MP 1-pt.2 CS 440

1. (a) When operating with a discount factor of 0.9, learning rate of 0.1, and ϵ of 0.0, the first episode is negative and all subsequent trials return 0.0. This could be because not returning any random actions results in the first action being taken because the expectation is 0.0, and if that action's cost is greater than its reward then the next expected return value is always negative, so no further actions are taken.

   (b) Upon repeating the same test described above but with $\epsilon = 0.1$ and taking the average output values saved into text files, I found that the average reward per episode was approximately 135. This was done by routing the reward per episode into a text file and then writing a script to compute the average value in that file. The reason that this is positive compared to the prior one is that $\epsilon > 0.0$ so there is some small degree of randomness, so there can be some lossy actions taken early on so that the algorithm can learn. This seems sensible because the first several episodes consistently have negative rewards near zero.

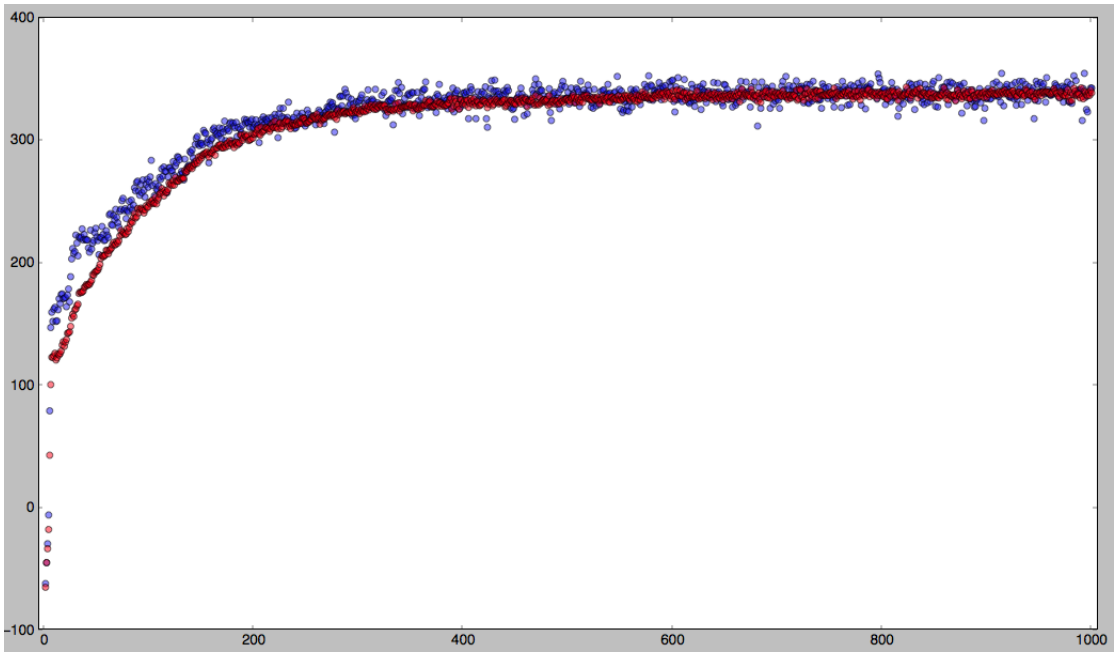
   (c) When testing with $\epsilon = 0.5$, the average reward was -30 per episode. This is most likely because $\epsilon = 0.5$ means that half of all actions are completely random. If half of actions are random and most possible actions are lossy, then that high a level of randomness results in an overall lossy strategy.

2. (a) When operating in WorldWithThief with knows_thief set as n and $\epsilon = 0.05$ the average return is -13.6. The same test in WorldWithoutThief had an average return of 146. The difference stems from a lack of knowledge about where the thief is, so the delivery bot is robbed frequently enough for delivery to become a lossy operation.

   (b) When doing the same test with knows_thief set as y, the average return is 281. This is most likely because the robot knows where the thief is, so it can avoid the theif the great majority of the time, or perhaps all the time. As the theif represents the highest penalty possible, the returns are much higher than if the robot did not know where the theif is.

   (c) I searched for the best learning rate by starting with the same learning rate and ϵ values as in part b, doubling or halving those values, choosing the value that returned the best average reward value, then testing halfway between that value and the next closest value. The values that I found seemed optimal for 1000 episodes of 10000 steps were $\epsilon = 0.0125$ and $rate = 0.125$.

3. The blue dots show the discounted reward at the end of each episode for the first trial. The red dots are the average of those values over 10 trials.



It's apparent in this that the rate of learning tails off very rapidly and approaches a slope of zero near

the 400th trial. This is readily apparent in the single trial plotted, but far more so as the mean over 10 trials is a much more concise plot with less visible noise about the mean. The reason for this tapering off is that with more trials, the average expected return changes less slowly because it is being averaged over far more data points. That's why asymptotic behavior is present.