CS440 MP1 Part 2 Experimentation report

Ruisheng Shi

September 27, 2014

1 In WorldWithoutThief

1.1 (a)

The result is that except of those first one episode -10 value, rest of episode is always 0.0. And the simulation of the robot just stayed at the same place without stepped onto any slippery grid, even not moving at all. The reason is that for each grid position, for each package, there are a state. Therefore the robot will have 5*5*2*2 = 100 states. And for each of state, there is only one determined policy at a time. That means the robot will always execute the same action in the same state without updating policy. And the robot will get a negative(positive) feedback only if it had a new reward after took an action. For example, the robot can step freely from left to the middle grid as long as he is lucky(he didn't slip). But only one slip, could result in the negative value for the state of left grid and action to move right. Because the ϵ is 0, the robot will cautiously avoid any negative value. So the robot will never forward right from the left grid. Since the there is only one postive reward, and it requires at least 14 steps and step onto at least three small slippery grid to get a positive reward and it is even harder to reproduce the positive reward(Without reproduce of the positive reward, it is very hard/ impossible to propagate the positive value to the robot). Therefore it is very common that the robot will stay in one place because for so many steps of simulation, any action to move up or right looks very bad. Move to the wall at least get a 0 reward.

1.2 (b)

Despite of first a hundreds or so negative total reward, the total reward went up abruptly. The reason is that because we walk randomly with $\epsilon = 0.1$. The robot are forced to try some actions which the imperfect information would tell him not to. Therefore, this increase the chances for the robot to break the barrier of the slippery grids and reach the sweet pot(delivered both packages). But since the robot need to reach the grid adjacent to last sweet pot in order to connect the path. It requires many of episodes to do so. But once the sweet pot is back propagated from the delivered location to the start location. The robot will learn the sweet path and repeat this path as long as it is lucky enough(it is possible that robot dropped several packages at the same spot and choose an another path

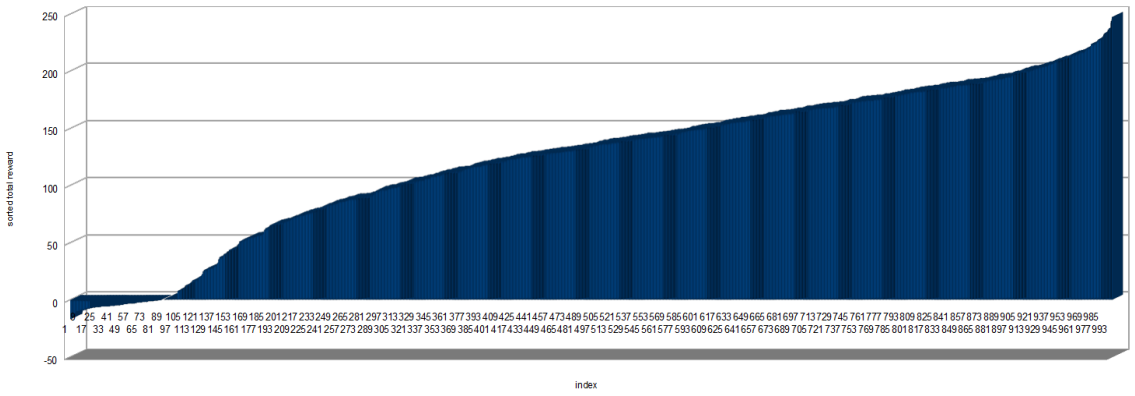

Figure 1: sorted total award of 1000 episodes of 1(b)

at that spot. This did happened in the simulation. But it may come back later because the incentive are greater than negative impact). So I run the policy simulations and robot run either an optimal path of 22 steps or a suboptimal path with 24 steps.

I calculated the expectation for a single move of the optimal path. since the shortest path to get reward 1 is 22 steps. with possibility of $0.9^3 = 0.729$, similarly, it has a chance of 0.1 to drop the package and return to company in 4 steps and a chance of 0.1*0.9 = 0.09 to drop package and return company in 8 steps and $0.9^2 * 0.1 = 0.081$ possibility to drop package when third time passing low slippery grid and return base in 16 steps. Therefore, if the experiment can repeat forever, the path along the optimal will get a positive reward value around 0.01248, so repeat the steps by 10000 times, the expectation is 124.8, which looks convincingly from the figure 1(Here, I first sorted all episode's total reward and then displayed them).

1.3 (c)

The result of 1(c) is very interesting(figure 2). The overall average performance is terrible, -29 in average. The reason is that this world is full of penalties and only one reward. Robot running too sloppy/random not only get lots of negative rewards but also even if robot 'know' the right path, it can't execute the path very well and may step into slippery area even if it knows it is a terrible choice. So setting a very high ϵ is a terrible thing. On the other hand, the output policy still has significant chances to find a positive path from the output policy.txt.

