Sayan Roychowdhury

CS 440

MP 1 Part II

1. WorldWithoutThief
   a. Conditions:
      i. *Discount factor = 0.9*
      ii. *Learning rate = 0.1*
      iii. *ε = 0.0*
      iv. *Episodes: 1000*
      v. *Steps: 10000*

   Observations: The reward always started out as some sort of negative number. However, within a few episodes (usually around 5), the reward hits 0.0 and then holds this reward until the last episode.

   This occurs because we set ε = 0.0 (no random actions except for ties). This means our agent is too greedy when executing actions. It will always choose the best action to take (random in the case of a tie). Once a Q value for a certain action is set to less than 0, the Q learning agent will never to take that path given these conditions, even though in the long run, that path might be the most optimal even though the first action in that path is suboptimal.

   b. Conditions:
      i. *Discount factor = 0.9*
      ii. *Learning rate = 0.1*
      iii. *ε = 0.1*
      iv. *Episodes: 1000*
      v. *Steps: 10000*

   Observations: This reward also always started out as some sort of negative number. For approximately the first 25 episodes, the reward hovered around some low negative numbers and single digit positive numbers. After that, the reward starts becoming much higher (around 100-200) and generally in that area for the rest of the episodes.

   This occurs because we set ε = 0.1. This allows for some random actions (meaning that the agent might take an action that is not the best action 10% of the time). This allows for variation in the actions that the agent takes and will explore new paths even if the first action in that new path might have a lower Q value than the best action.

c. Conditions:
   i. *Discount factor = 0.9*
   ii. *Learning rate = 0.1*
   iii. *ε = 0.5*
   iv. *Episodes: 1000*
   v. *Steps: 10000*

   Observations: Once again, the reward started out as a negative number. Just as the previous setup, the reward generally ends up between 100 and 200 after 1000 episodes. However, this time, it takes a lot longer for the agent to optimize its path. Sometimes it took about 100 episodes before the reward first hits a positive 3 digit number; other times it took up to 400.

   This occurs because we set ε = 0.5. This introduces a lot of randomness, because 50% of the time, the agent chooses a random action that might not be the best action. This accounts for the longer number of episodes that this agent takes to optimize its path in comparison to the agent where ε = 0.1. However, since the agent was run for a large number of episodes, it would still reach the high rewards that the agent in part *b* received.

2. WorldWithThief
   a. Conditions:
      i. *Discount factor = 0.9*
      ii. *Learning rate = 0.1*
      iii. *ε = 0.05*
      iv. *Knows thief: no*
      v. *Episodes: 1000*
      vi. *Steps: 10000*

      Observations: The agent does not perform well at all. It generally outputs a reward between -20 and -40 during the early episodes and still remains the same at the end of 1000 episodes, even after several simulations.

      If the thief's location is not known, it seems as if the reward is always negative, meaning that the robot's package is stolen quite often. Without the knowledge of the thief's location, the robot cannot make the decision to avoid him and goes in blind every time it crosses the blocks that the thief could be in

      .

b. Conditions:
    i. *Discount factor = 0.9*
    ii. *Learning rate = 0.1*
    iii. *ε = 0.05*
    iv. *Knows thief: yes*
    v. *Episodes: 1000*
    vi. *Steps: 10000*

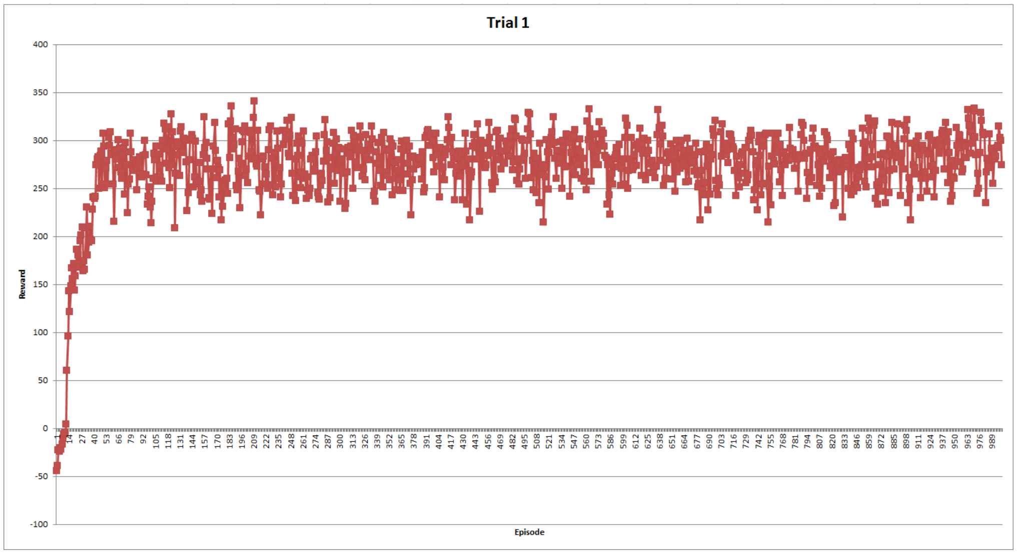
   Observations: The agent performs much better than the previous agent. Approximately the first 20 episodes are large negative numbers, but from then on, the reward increases very fast and is in the triple digits within a handful of episodes and generally stays around 200 until the end of 1000 episodes.

