CS 7641 - Markov Decision Processes

Author: Swapnil Ralhan(sralhan3@gatech.edu)
Date: Nov 30, 2014

Abstract

In this paper, we explore Markov Decision Processes. We take 2 MDPs, one with a small(~10) number of states and a large(~600) number of states. To solve these problems, we compare 3 approaches - Value Iteration, Policy Iteration and Q-Learning. We consider how long each of the approaches take to converge, the solutions they converge to, and different approaches used within each problem(different reward functions and terminal functions, as well different values for the various variables).

Problems

The following are two variations of the traditional grid problem. They are interesting as they easily highlight the different features and applicability of the RL and MDP algorithms, as well as being simple to understand and easy to visualize. Additionally, they also find direct applications in path-finding problems, such as robotics, and AI in computer games.

Small Problem

Size

The first problem we look at is a 3 by 3 grid

Obstacles

The grid has 2 obstacles - one at (0,2) and another at (1,0).

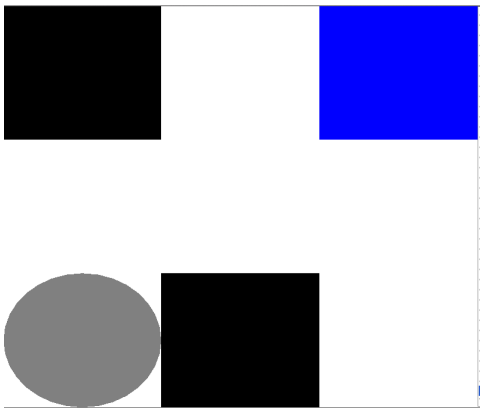
Starting Positions

The Agent starts at (0,0) and the goal is at (2,2).

Reward Function

Each step was uniform cost. Entering the goal state resulting in termination, thus the agent wanted to end the game as soon as possible. The cost of each step was 1.0

The representation of the grid looks as follows:



Large Problem

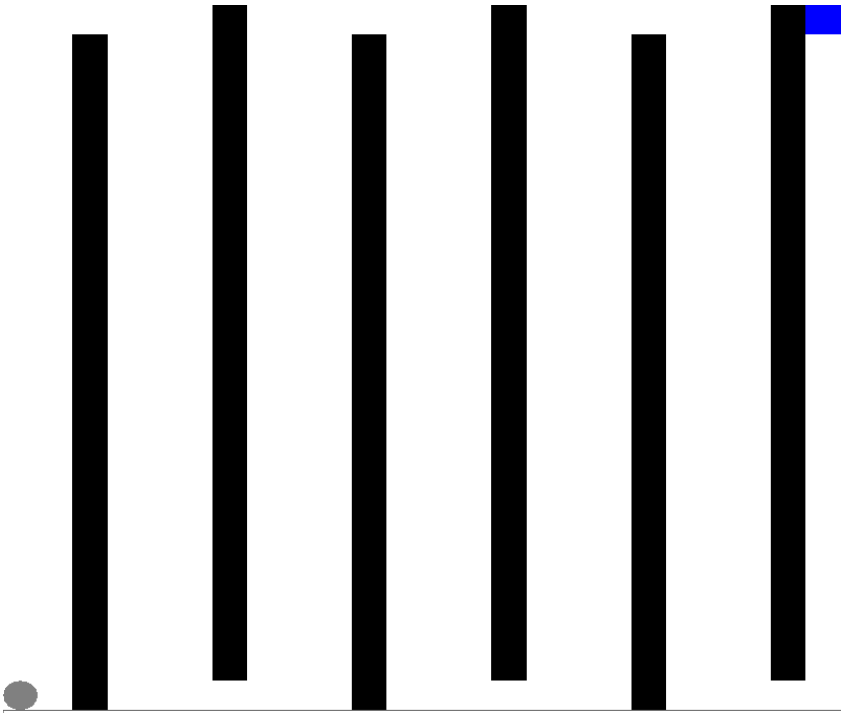
Size
The second problem we examine is a 24 by 24 grid.

