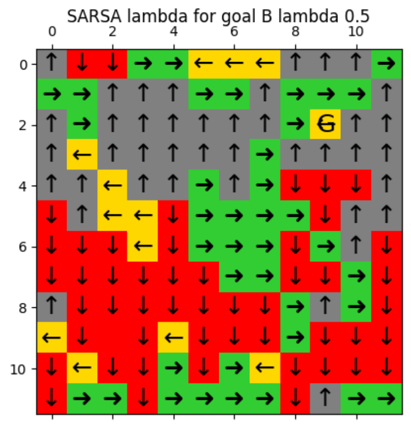


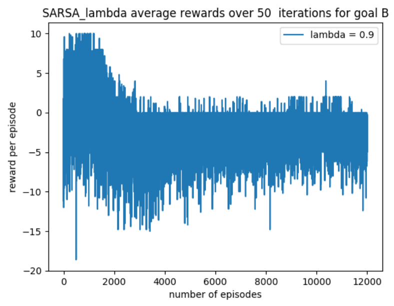


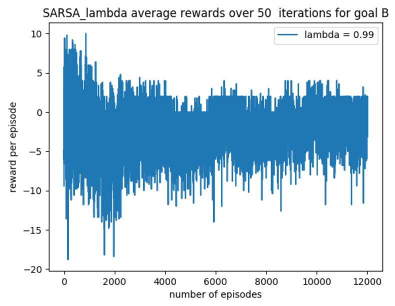


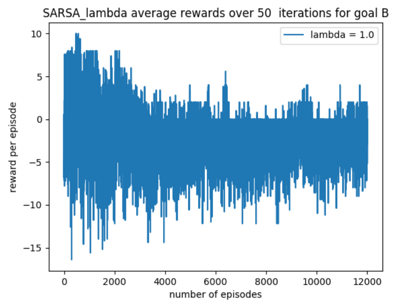


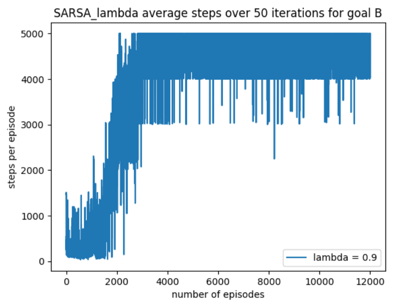


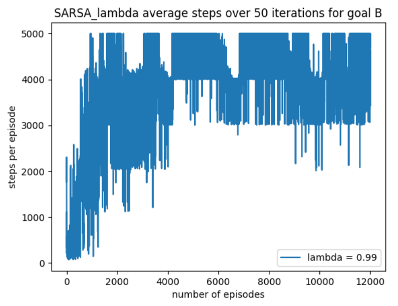


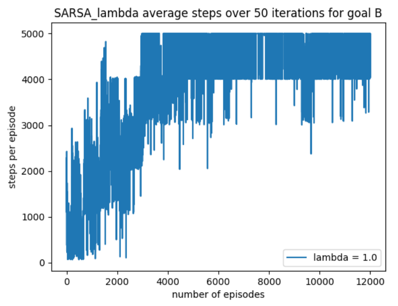


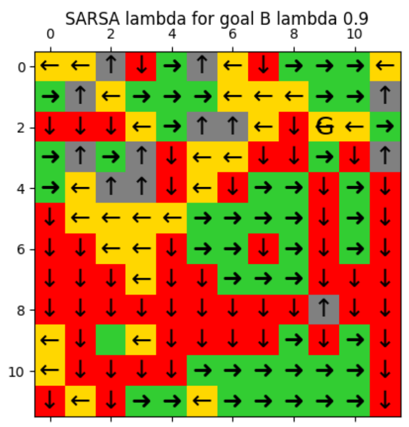