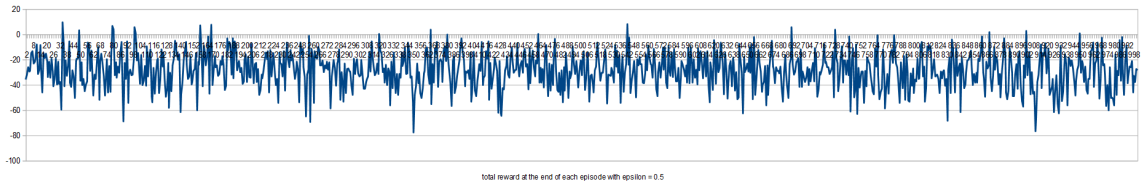

Figure 2: total award at the end of 1000 episodes of 1(c)

2 in World With Thief

2.1 (a)

If the robot is not aware of the thief. Even if it has a $\epsilon = 0.05$ chance to random walk. The average outcome is still -13. The reason is that the thief almost block the only entrance of the package 2. Since robot need to deliver both packages to get a positive reward, without knowing the thief, it is very likely to met the thief and be hurt very bad(bigger punishment than dropping the package). Therefore, the simulation of the robot almost always sit in the company. The total reward for each episode is negative because robot will random go out and hurt himself(figure 3).

2.2 (b)

If the robot recognize thief(now the number of states are 500), it will take the position of the thief into account. Especially when it is near the thief. Since the thief's location could be well aware and the optimal solution wouldn't need to go through the slippery grid. So the only hindering factor for delivering package is the random position of the thief.At the end, we have a very stable excellent performance from the robot. The average total reward is 272 and the distribution of the total reward hold a very high bar and converge very fast(figure 4).

2.3 (c)

Refer to figure 5, I have repeat this experiment for 20 configurations and each running 4 times. Since the $\alpha = 0.01$ looks Superior than other α value, I will compare the five ϵ value

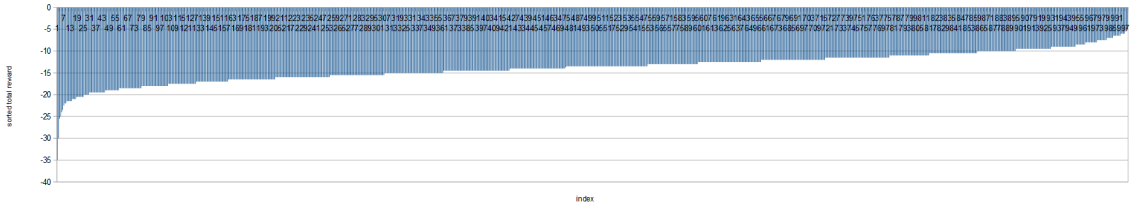

Figure 3: sorted total award of 1000 episodes of 2(a)

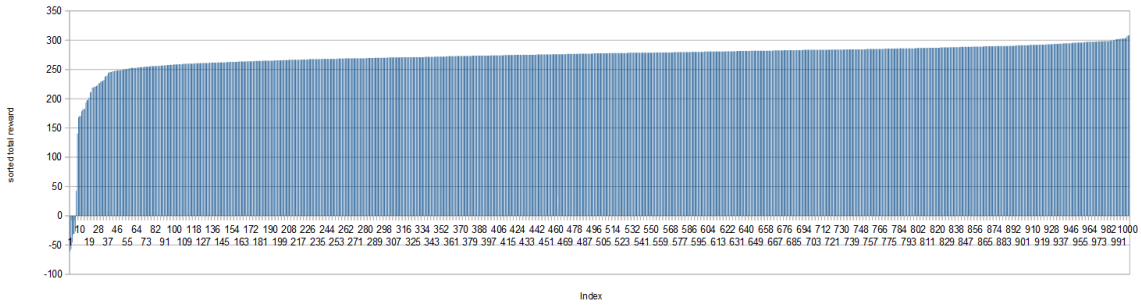

Figure 4: sorted total award of 1000 episodes of 2(b)

with $\alpha = 0.01$ and four α with $\epsilon = 0.06$

ε \ α	0.05	0.1	0.15	0.2
0.02	300.985	308.639	307.154	304.409
0.04	283.305	286.914	286.314	282.124
0.06	255.748	263.88	260.74	258.509
0.08	235.35	238.807	236.101	233.458
0.1	210.376	211.659	212.134	207.54

