Matthew Potok
ECE 448

MP1 Part 2

1a) If $\epsilon$ is set to 0, then the agent never finds a path to deliver packages because either it gets stuck near a wall or decides on a policy where it just goes back and forth between two states infinitely. This happens because the agent is greedy and doesn't explore the whole world which results in it staying with the safest policy it found.

1b) With $\epsilon$ set to .1, the agent explores the world while retaining the ability to learn a policy to continually deliver packages and return to the company. The reward values vary a lot between the episodes even after the agent has learnt an optimal policy because of the randomness associated with dropping the packages. Looking at the episodes output file, we can see that the agent learns relatively quickly and only after about 20 episodes it has essentially found an optimal policy.

1c) If $\epsilon$ is set to too high of a value, then the agent may sometimes learn an optimal policy where it continually delivers packages while other times it learns a policy where it gets stuck. This occurs because the agent is exploring the world a large portion of the time rather than strengthening the utility of certain actions and defining a clear policy that it should follow.

2a) If the agent is unaware of the thief, it learns a policy where it never delivers the packages and gets stuck along the wall. This occurs because this is safest policy the agent can generate and is guaranteed to never achieve a result below 0.

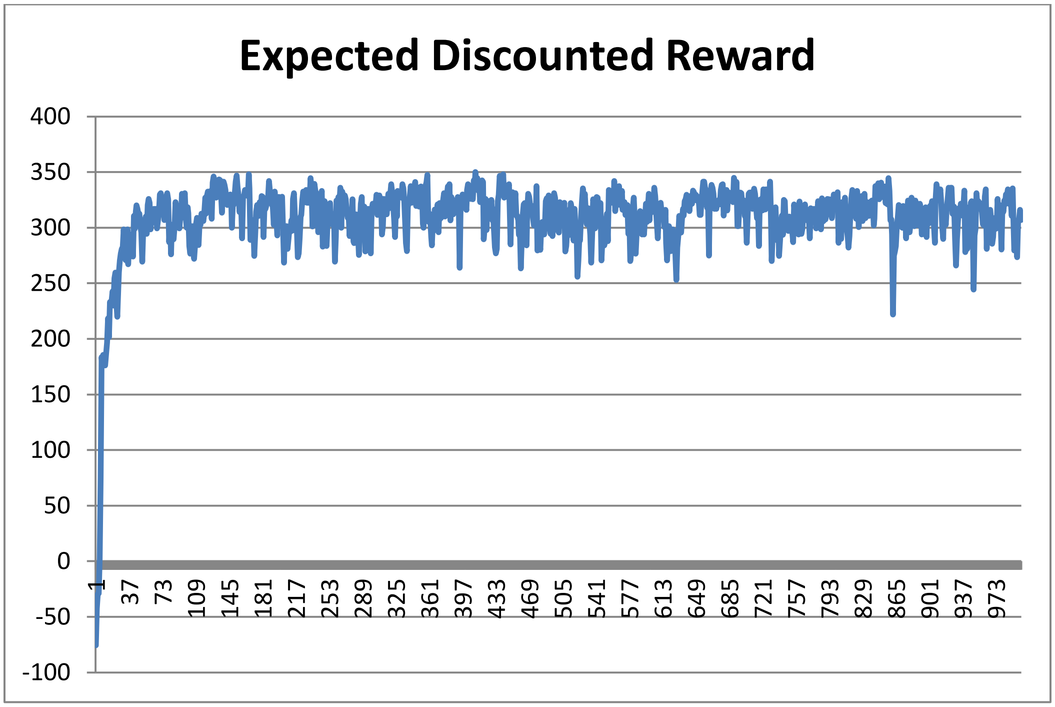
2b) By knowing where the thief is on the map, the agent can avoid her/him and generates a policy where it continually delivers packages.

2c) To find the best combination of learning rate and $\epsilon$, I created a script that would go over the episodes.txt file generated after each simulation and report certain statistics about the simulation. With this data, I created a table of various combinations and its associated minimum, maximum, average, and standard deviation. Below are the results of experimenting with the various combinations with the restrictions that the simulator ran for 1000 episodes for 10000 steps each:

| Learning rate | epsilon | min | max | average | std |
|---|---|---|---|---|---|
| 0 | 0 | -141.5 | -65.5 | -102.667 | 11.4991 |
| 0.05 | 0.005 | -64.5 | 341 | 268.49 | 62.2566 |
| 0.05 | 0.01 | -67 | 355 | 305.895 | 50.8267 |
| 0.05 | 0.025 | -68 | 334 | 290.767 | 41.5799 |
| 0.1 | 0.001 | -70 | 306 | 221.433 | 47.3461 |
| 0.1 | 0.005 | -69.5 | 353.5 | 316.706 | 45.5276 |
| 0.1 | 0.01 | -53 | 343 | 308.229 | 47.8863 |
| 0.1 | 0.025 | -63.5 | 332.5 | 301.048 | 36.3972 |
| 0.2 | 0.005 | -70.5 | 306 | 221.433 | 47.3461 |
| 0.2 | 0.01 | -66 | 350 | 318 | 39.0117 |
| 0.2 | 0.025 | -61.5 | 329 | 302.712 | 31.2564 |
| 0.3 | 0.005 | -65.5 | 351.5 | 313.567 | 38.3845 |
| 0.3 | 0.01 | -73 | 352.5 | 314.683 | 36.2323 |
| 0.3 | 0.025 | -69 | 329 | 295 | 29.4722 |
| 0.4 | 0.001 | -65.6 | 357 | 315.416 | 43.6283 |
| 0.4 | 0.005 | -68.5 | 355 | 311.257 | 33.9134 |
| 0.4 | 0.01 | -87.5 | 345 | 305.733 | 33.5383 |
| 0.4 | 0.025 | -66.5 | 329 | 285.162 | 28.4711 |
| 0.5 | 0.005 | -75 | 351.5 | 304.286 | 39 |

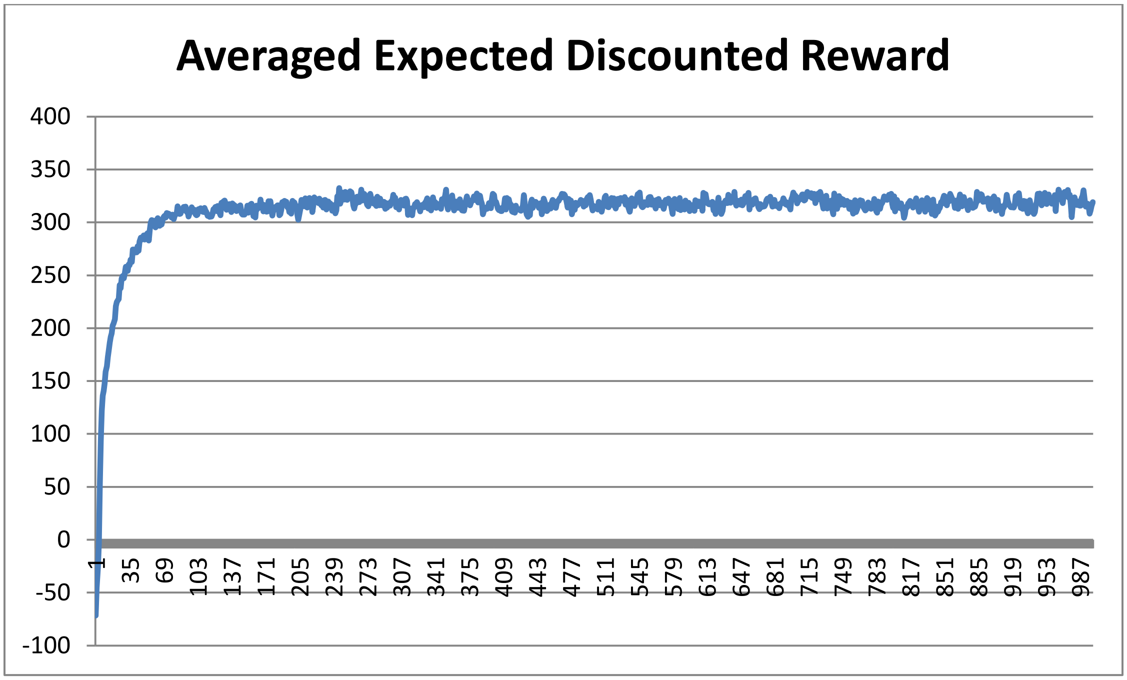
By inspection, we see that a learning rate of .4 and an $\epsilon$ of .005 generate the best results with nearly the highest maximum and average values and a one of the lowest standard deviations.

3) The expected discount reward for a single simulation:



The expected discounted reward increases rapidly and then it randomly oscillates around some value.

The expected discount reward averaged over 10 simulations:



Similarly to the previous plot, this plot increases very rapidly until it hits a limit around which it oscillates; however, the oscillations are much less noticeable since the values were averaged over 10 simulations. If we were to average infinitely many simulations, the plot would resemble a logarithmic has reached its limit.