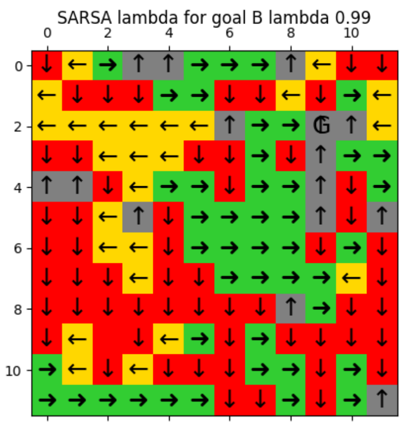


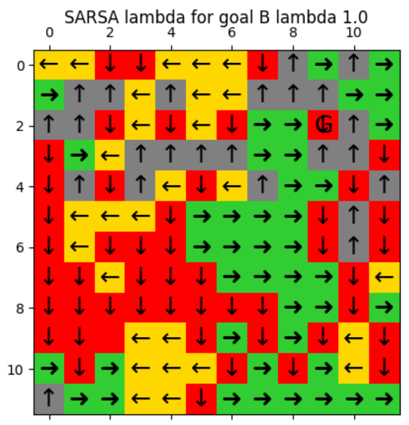




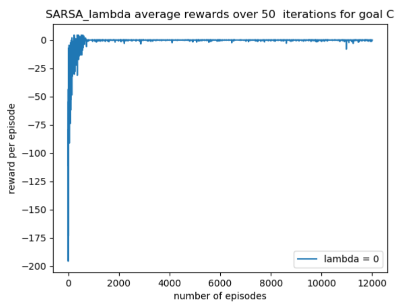


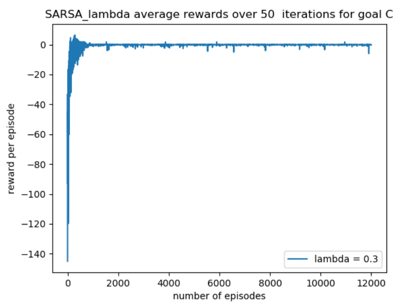


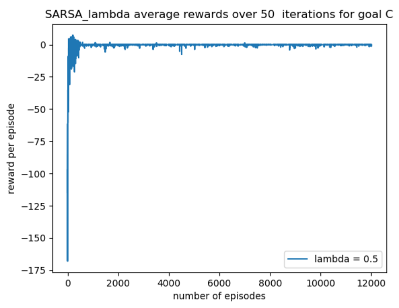


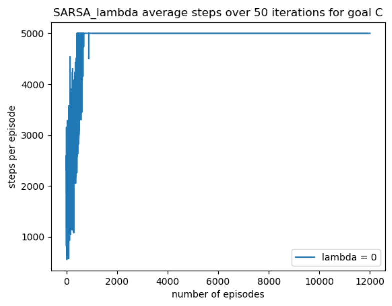


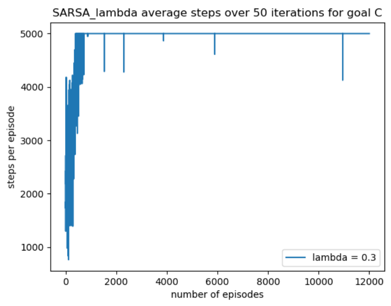
For Goal B as explained above same procedure is repeated.

