CS 440 / ECE 448: Introduction to AI
MP1 – Part2

Amirhossein Aleyasen (aleyase2)

1. WorldWithoutThief

(a) In this scenario, the robot stuck in first row and didn't move to 2nd row. In episode.txt all values are 0.0 (except first line that is -8) so the robot couldn't deliver any packages using this configuration. Since in this case, we haven't any random actions, QLearning stuck in local minimums.

(b) In this case, usually the robot has good behavior and delivers both packages. In some cases, it stuck in some situations (monitored by PolicySimulator), but totally the behavior was acceptable. In addition, the ranges of rewards in episode.txt file were between 100 to 200 in most of the cases. It seems epsilon=0.1 is acceptable value when the QLearning stuck in local minimums and the random actions improve the performance of the learning method.

(c) In this case, the behavior of robot was not acceptable and usually it stuck in the maze. The rewards in episode.txt file were negative in most of the cases. It seems with epsilon=0.5 the randomness of QLearning is very high and dominate the greedy behavior of algorithms and cause the robots don't reach to the destinations in most of the cases.

2. WorldWithThief

(a) In this case, the robot usually stuck in first row and couldn't deliver any packages. Most of the rewards in episode.txt are negative. Since the robot don't know the thief location, the uncertainty (randomness) of the environment is high and QLearning cannot train very well, so the behavior of robot in this environment is not acceptable.

(b) In this case, the robot works well. The rewards in episode.txt are usually greater than 200. Since the robot knows the thief location, QLearning consider it in learning process, in the result, the behavior of robot is good.

(c) By using some brute-force method, the best learning rate is on range [0.005,0.015]. For learning rate, the learning process in not so sensitive to it, for example when epsilon=0.01 by modifying learning rate from 0.1 to 0.4 the results are almost same and it hadn't significant effect on the robot behavior. However for learning rate less than 0.05 or greater than 0.5 the performance changes are significant. It seems than calibration of epsilon is more important than learning rate and for optimizing the method, it's better to optimize epsilon and then modify learning rate for fine calibration.