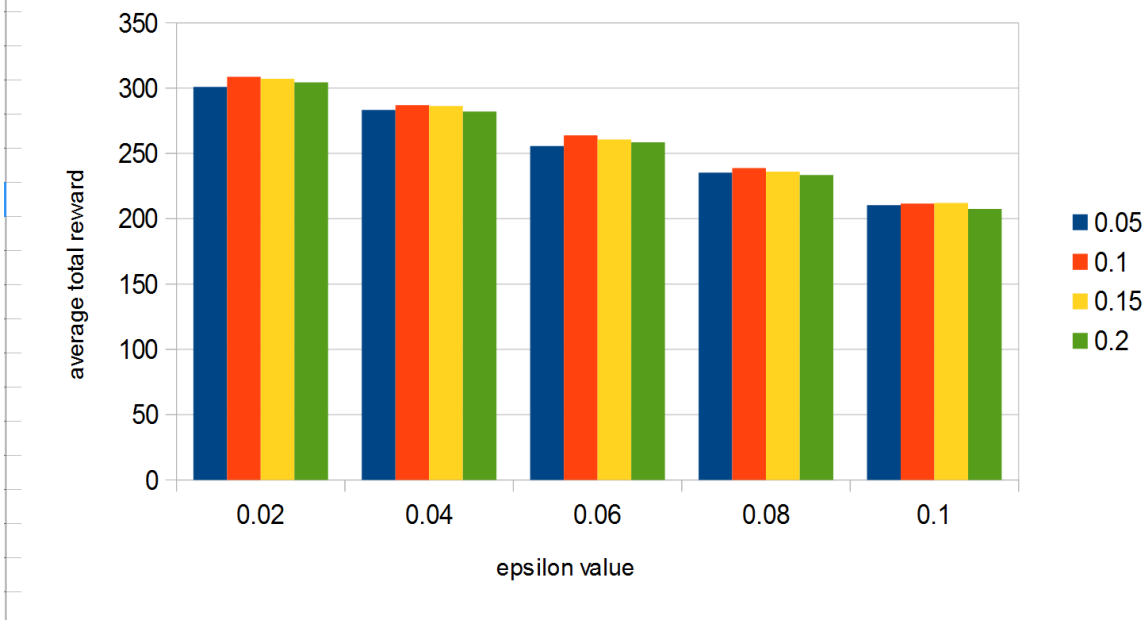

Figure 5: average total awards from different $\epsilon = 0.02 - 0.10$ and $\alpha = 0.05 - 0.2$ of 2(c)

2.3.1 $\alpha = 0.01$ and different ϵ value

From figure 6, it is seen very clearly with the increasing of the ϵ value, the 'ceiling' of the total reward drops significantly. Also,refer to figure 7, from $\epsilon = 0.02$ to $\epsilon = 0.06$, there is an improvement for the converge from episode 10 - 60(First 10 episode contains some randomness). Furthermore, there is no improvement from $\epsilon = 0.06$ to $\epsilon = 0.10$ in terms of both convergence rate and total rewards. Because the world for 500 states is not changing, after find the optimal strategy, the larger random move ϵ is, the lower total reward will be.

2.3.2 $\epsilon = 0.06$ and different α value

Figure 8 comparing to different value of α, we can see that it has almost nothing to do with the average total rewards. The only thing it affects is the converge rate. This is because the learning rate reflects the relative confidence in the old and new information, typically does not preclude convergence but can have a significant effect on the speed of convergence. Despite the low performance of $\alpha = 0.05$, rest of the α value have not much difference.

So, after the comparison between different α and ϵ. The best value I found is $\alpha = 0.1$, and $\epsilon = 0.02$ for long episodes(≥ 100) and $\alpha = 0.1$, and $\epsilon = 0.06$ for short episodes($10 <$ Number of episodes <100). But this value is totally world dependent.

3 Plot

Since the number of episode is large, I will use $\alpha = 0.1$, and $\epsilon = 0.02$.
Refer to the graph 9, it is clear that after 100 episodes, all ten attempts run very smoothly and achieve a very high score(305 in average). But since we have a relatively small ϵ, the first 100 episodes performance is total random and diverged wildly. however the average value of first ten attempts are relatively smoothly. This average curve can better represent the property of current configuration. It is like tossing a coin, despite the first limited times of tries. After we tried again and again and again, the estimate of probability of seeing a head will approach 0.5.

