CS 440/ECE 448 MP1: Part II

Sean Yen

1. WorldWithoutThief

(a)

Following are the rewards for the first 10 episodes.

```
−7.0
−0.5
−0.5
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

The reward remains at 0 for the rest of the simulation. This seems to be because the agent is "afraid" of trying new actions for fear of negative rewards. Here, we see that the agent starts by receiving a negative rewards, reduces that negative reward by a bit, and when it hits 0.0, it seems to settle for this being the best that it can do.

(b)

Following are the rewards for the first 30 episodes.

```
−21.5
−10.0
−11.5
−7.0
−13.0
−8.0
−6.5
−4.5
−5.5
−9.0
−9.0
−3.0
−5.0
0.0
−5.5
−5.0
−0.5
2.0
−7.5
−5.0
2.0
−4.0
3.5
−3.0
−4.5
−3.5
−6.0
2.5
1.0
17.0
```

We clearly see that the reward passes 0.0 and becomes positive in this case. In fact, in episode 31, we see that the reward jumps to 147.5! The reason that we are able to achieve such good rewards is because with $\epsilon = 0.1$, our agent is randomly selecting actions that may not be in the policy. This is allowing it to explore the "world" more fully to find more optimal solutions to the problems.

(c)

Following are the rewards for the last 10 episodes.

```
    -34.5
    -28.5
    -15.0
    -29.5
    -9.5
    -33.5
    -18.5
    -41.0
    -49.0
    -16.0
```

In this case, notice that even after 1000 episodes, the reward is still negative (and *very negative*). It seems that ϵ is set too high and the agent is taking random actions too often. This is causing it to choose random actions more often than it should and it is "stumbling" into a non-optimal path.

2. WorldWithThief

(a)

Following are the rewards for the first 10 episodes.

```
    -26.5
    -20.0
    -14.5
    -20.0
    -15.0
    -11.5
    -14.5
    -9.5
    -11.0
    -18.0
```

The rest of the episode rewards continue much like this. The agent rarely achieves positive rewards. It seems that without knowing where the thief is, since the thief moves randomly, when the agent encounters a thief, it can't figure out why it's incurring a negative reward and so it doesn't know how to avoid the thief effectively.

(b)

Following are the rewards for the first 10 episodes.