Obstacles
There are 6 columns of obstacles - At x index 2, 10 and 18 we have a column from y index 0 to 22, and at y indices 6, 14 and 22, we have a column from y index 1 to 23.

Starting Positions
The Agent starts at (0,0) and the goal is at (23, 23).

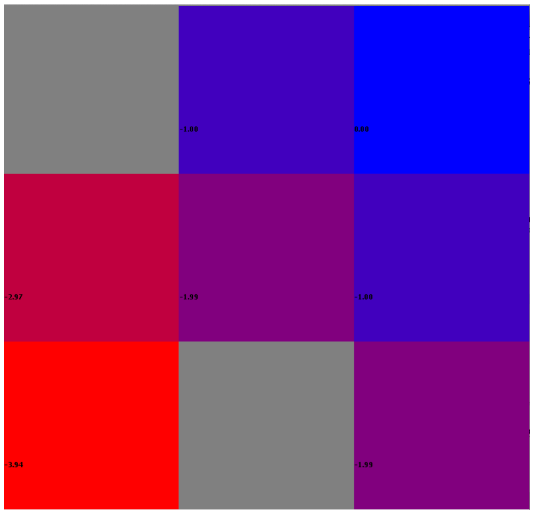
Reward Function
Each step was uniform cost. Entering the goal state resulting in termination, thus the agent wanted to end the game as soon as possible. The cost of each step was 1.0

The grid looks as follows:



Problem 1 - Small Grid

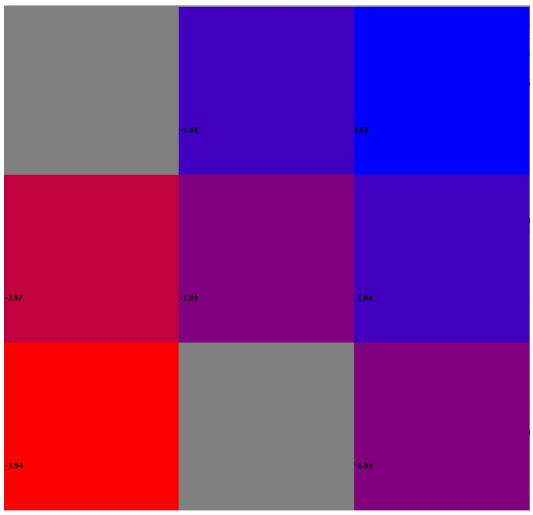
Value Iteration
Value Iteration solved the problem within 4 Passes. The values assigned to each of the states are shown in the diagram below.



The solution obtained was optimal, taking 4 steps to the solution state.

