Jonathan Pierce (jmpierc2) - CS 440 – MP1 (Part 2)

(Even though my solution for Part 1 got full points, I used the official solution for these experiments.)

Part 1: WorldWithoutThief

a)

With these parameters, we can see from PolicySimulator that the robot will move to the upper left corner of the map and stay there forever, never delivering any package. This results in a total reward of 0.

b)

According to PolicySimulator, the robot now behaves more reasonably. It will repeatedly deliver a package to customer 2, deliver a package to customer 1, and return to the store (taking 22 steps total). While there are packages, the robot minimizes time spent on slippery patches. When all packages are delivered, the robot takes a more efficient path back to the store that involves several slippery patches. If the robot drops the packages, it immediately returns to the store. Overall, the total reward increases as time goes on.

This improved behavior is the result of random exploration. When epsilon is set to zero, we behave too greedily, and get stuck repeatedly choosing the best option seen so far (which happened, as we can see above, to result in the robot getting stuck in the corner). With epsilon set to 0.1, we are going to explore alternative paths and the improved or lessened rewards they may bring.

c)

With epsilon set to 0.5, according to PolicySimulator our robot will still visit customer 2, visit customer 1, and then return to the store (still taking 22 steps). The exact 22 steps are slightly different, but the general behavior of avoiding slippery patches while holding packages remains the same.

Notably, while Simulator was running, most of the episodes resulted in negative reward (a result of the increased amount on random actions, many of which may not be optimal). While Simulator was running in Part B, most episodes resulted in positive reward. This clearly didn't affect the final policy much, but is worth noting.

This suggests to me that there are diminishing returns with increased epsilon values.

Part 2: WorldWithThief

a)

With epsilon at 0.05 and no knowledge of the thief, according to PolicySimulator the robot returns to its behavior of moving to the upper left corner and getting stuck. Here, it collect no reward, but this is better than the negative reward generated from approaching the corridor of the thief (which, since we don't have any information about how to avoid him, will likely steal our packages). Hence, with no knowledge of the thief, the robot is maximizing its reward in this case.

b)

Here, the situation is much improved. PolicySimulator shows the robot delivering to customer 2, delivering to customer 1, and returning to the store. All of this occurs while the robot both avoids

slippery patches (it only goes over them when it has no packages) and the thief (if the thief is in the way, the robot will move to avoid it).

Knowledge of the thief allows the robot to learn the optimal behavior for every position of the thief, every position of the robot, and the number of packages held.

c)

During this investigation, I ran the Simulator for 1,000 episodes of 10,000 steps each for several combos of learning rate and epsilon, where the learning rate ranged from 0.1 to 0.9 in 0.2 increments, and epsilon ranged from 0.05 to 0.45 in 0.1 increments. I chose to limit my exploration of epsilon to less than 0.5 because we saw above that it appears to have diminishing returns with higher values.

Then, I ran PolicySimulator for each for 1000 steps three times, and recorded the median total reward.

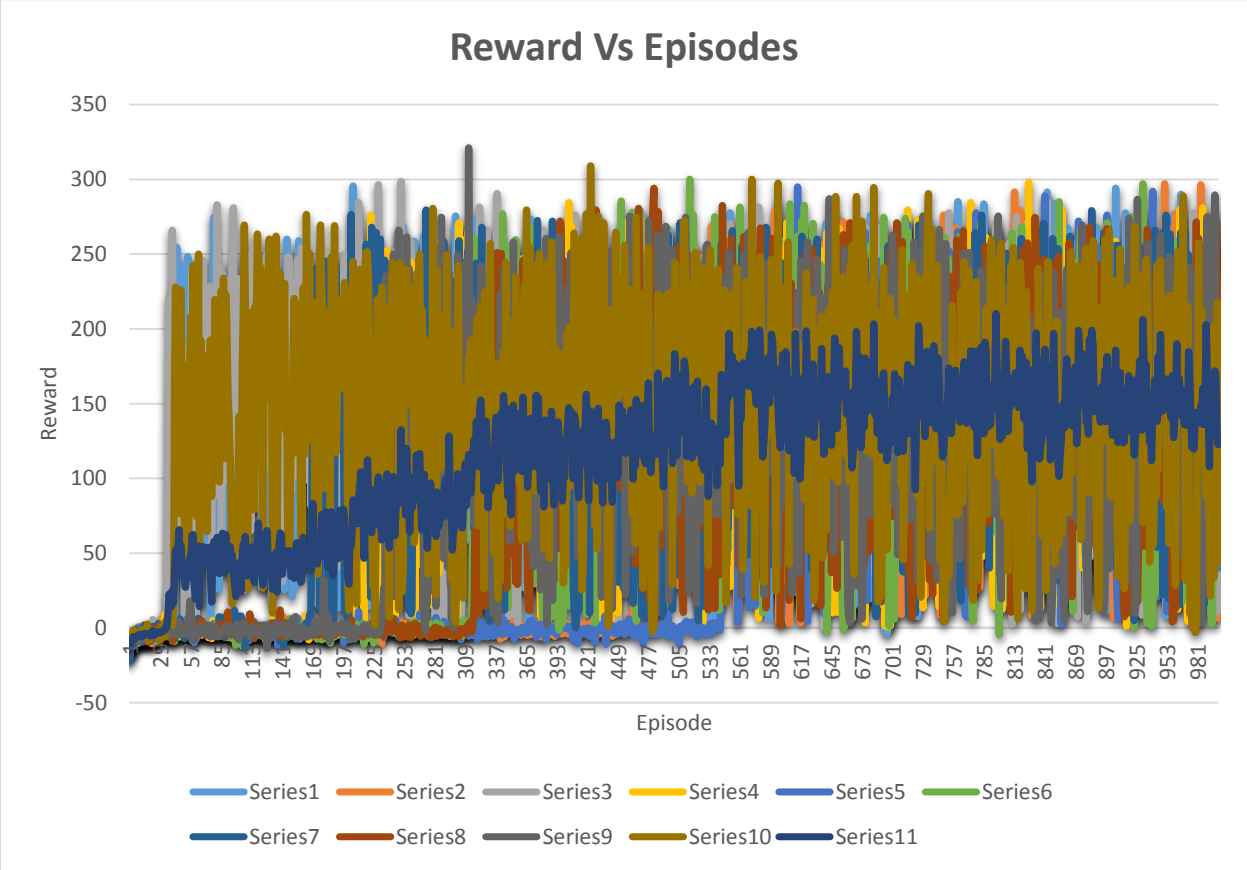
Epsilon \ LR	0.1	0.3	0.5	0.7	0.9
0.05	35	33	35	32	26
0.15	35	32	32	33	17
0.25	35	35	29	19.5	23.5
0.35	33	32	25	24.5	20.5
0.45	35	34.5	25.5	21	19

