CS440
MP1, Part 2
Michal Ruopp
ruopp2

NOTE: These tests were run with the provided solution.

1.) (a) For the default values (discount = 0.9, rate = 0.1, epsilon = 0.0), the agent moves to the upper left corner and becomes stuck, thus not receiving any rewards, positive nor negative. The episodes.txt displayed a net reward of 0.0 for all episodes.

(b) Now with epsilon changed to 0.1, the agent is slightly better. Every few runs it will successfully deliver some packages. It wasn't until my fifth time running the agent that it successfully delivered both packages and went back to the company for more (observed with the PolicySimulator). (The first two runs it immediately was stuck in an infinite loop -- one with two states and one with four states; and the fourth run didn't even leave the starting position). The episodes.txt file outputs mostly large numbers (usually between 100.0 and 250.0) with the occasional small (< 50.0) or negative number.

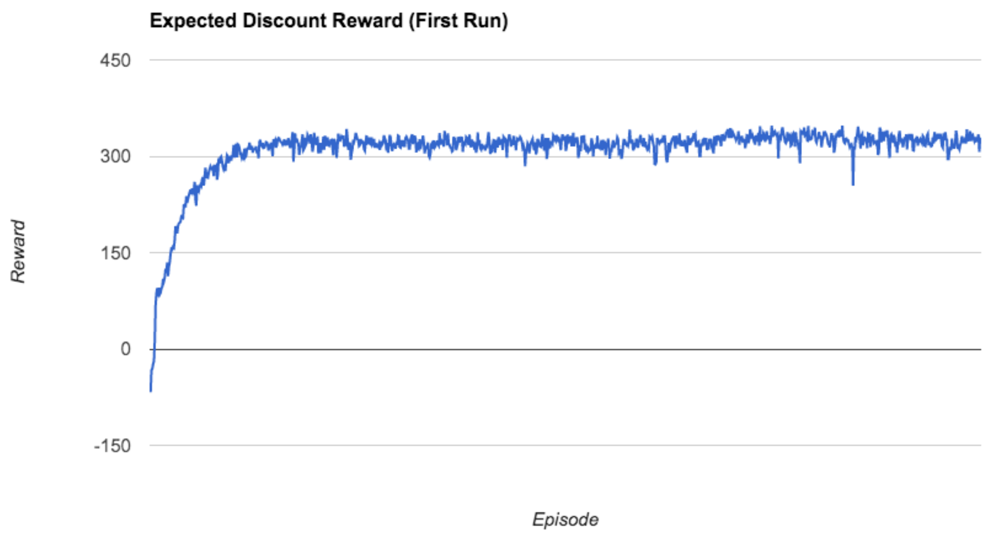
(c) Epsilon is now changed to 0.5 (with all discount and rate kept the same). When the agent succeeds it generally takes the same path, with a few variations from time to time. It still fails much of the time, either getting stuck in an infinite loop immediately or at some point in the delivery process, or just failing to leave the company. episodes.txt outputs mostly negative numbers between -10.0 and -70.0, with the rare positive number mostly between 0.0 and 10.0.

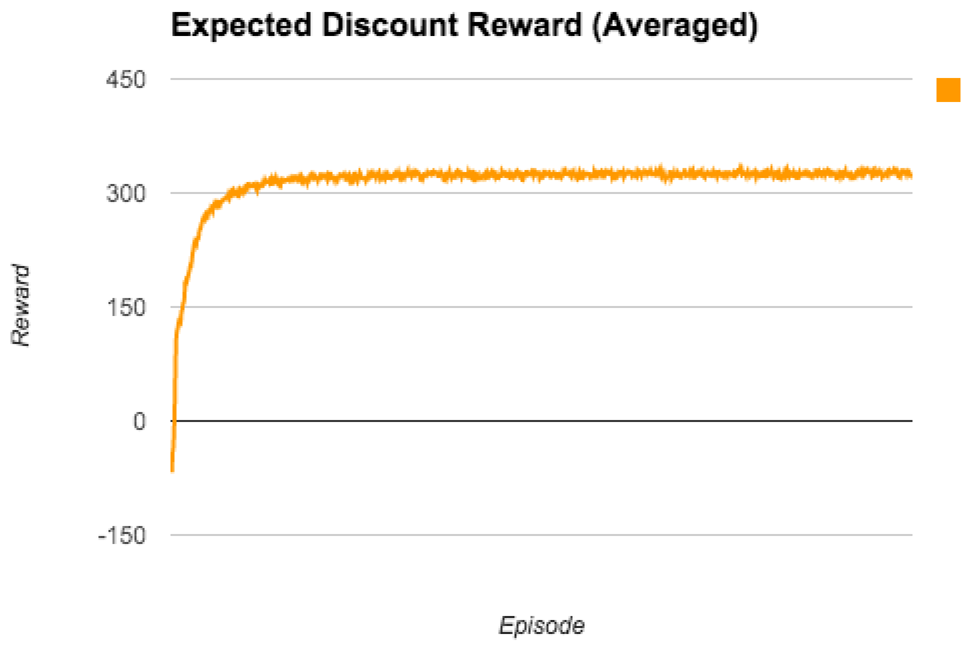
2.) (a) Now discount = 0.9, rate = 0.1, and epsilon = 0.5. The world now has a thief in it but the agent is not aware of its position. After multiple runs of this simulation, the robot fails to find a path in all runs. It mostly moves to the top left corner and then remains there. The agent probably assumes it is better to not deliver any packages and have a 0% chance of being caught by the thief, than to risk being caught. Values in episodes.txt are all negative, usually between -60.0 and -80.0.

(b) With the agent aware of where the thief is, it runs much better. The agent delivers packages multiple times all while avoiding the thief, thus accumulating a positive reward. The values in episodes.txt are all negative and range from about -150.0 to about -220.0. (This seems very odd to me since a working agent should have large positive values indicating it delivered many packages, but I re-downloaded the most up-to-date code and ran it again and received the same result).

(c) After thorough experimentation, approximately the highest reward values are when rate = 0.2 and epsilon = 0.01. I used an orthogonal search method to come to these conclusions. I first kept the learning rate at a consistent 0.1 while I searched for the most optimal epsilon. I started at 0.1 and incremented upwards by 0.1 until I reached 0.9. I discovered 0.1 was the best so I incremented downwards by 0.01. I discovered 0.01 was an optimal number. I used the same method for the learning rate, but now keeping epsilon at a consistent 0.01.

3.)





It reaches the maximum reward relatively quickly (~100 episodes in). It starts negative and converges pretty quickly. The average takes away some most of roughness.