3. For this experiment, we used epsilon=0.01 and learning rate=0.15. The results of 10 times runs given in Figure 1.

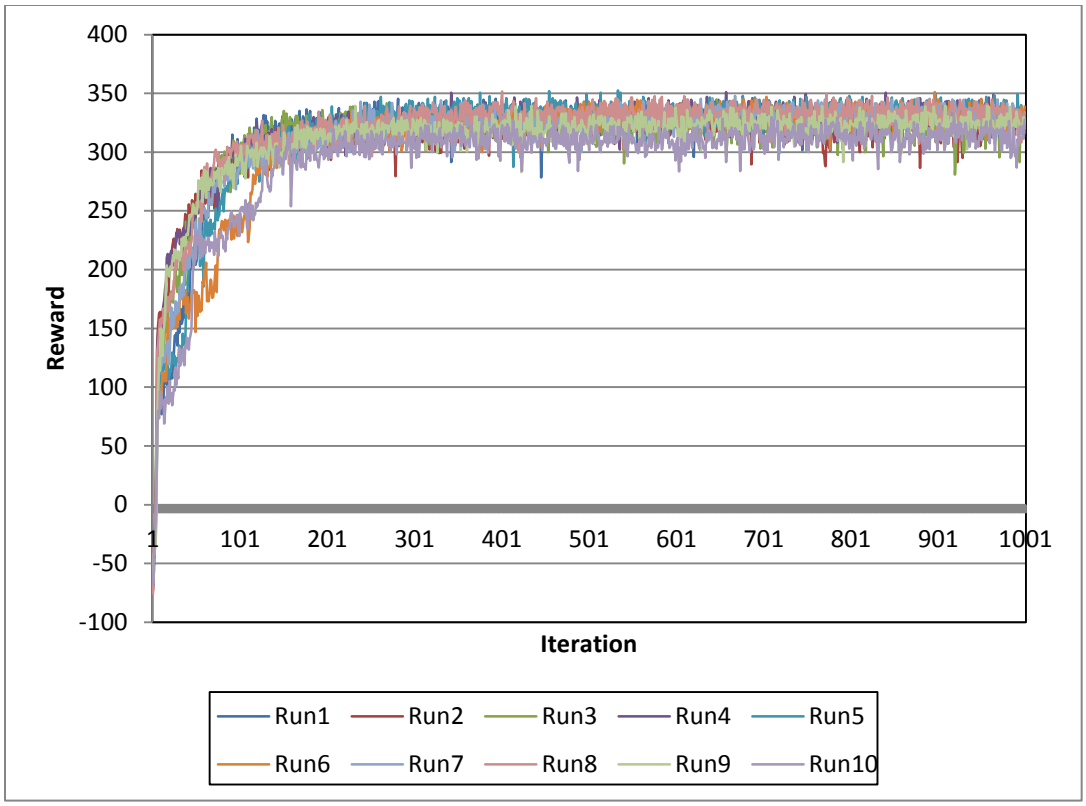
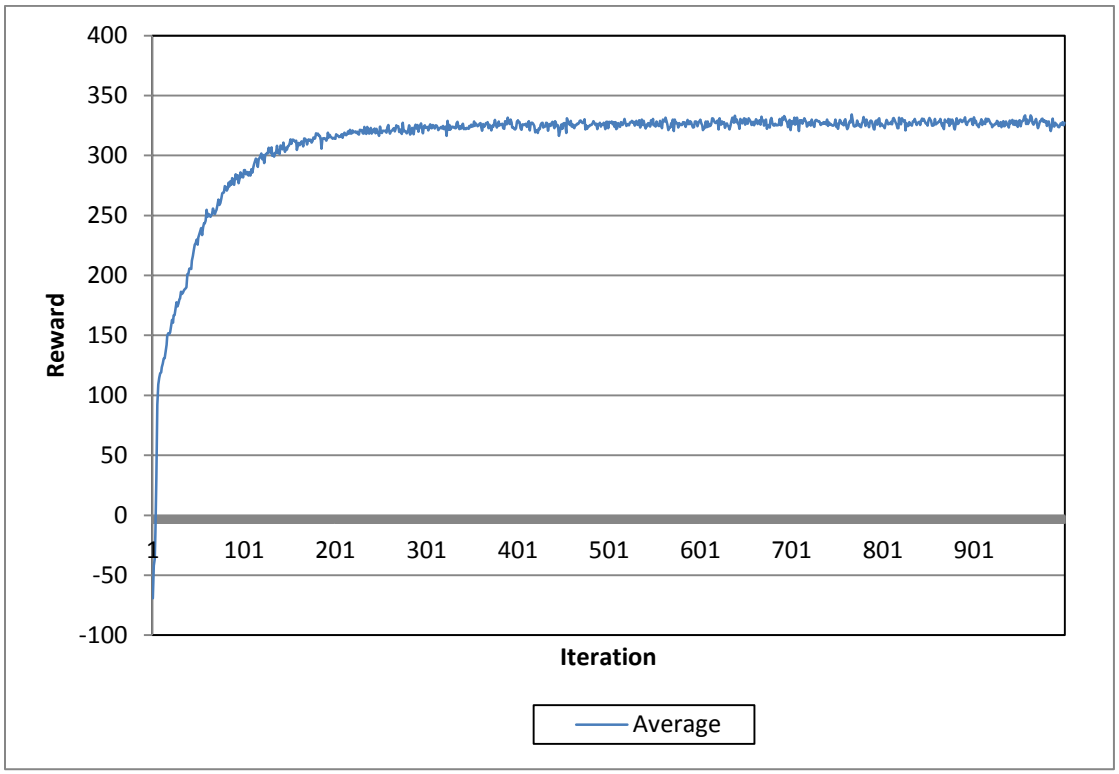


Figure 1. expected discounted reward for 10 times runs

And the average of 10 times runs given in Figure 2.



The rewards has minor changes during learning process but as a big picture view, the reward reach to its maximum on iteration=~300 and stay almost there for remaining of the simulation. However in Figure 1, there are some cases the behavior of reward is less steady. One possible reason of chaotic behavior could be the random behavior of thief; however QLearning could handle this behavior in most of the cases.