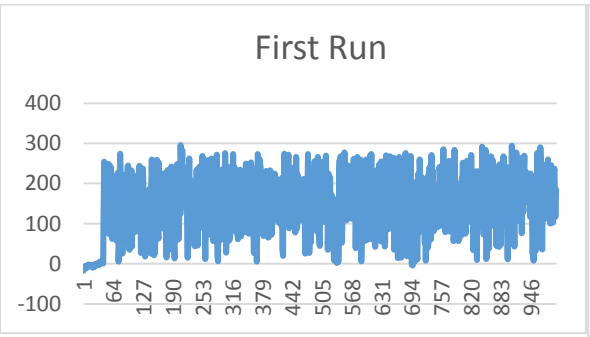
This data suggests that:

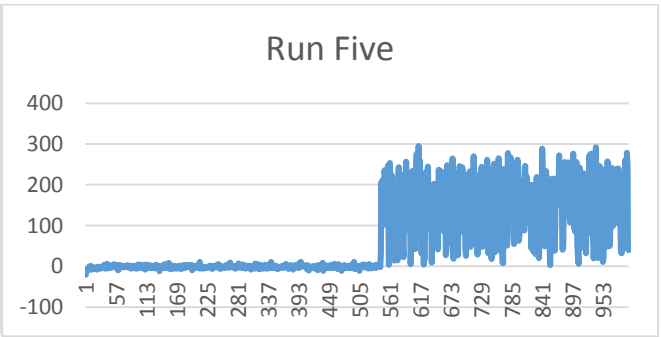
- Lower learning rates generally result in improved results.
- Lower epsilon values generally result in improved results.

So, I chose optimal values of epsilon to be 0.05 and the learning rate to be 0.1.

Part 3:







(Charts generated in Microsoft Excel. Here, series 11 is the average of the previous 10 runs and is the topmost dark blue line plot.)

As we can see from the average, the first few episodes tend to start of negatively. Then, after about 30 episodes, we begin to rapidly increase our reward. This reward growth then slows until the point where, after about 500 episodes, it begins to level off around a reward of 150. This suggest to me that the optimal policy is generally established at around 500 episodes.

All the runs generally showed the same growth at the beginning. Some of them, like runs 2,3, and 5 took longer than others to get off the ground (as we can see in the charts above, they hug the x-axis for a while before exhibiting explosive growth). One of them (run 5) even took over 500 episodes to get out

of single digit rewards! This is a likely result of being ‘unlucky’ and not having randomly explored a better path yet.

I suspect that the explosive growth seen in each run is the result of one episode being the first in which random actions result in the delivery of both packages and a return to the store. This being randomly successful once sets up the Q-values in such a way that this behavior is likely to repeat when we are taking non-random actions.

Once each run is ‘off the ground’, things can become very sporadic. As we can see, we could very well have an episode with reward 300 followed by one with reward 0! Even with an optimal policy, the world is still somewhat random: we can still drop packages, and the thief can still steal them. As our average shows, on the whole, reward generally improves to about 150 before leveling off.