Policy Iteration

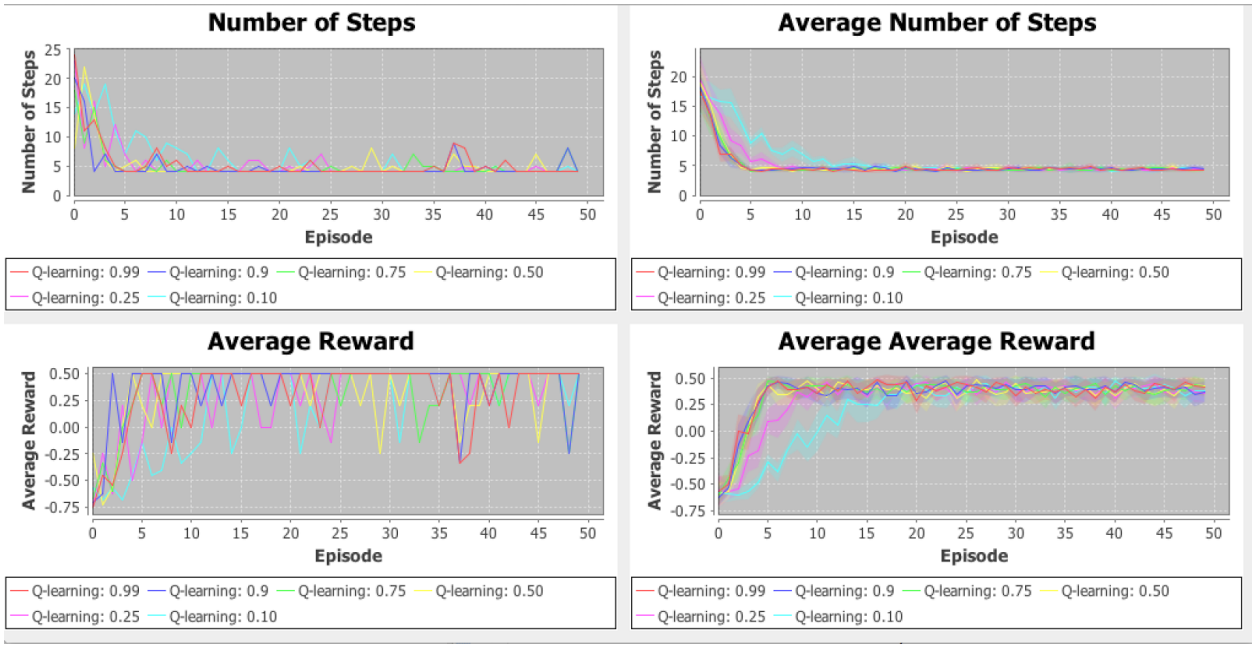
Policy Iteration also solved the problem in 4 Iterations, with the values assigned to each state being the same as those assigned by Value Iteration:



The solution obtained was also optimal.

Q-Learning

Q-learning converged in approximately 2 iterations. The solution obtained is optimal as well(taking 4 steps).
Trying out different values for the discount rate, we see that lower discount values slow down the learning, however, the optimal is achieved within 20 iterations for every single value of gamma.



