Machine Problem 1
Part 2: Experimentation

Rui Guo
ruiguo2@illinois.edu

September 29, 2014

Question 1.

(a) All zero rewards. The robot will act exactly as q learning. However there is a slipperiness path that block the way for robot delivery so the robot will not risk going through it by q learning.

(b) The robot has a possibility to drop a package. Otherwise, it learned how to deliver the package quickly and with least risk. The reward collected is accumulated.

(c) When epsilon is higher, the robot is more likely to go randomly. Therefore it has higher chance dropping packages by moving between the slipperiness areas. As a result, we will observe a lot of negative rewards in the policy file.

Question 2.

(a) After few simulates, the rewards shown at each episode are mostly negative. In addition, the policy simulator shown that robot tends to stuck at the top left corner. I think this is due to the robot is not aware of the position of thief. It dooesn't risk taking step into slipperiness path and thief column since epsilon is relatively small.

(b) After changing the parameter of know_thief to y. The performance is perfect. In the simulator, the robot delivered packages while avoiding thief and slipperiness path.

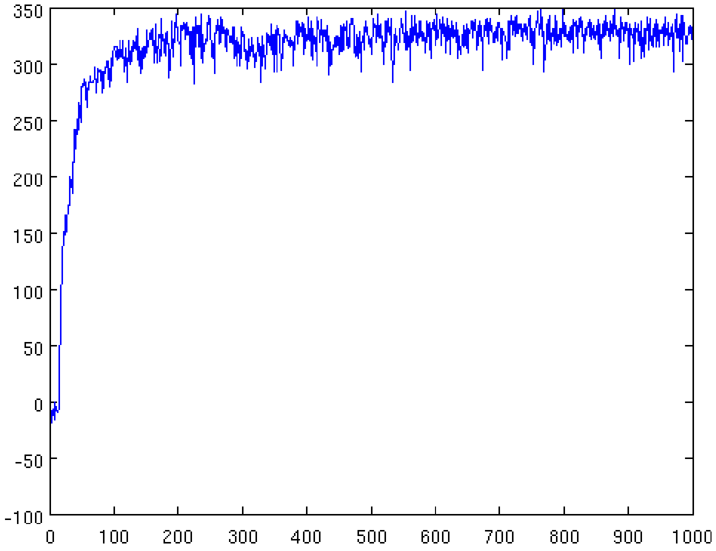
(c) $\epsilon = 0.01$; learning rate $= 0.25$. Here is how I found these values:

I first left the learning rate as 0.1 and try to find ϵ that maximizes the reward. If I increase ϵ, it will give me decreased rewards. So I modified ϵ by stepping down 0.01 each time to see the difference. I found $\epsilon = 0.01$ is a good approximation for maximum rewards.

Then I played with value of learning rate. I first decreased it and discovered that my rewards is decreasing. So I scaled discount rate up by 0.05 each time to find that 0.25 is a good approximation for max rewards. In this case, I can get rewards mostly from 330 to 345.

Question 3.

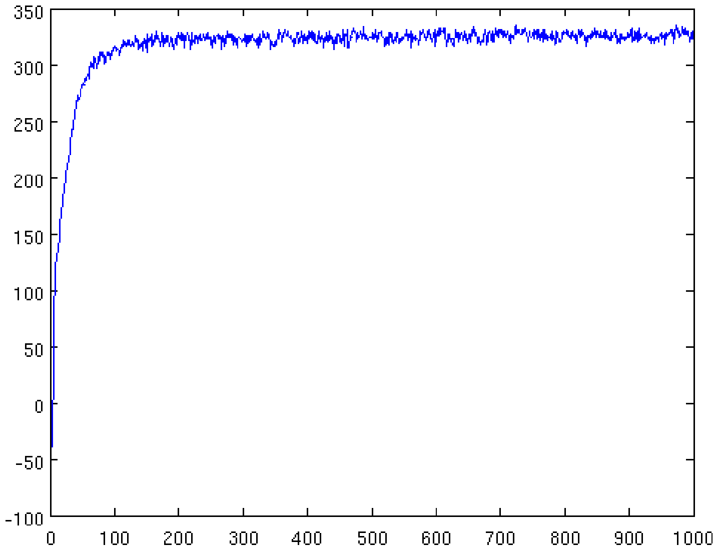
Figure 1: plot of expected discounted reward at each episode



Please refer to the plot above to graphically see how the robot learned by each episode.(the figure is obtained from matlab)

The following plot is the plot of average expected discounted reward at each episode.

Figure 2: plot of average expected discounted reward at each episode



By taken the average of ten data at each episode, we can see that the plot is more smooth than first graph. There is no sharp jump between each episode. In addition, we can get a good approximation about the reward at each episode.

For your information, here is how I generate the plot in matlab: First import 10 dataset.
A = [episodes1 episodes2 episodes3 episodes4 episodes5 episodes6 episodes7 episodes8 episodes9 episodes10];
B = mean(A,2);
figure;
plot(B);