   If the thief's location is known, the robot learns how to avoid it very well, shown through the consistent high rewards. Knowledge of the thief's location drastically changes the agent's performance.

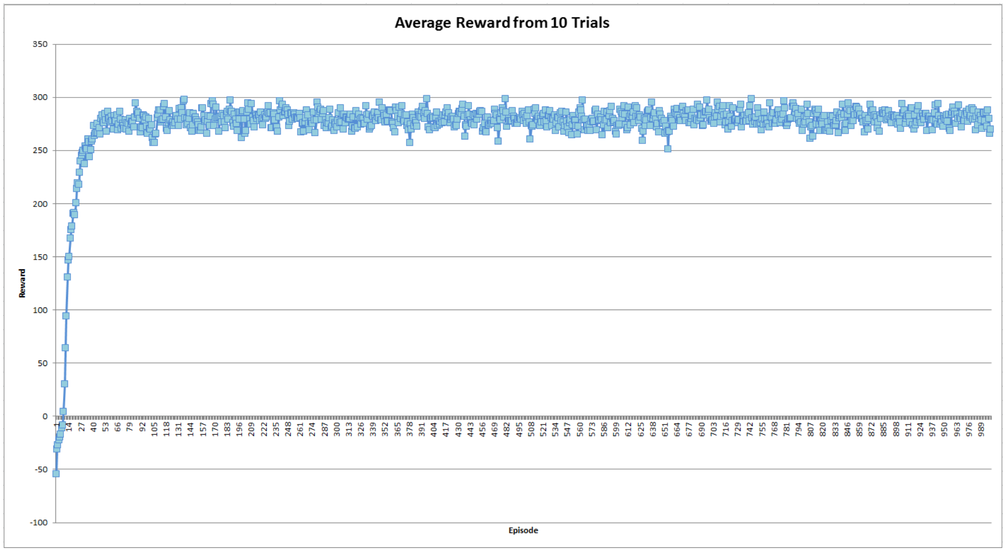
c. To find the best learning rate, I first held all of the other parameters at the values that were held in part b, including ε = 0.05. First I changed the learning rate to 1.0, and then decremented by 0.1 until I found that 0.7 gave me the end results as 0.6. After several simulations, I looked at the early episodes for both stages, and saw that for the one where the learning rate was 0.7, the agent started going from negative to large positive rewards earlier. Next I looked at a learning rate of 0.75 just to make sure, and I saw that the reward was not as high as the one with the learning rate of 0.7.

   To find the best epsilon, I held all of the other parameters at the values that were in part b and the recently found best learning rate of 0.7. The first thing I did was to set epsilon to 1.0, and then kept decrementing by 0.1. It started off with very negative rewards and kept slowly increasing until it finally hit positive single digit and low double digit numbers around ε = 0.2. At ε = 0.1, the final rewards hovered somewhere in the 100s. Once I hit ε = 0.1, the rewards were still increasing, so I started decrementing the epsilon value by 0.01. At ε = 0.01, the rewards were consistently hovering between 250 and 300. Here I decided to keep decrementing the value by 0.01, but now all of the end rewards were all very similar. For all of them, the values went from negative to positive within the same amount of episodes. Therefore I decided to use ε = 0.01 as my best epsilon value.

3.



The very first thing I notice about this graph is how steeply the reward increases within the first 50 episodes. After that, the reward per episode hovers between approximately 250 and 300.



This graph is very similar to the one from the single trial. There is a very steep incline during the first 50 episodes, just like the previous graph. However, the range of rewards is smaller than the other graph. That is, the average rewards between about episode 50 to episode 1000 on this graph are all closer to each other than the rewards in trial 1. The average is a better representation of the data than a single trial.