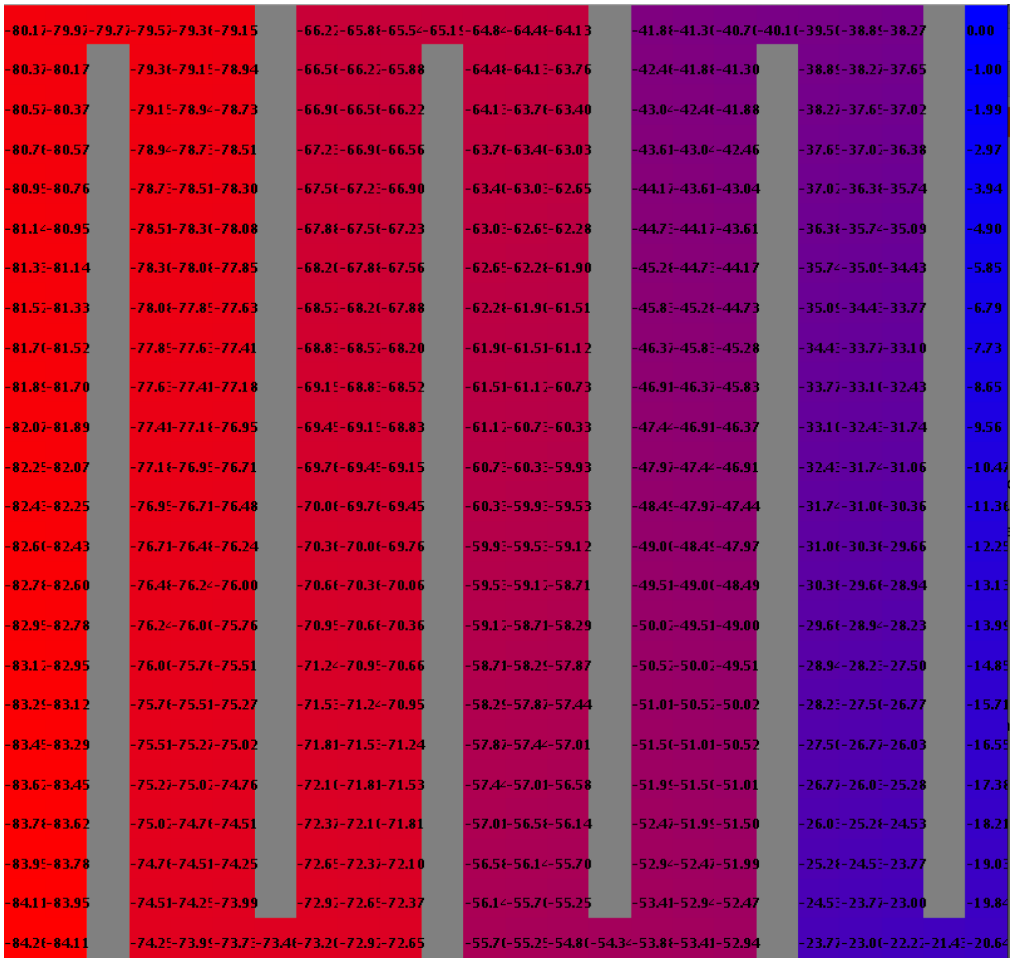
Comparison

Among the 3 algorithms, each of them have very similar performance, reaching convergence with a few iterations. While Q-learning is marginally faster in terms of the number of iterations, VI iterations run much faster than either Q-learning or Policy Iteration.

Problem 2 - Large Grid

Value Iteration

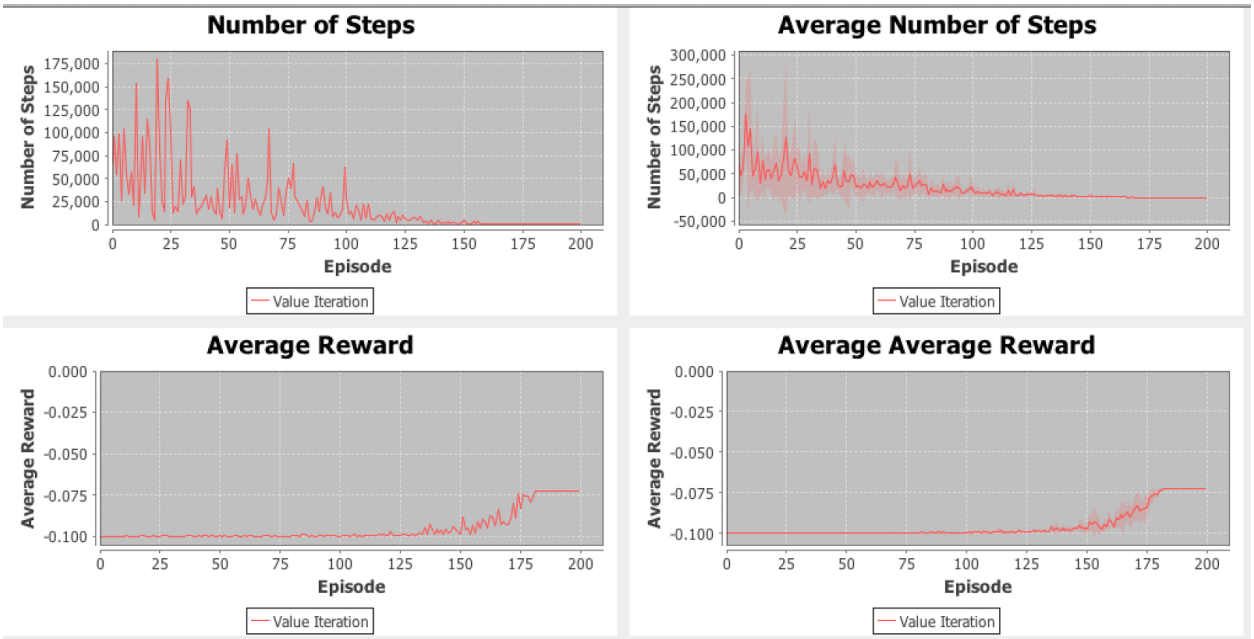
Value Iteration takes 184 Iterations to find the optimal solution of length 184. The values produced for each location are shown below:



These values are produced by having a cost of -1 per step, and the terminal position being the only way to end.

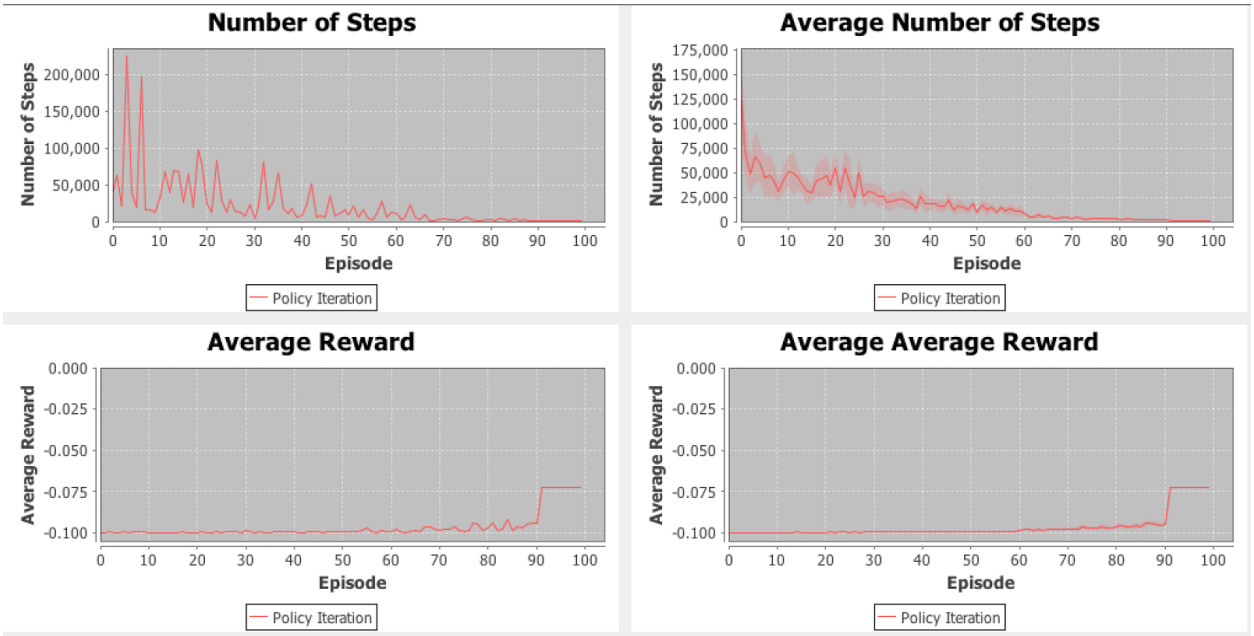
In the following graphs, we show the number of steps suggested by value iteration to get to the solution state, as a factor of the number of iterations taken by VI. As you can see, it starts off by randomly exploring the space, taking up 250,000 steps in some cases. Eventually, as the number of iterations increases, the solution begins to converge. At exactly 184 iterations, it has found the optimal solution.