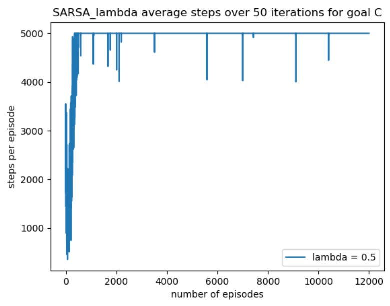


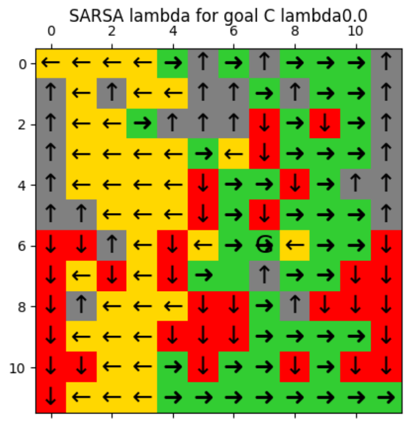


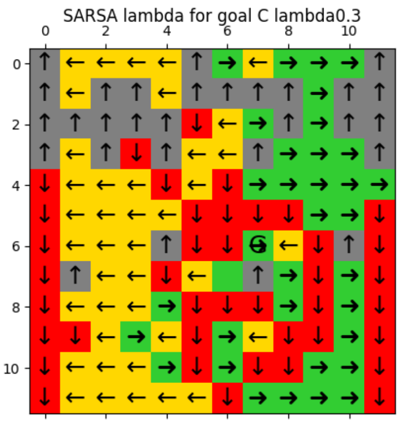


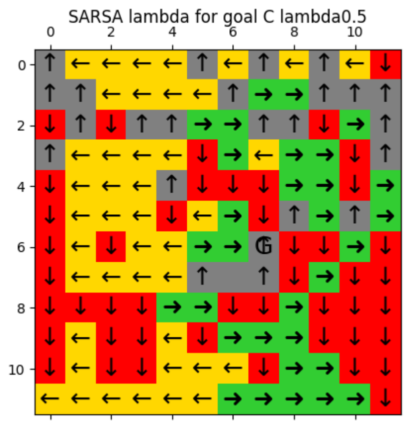

SARSA_lambda average steps over 50 iterations for goal C
steps per episode
number of episodes
lambda = 0

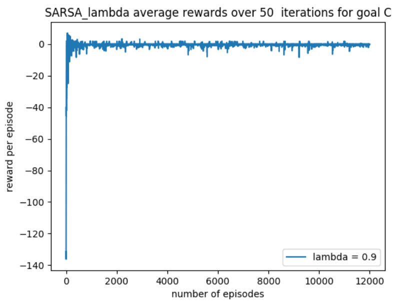

SARSA_lambda average steps over 50 iterations for goal C
steps per episode
number of episodes
lambda = 0.3

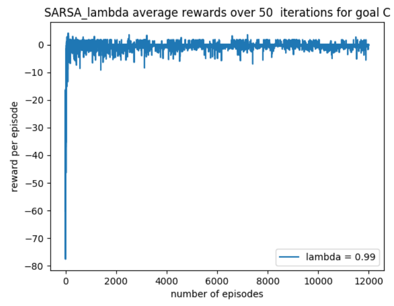

SARSA_lambda average steps over 50 iterations for goal C
steps per episode
number of episodes
lambda = 0.5

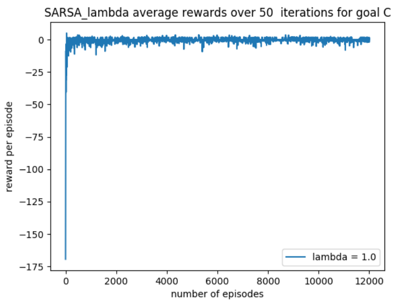

SARSA lambda for goal C lambda0.0
G

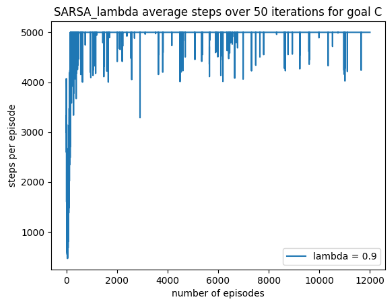

SARSA lambda for goal C lambda0.3
G

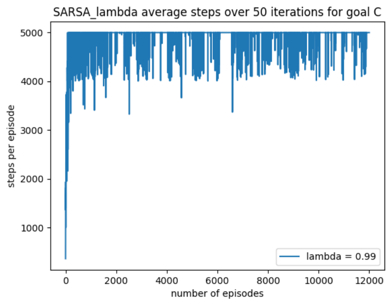

SARSA lambda for goal C lambda0.5
G

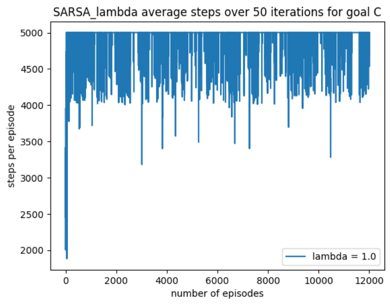

SARSA_lambda average rewards over 50 iterations for goal C
reward per episode
number of episodes
lambda = 0.9

