MP 1 Part 2

Problem 1: In WorldWithoutThief

(a) With the discount factor set to 0.9, the learning rate to 0.1, and the ϵ to 0.0, after $1,000$ episodes of $10,000$ steps, the reward for each episode is 0. This is because the lack of random actions prevents the robot from moving onto a tile with a # on it. This means that the robot is unable to obtain any positive or negative reward.

(b) After setting ϵ to 0.1 and keeping all other setting the same as before, it takes about 16 episodes for the reward to rise from -22 to 0. From there, the reward gradually rises to about 150. This is because initially, the low ϵ value caused the robot to behave very similar to part a. However, since it is non-zero, it eventually crosses onto a tile with a # on it. This gives it an opportunity to learn about better delivery routes that produce higher reward.

(c) Using a larger ϵ value of 0.5 and keeping all the other settings the same, causes the average episode reward to be between -10 and -30. This is because while the robot might learn better routes, it is choosing a random action half of the time. This prevents it from taking the best route it knows of, causing it to make mistakes and get a negative reward.

Problem 2: In WorldWithThief

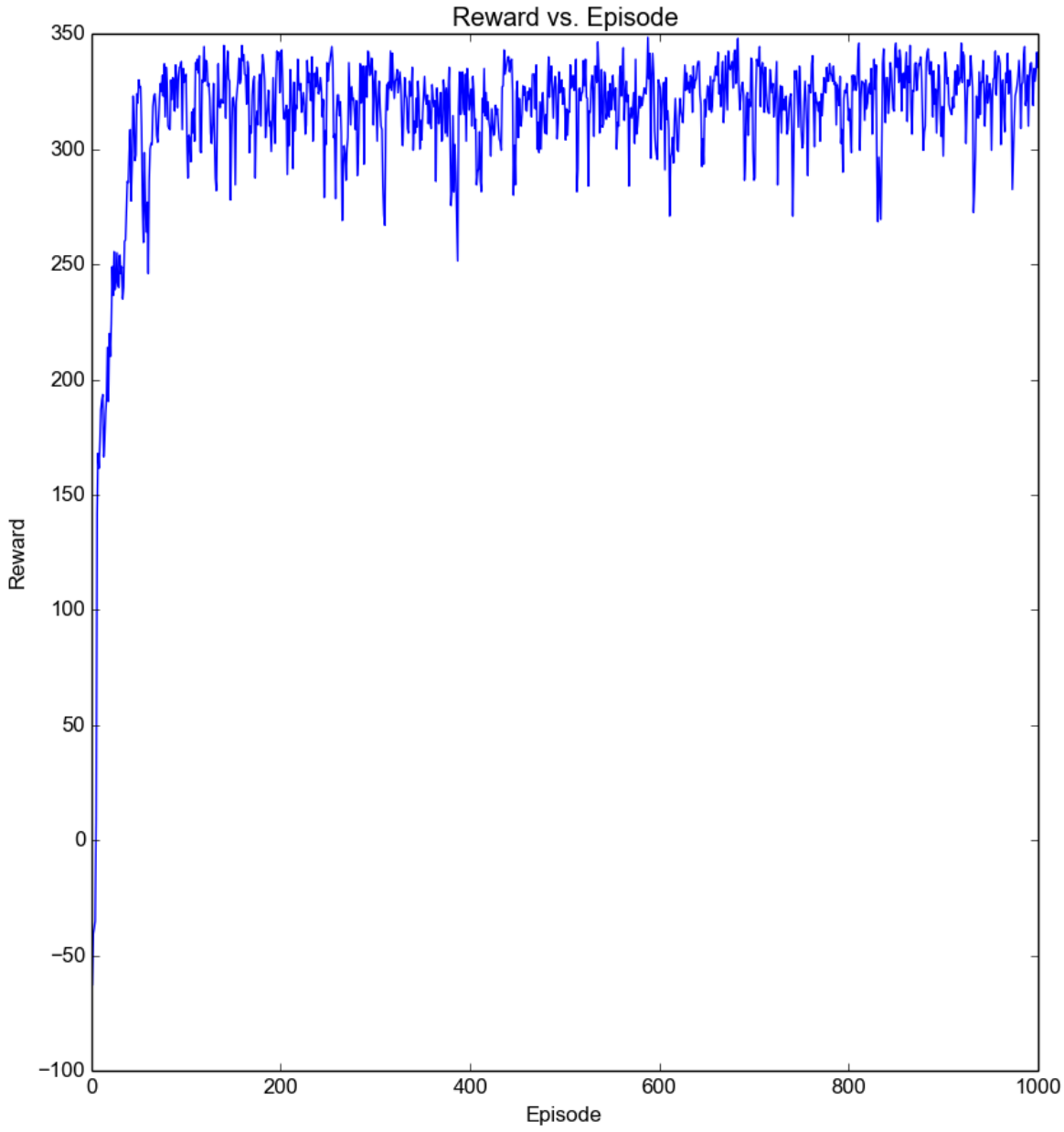
(a) Setting $\epsilon = 0.05$, keeping all the other settings the same, and setting $knows_thief$ as n results in negative rewards around -15. This is because it does not have the extra states to distinguish the position of the thief. As a result, the robot is unable to learn to avoid the thief, causing it to constantly acquire a negative reward.