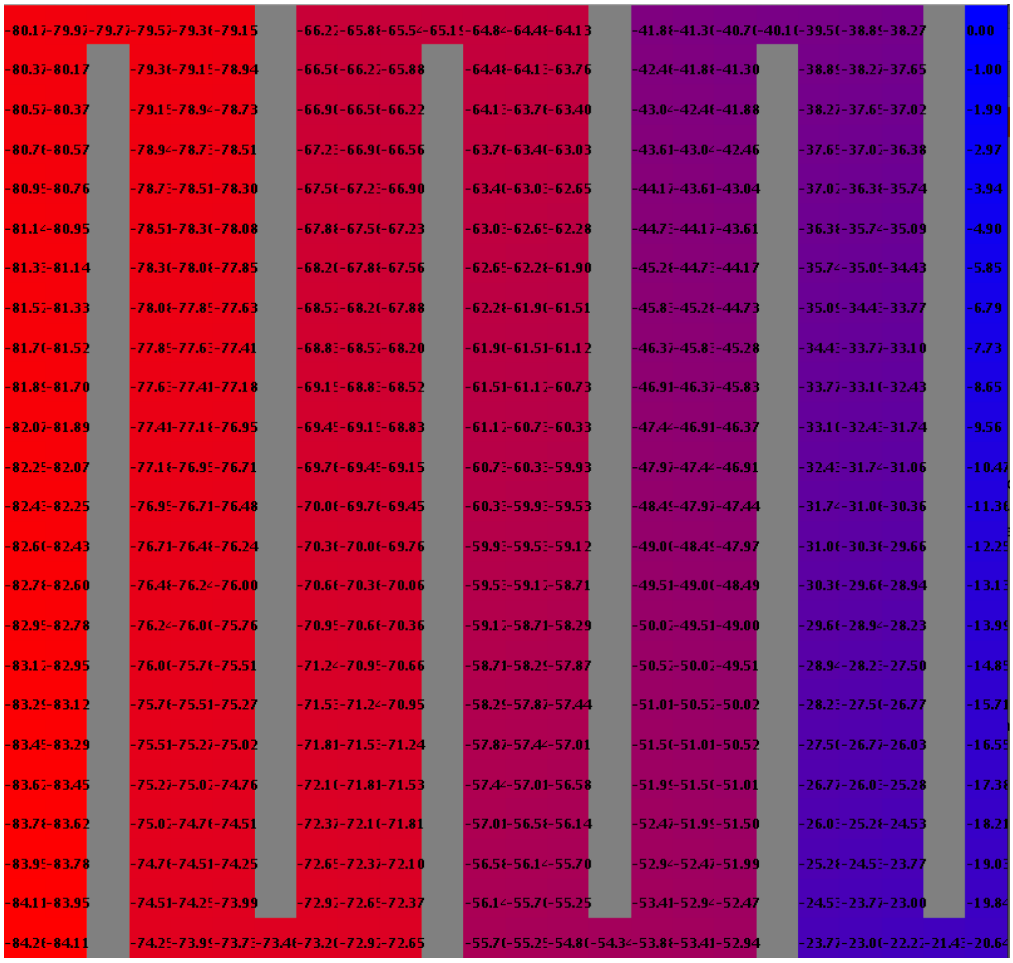
In terms of the average reward, as we can see, the reward starts rising steeply at around 160 iterations, peeking at the optimal value obtained at 184 iterations.



Policy Iteration

Policy Iteration takes much fewer iterations. Within 93 Iterations, Policy Iteration is able to find the optimal solution of 184 steps. However it should be noted here that the 93 iterations takes about twice as long as the 184 iterations taken by Value Iteration(5.26 seconds as opposed to 2.50 seconds).