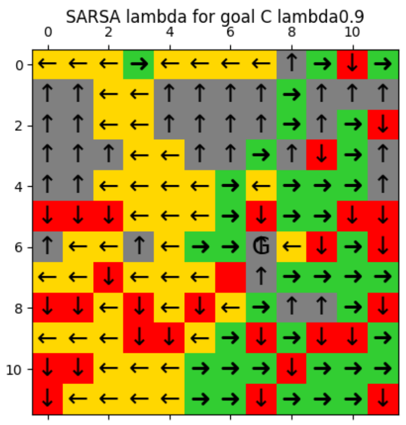

SARSA_lambda average rewards over 50 iterations for goal C
reward per episode
number of episodes
lambda = 0.99

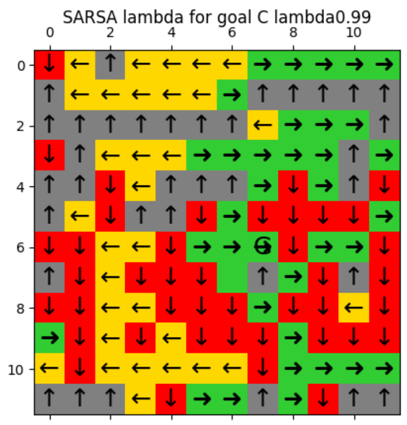

SARSA_lambda average rewards over 50 iterations for goal C
reward per episode
number of episodes
lambda = 1.0

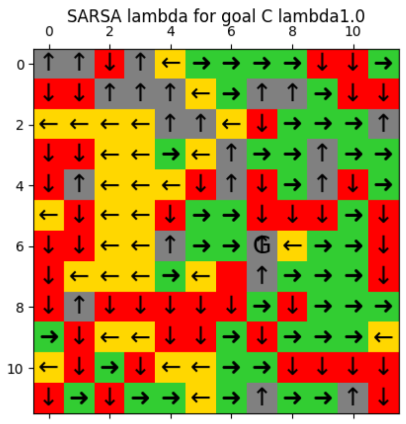

SARSA_lambda average steps over 50 iterations for goal C
steps per episode
number of episodes
lambda = 0.9


SARSA_lambda average steps over 50 iterations for goal C
steps per episode
number of episodes
lambda = 0.99


SARSA_lambda average steps over 50 iterations for goal C
steps per episode
number of episodes
lambda = 1.0



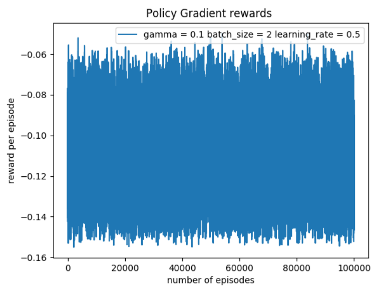


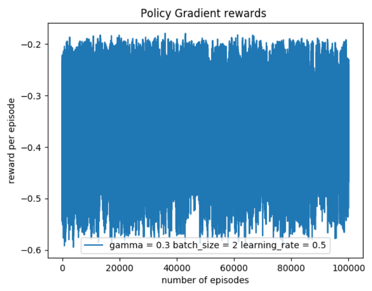


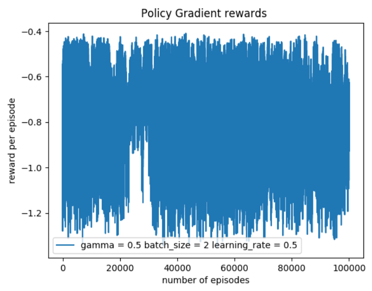
For Goal C the average rewards plot is nearly same for all values of lambda.