```
    -79.5
    -50.5
    -43.5
    -35.5
    -15.0
    85.5
    148.5
    149.0
    135.5
    133.0
```

Over the course of the 1000 episodes, the agent goes on to achieve awards of around 300! In this case, since the agent knows where the thief is and has states to account for the position of the thief, it can find a deterministic way of avoiding the thief and delivering the goods to incur the greatest reward.

(c)

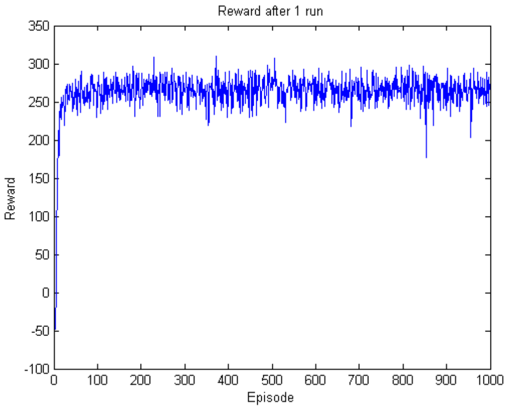
See the following table of a survey of possible $\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ and $\epsilon \in \{0.0, 0.05, 0.1, 0.5, 0.7\}$ values and the resulting rewards at episode 1000.

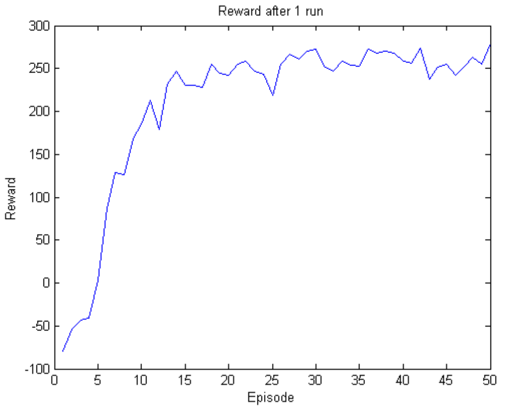
				ϵ		
		0.0	0.05	0.1	0.5	0.7
α	0.1	77.0	267.5	200.5	-192.5	-213.5
	0.3	111.0	281.5	219.5	-222.0	-247.0
	0.5	109.0	286.0	186.5	-212.5	-270.5
	0.7	107.0	222.5	176.0	-187.5	-269.5
	0.9	76.0	178.5	142.5	-200.5	-250.0

The values that I chose cover the main support of α and ϵ and show what I think is a representative survey of the effect of these parameters on the reward at episode 1000.
Looking at these results, it's clear that choose ϵ as 0.5 or 0.7 clearly is not optimal. We get very large negative rewards in these cases. It seems like choosing a small, non-zero ϵ (0.05 seems to be the best) produces the best results. For α, we see a more hyperbolic relationship. Small and large α's given worse results. This is likely because in these cases, we are trusting the new information too much (or not enough). The best α's appear to be 0.3 and 0.5.

Problem 3

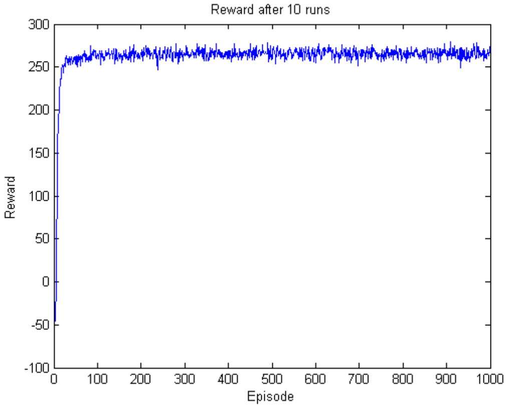
The following plots were generated using $\alpha = 0.3$ and $\epsilon = 0.05$. These plots are episode by reward which shows the change in the reward as our agent refines its policy. The plot on the left will show rewards as they change over the course of 1000 episodes. The plot on the right shows rewards over the first 50 episodes where most of the growth will occur.

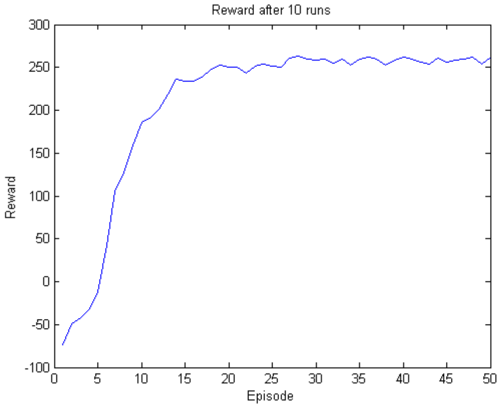




Notice that the reward seems to become level around 250 ($\mu_{\text{episode}\in[20:1000]} \approx 265$). In fact, within the first 20 episodes, the reward reaches this asymptote. Afterwards, the values hover around this asymptotic value ($\sigma_{\text{episode}\in[20:1000]} \approx 15$).

Next we consider the same plots but averaged over 10 runs of 1000 episodes.





We now observe the same general behavior, but the function seems to be much smoother. This can also be seen in the parameters $\mu_{\text{episode}\in[20:1000]}$ and $\sigma_{\text{episode}\in[20:1000]}$. In this case, $\mu_{\text{episode}\in[20:1000]} \approx 265$ and $\sigma_{\text{episode}\in[20:1000]} \approx 5$. It's notable that the standard deviation, σ, is much lower than it was in the previous case when we just considered 1 run of 1000 episodes. This is due to the fact that averaging here reduces the noise in the estimate of our behavior.