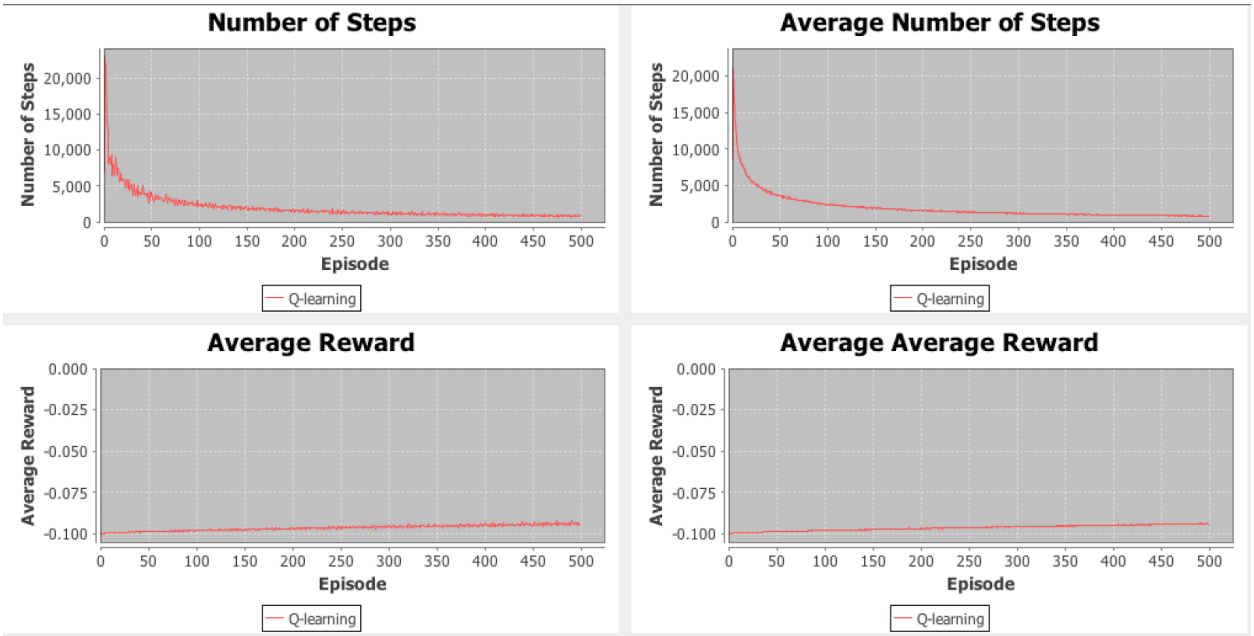


Again, we see that the values for each of the states produced by Policy Iteration are exactly the same as those produced by Value Iteration.

Here we investigated different values of gamma(discount factor). At values of 0.99, 0.95, the same optimal values are obtained, albeit at a slightly higher number of iterations. However, for discounts of 0.90, 0.80 and 0.50, we are very far from the optimal value(in the 1000s of steps), and we are unable to get to the optimal value within 100 iterations.