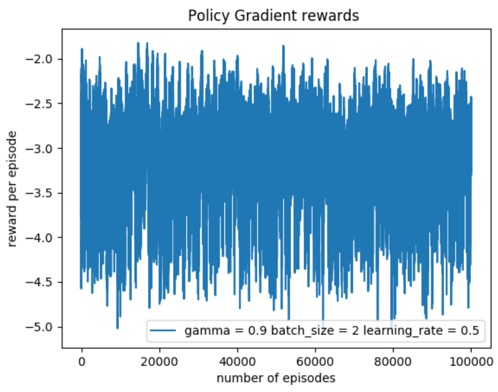
2. Carry out hyperparameter tuning: Tune the hyperparameters, like batch size, learn- ing rate, discount factor, etc. What do you observe? Can you learn a policy faster?

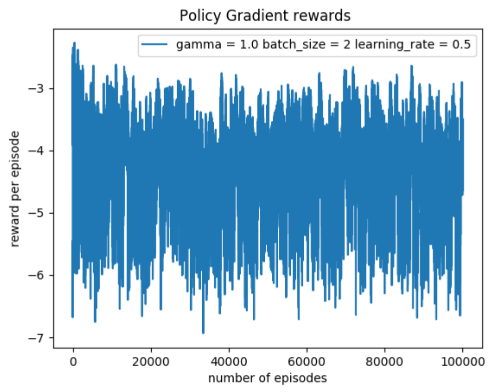
   - Tuning learning rate.
     Its obvious if the learning rate is set to zero policy doesn't change and improve. The hyperparameter learning rate was fine tuned in between 0 and 1 and final value of 0.5 was chosen for further experimentation. It was observed that the policy converged and rewards reduced as number of episodes increased.
   - Tuning discount factor
     Here gamma is changed from 0.1 to 0.3, 0.5, 0.9 and 1.0.





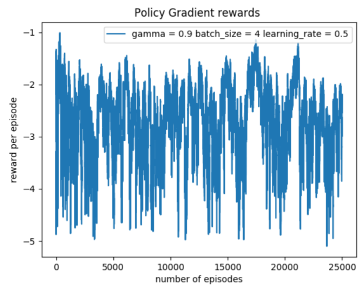


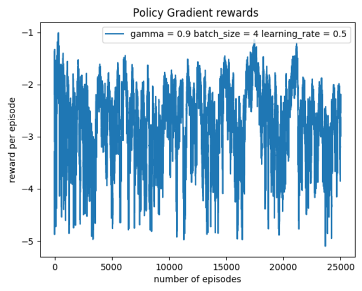


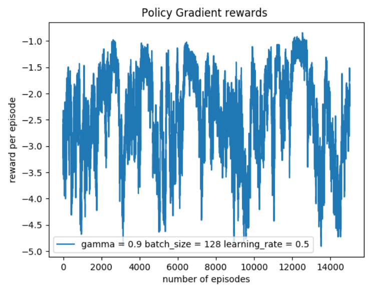


From above plots it can be clearly observed the average reward in case of gamma equal to 1.o is in the range from negative 3 to negative 6 for over 100000 episodes. however when its value is decreased to 0.9 range is $[-4.5to-2]$, for 0.5 it is $[-1.2to-0.4]$, for 0.3 it is $[-0.6to-0.2]$ and for 0.1 it is $[-0.16to-0.06]$. Thus as the value of gamma decreases the agent travels closer to the goal and average reward here the distance from the center is reduced. Thus policy is learnt faster if gamma is reduced.

- Tuning batch size
  Here gamma is chosen 0.9 to see for different batch size does average reward changes.