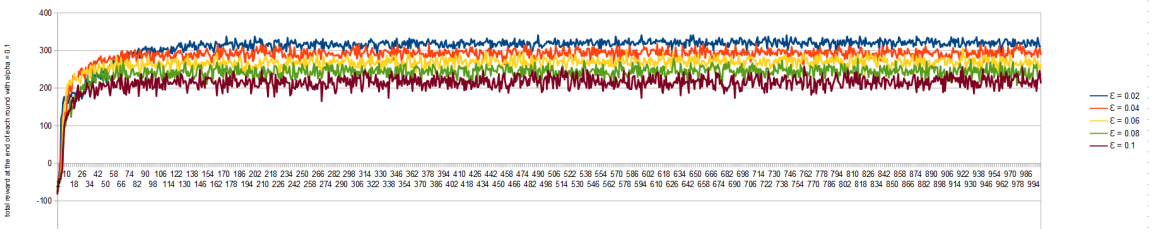

Figure 6: total awards at the end of each episodes from different $\epsilon = 0.02 - 0.10$ and $\alpha = 0.1$ of 2(c)

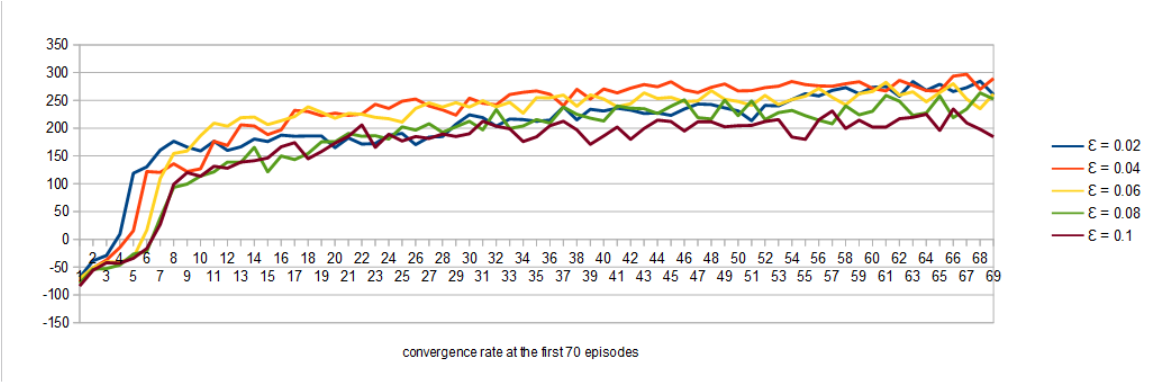

Figure 7: total awards at the end of each episodes from different $\epsilon = 0.02 - 0.10$ and $\alpha = 0.1$ for first 70 episodes

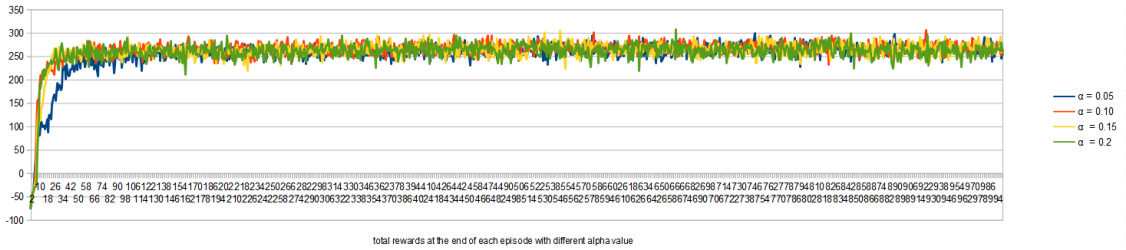

Figure 8: total awards at the end of each episodes from $\epsilon = 0.06$ and different α of 2(c)

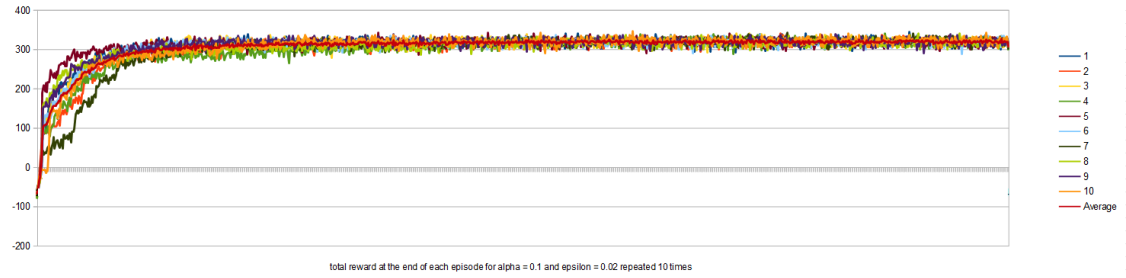

Figure 9: total awards from $\epsilon = 0.02$ and $\alpha = 0.1$, run 10 times