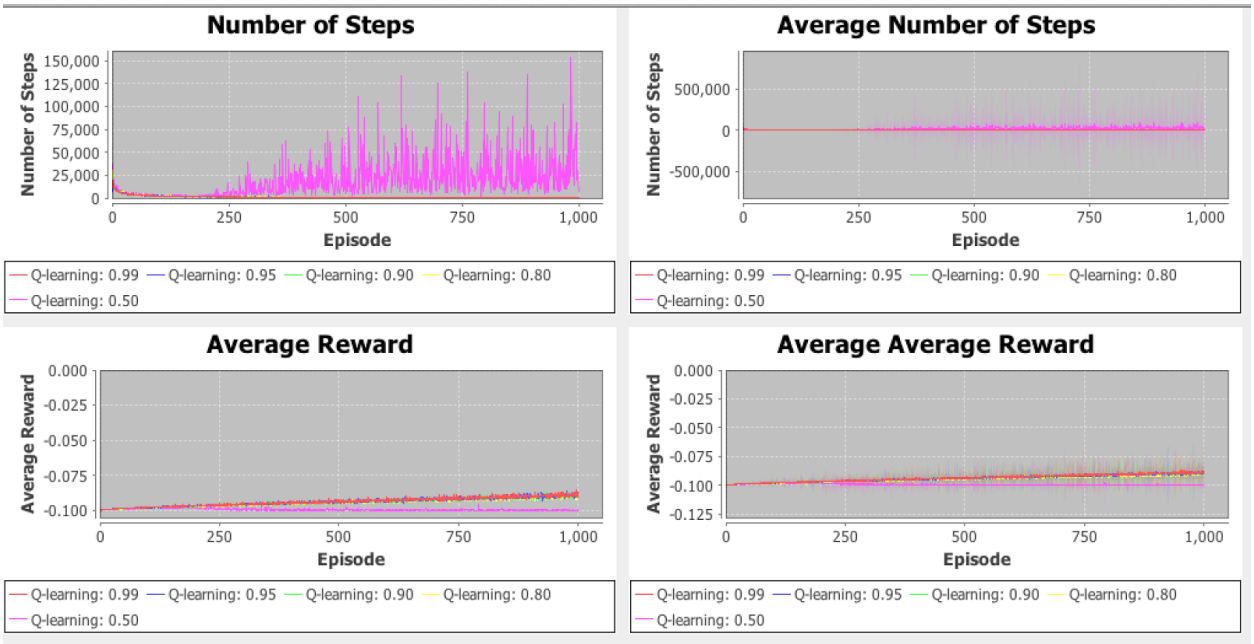
Q-Learning

Now, we tried Q-Learning on the larger problem. Trying out with a discount rate and a learning rate of 0.9, we see convergence in approximately 600 iterations. However, the iterations take a shorter time than Value Iteration or Policy Iteration, taking about 4.54 seconds to do the 600 iterations.
The convergence achieved is at 184 steps, the optimal value.
Here we plot the number of steps as a function of the episodes:



Next we experiment with different values of gamma, the discount rate. As we can see, for lower discount rates, q-learning sometimes fails to converge. Specifically, we can see that for gamma=0.50, we start diverging at about 250 episodes. The minimum value achieved is on the order of 10000 steps for gamma=0.50.

For higher values of gamma, we eventually converge, but taking more iterations for smaller gammas:



Next we investigate different learning rates. As we can see, different learning rates cause convergence to take longer. A learning rate of 0.25 causes convergence in over 800 episodes, and a learning rate of 0.10 does not come close to convergence in over 1000 episodes.

It should be noted though that even though we see reward for learning rate of 0.9 top off at around 300 episodes, this is at a suboptimal path of approximately 190-210 steps. Since the value of -1 is assigned to each step, the reward difference between 200 steps and 184 steps is minimal

We also explored different reward functions. We tried assigning values between 0 and 5000 to the goal state, and between -1 and -0.1 per step. Placing particular emphasis on the ratio of step cost vs goal reward, we did not see any significant change in the performance of the q-learning algorithm.

Conclusion

Thus we have evaluated the performance of the 3 algorithms - Value Iteration, Policy Iteration and Q-learning on the 2 different problems. As we have seen, on both the problems each of the algorithms achieves the optimal step value. While Policy Iteration takes a fewer number of iterations for the larger problem, Q-learning converges in a smaller number of iterations on the smaller problem. However, iteration time is different as well, and Value Iteration trumps the other algorithms when it comes to time taken to achieve the optimal value.

We have also seen, that choosing the right number of minimum iterations, learning rate and discount rate(gamma) is very important as well. We may fail to converge(if choosing < 100 iterations for Value Iteration), or even diverge(if choosing too low a discount rate for Q-learning for instance). Thus, it would be an interesting exercise to plot these values for much larger number of episodes, to see if we see any divergence in the future. For instance, if we stopped Q-learning at gamma=0.50 at 200 episodes, we would mistakenly conclude that the algorithm has converged.

An additional follow-up could be to investigate different problems of the gridworld, using a fixed(such as the alternating walls of the large problem). We can see the evolution of the tuning parameters as well as the number of iterations for convergence.