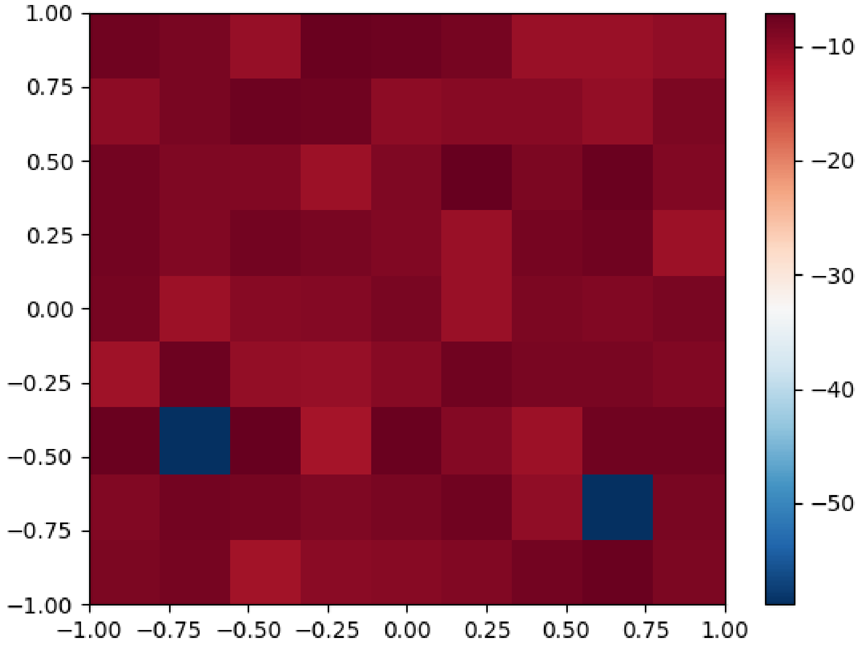


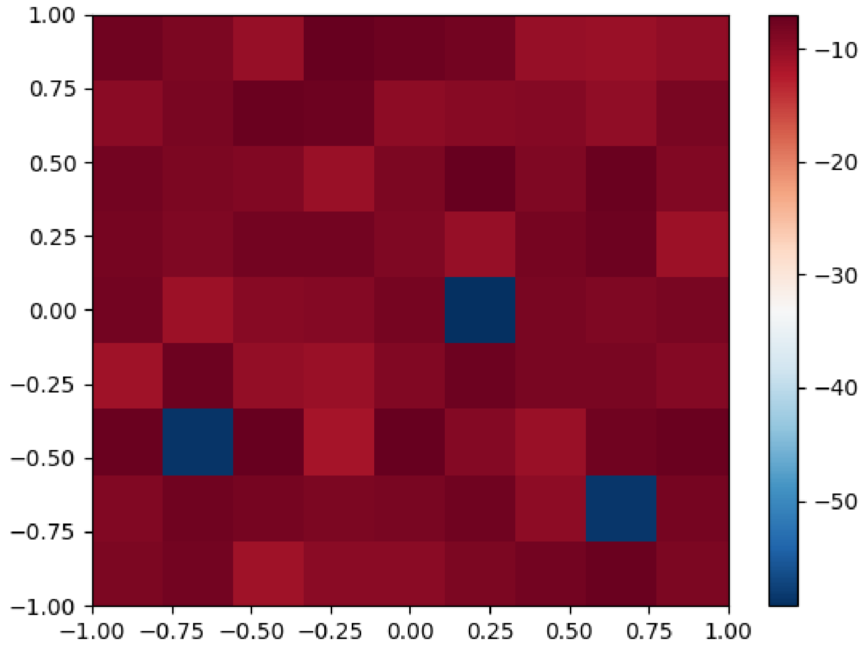




As the batch size changes from 4 to 32 to 128 over 10000 episodes keeping other parameters constant no visible changes in average reward is observed.