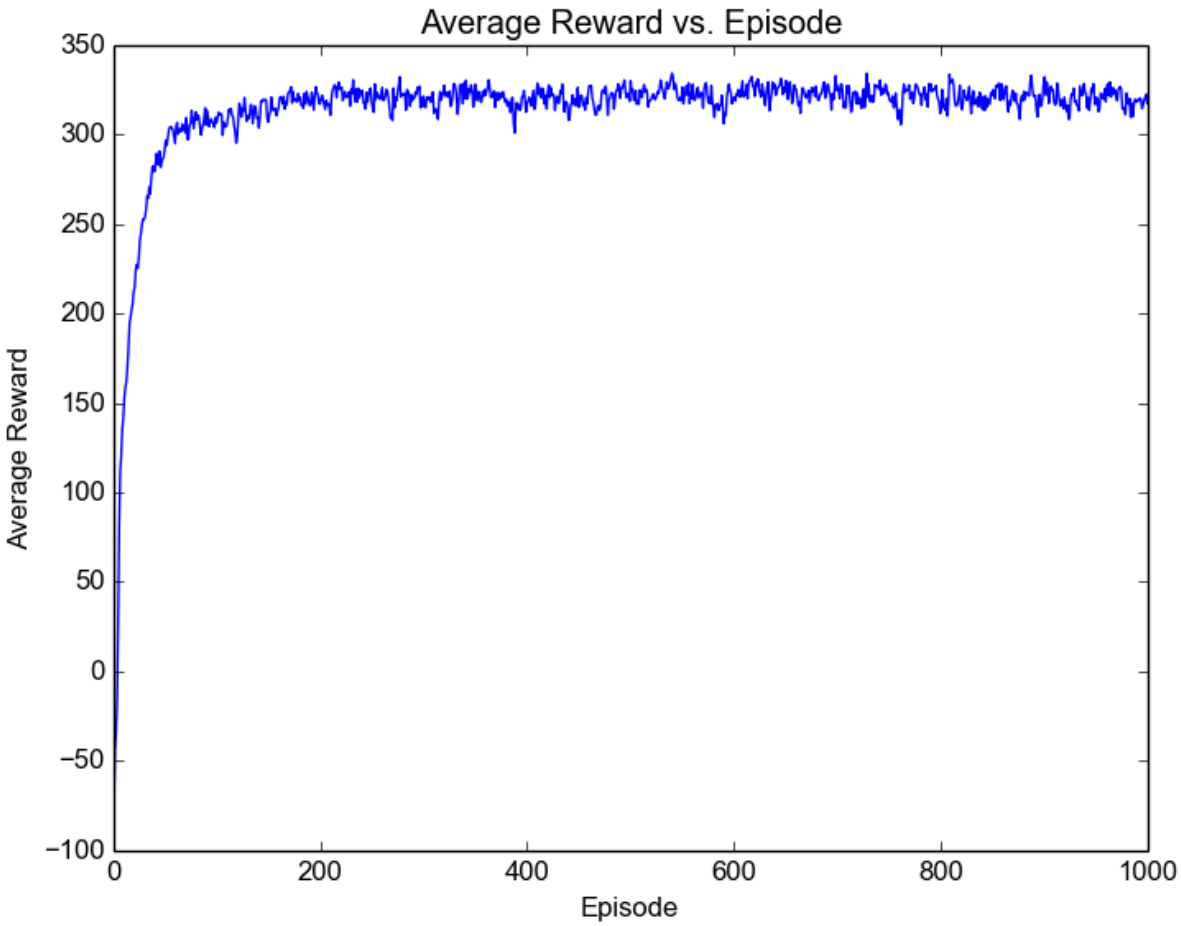
(b) With $knows_thief$ set as y, the performance improves drastically. The reward quickly rises from -75 to nearly 300. This is because it is now capable of learning about the thief and avoiding it. As a result, it can learn high reward delivery routes that avoid the thief.

(c) I started by adjusting the ϵ exploration probability. I knew that if it was too low, it would cause a slow convergence into a small good region around π^* and if it was too high, it would quickly converge into a large poor region around π^* (Lec7-8.pdf). Using this knowledge, I manually adjusted this value, while keeping the discount and learning rates constant until I found what I determined to be the optimal value of 0.0075. I then started to adjust the learning rate while keeping ϵ constant. I knew that this affected the speed of convergence. Keeping this in mind, I manually adjusted this value while monitoring the rate of convergence. I determined that a value of about 0.35 was ideal.

Problem 3

Using my best parameters from 2*c*, I noticed that my reward quickly increased from a negative value to an average value greater than 300. The quick convergence and high reward are indicators that I set my constants to good values. When I repeated the procedure plotting the expected discounted reward averaged over all ten simulations, I got very similar results. The difference was that the averaged values had much less variation in reward (they varied by a max of about 25 whereas the first plot varied by almost 100 at times).


Reward vs. Episode
Reward
Episode
350
300
250
200
150
100
50
0
−50
−100
0
200
400
600
800
1000


Average Reward vs. Episode
Average Reward
Episode
350
300
250
200
150
100
50
0
−50
−100
0
200
400
600
800
1000