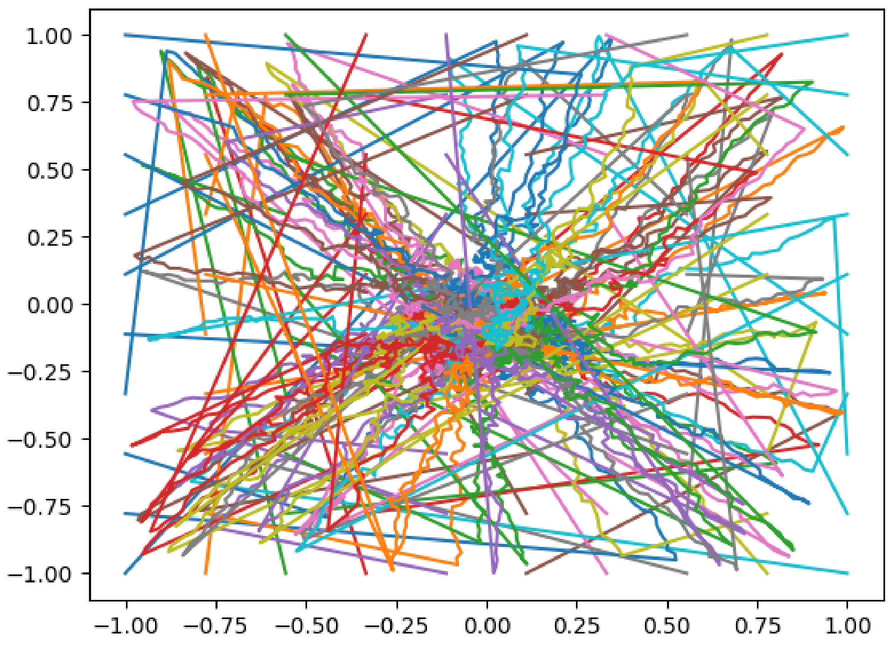
3. Can you visualize a rough value function for the learned policy? If no, too bad. If yes, how would you go about it? Turn in the plots.

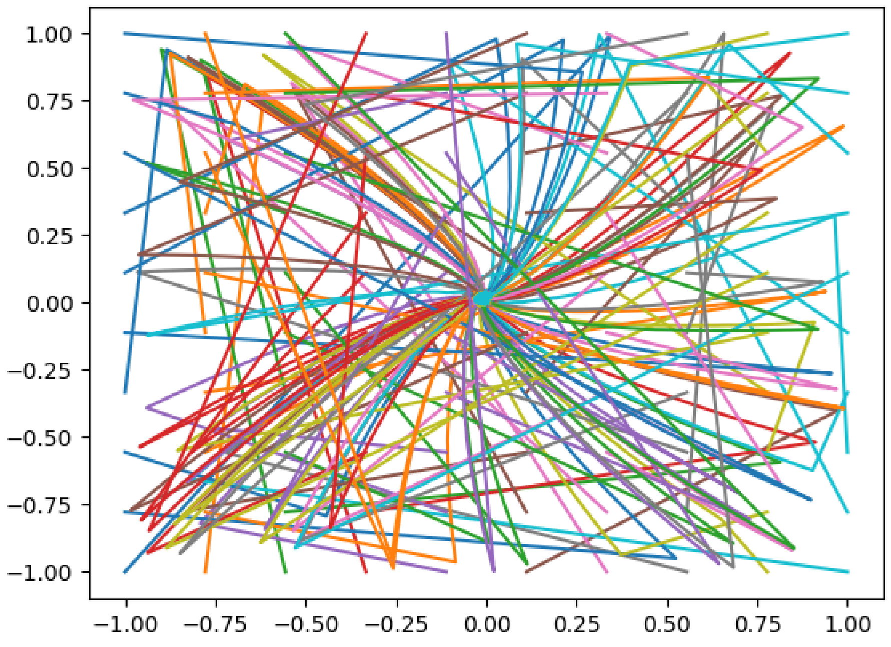




Here a meshgrid is created for the given range so that continuous world is discretised. For each of the states the policy is followed for trajectory length or if terminal state is reached and discounted reward is calculated. This can be termed as value function for each state. The above plots are value function (heatmap) for chakra and vchakra respectively.

4. What do you observe when you try plotting the policy trajectories for the learned agent? Is there a pattern in how the agent moves to reach the origin?





Similarl to above a meshgrid is generated and from there following the obtained policy trajectories are recorded and plotted. It is observed that in most of the states the agent travels in a jiggly fashion in first case and in a curved path for the second respectively.