Machine Problem 1 - Reinforcement Learning

Tsung-Han,Lei

UIN : 658387539

1. In WorldWithoutThief

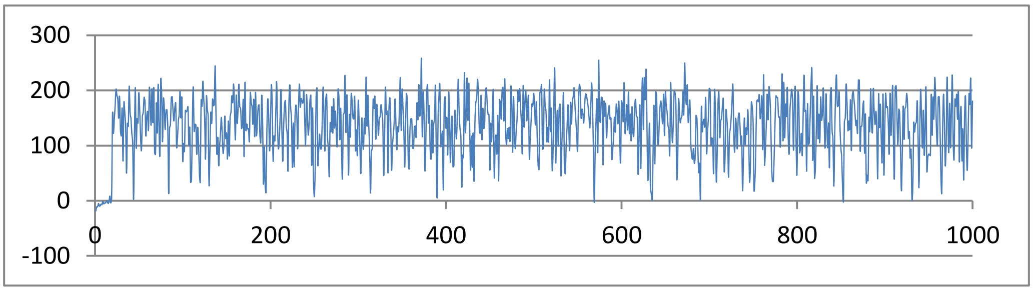
(a) Discount factor=0.9,learning rate=0.1, ε =0.0

Average reward: 0.0

The robot walk upwards, then stocks at the left-top corner at all. Then it does not go anywhere.

In the first episode, the robot tries to go to the right side of the map randomly (at this time, almost every Q-value one the Q-table is 0, so it walks randomly). When it tries to go right, it touches the areas which have some slipperiness, and it has chances to drop the packages. Even if it successfully walk through the dangerous area, it is also hard for the robot to safely deliver the packages to both families. Finally, after several episodes, it learns that walk upward is the only way that prevents it from getting minus reward. That's why the reward of first episode is -8, and others are all 0.

(b) Discount factor=0.9,learning rate=0.1, ε =0.1

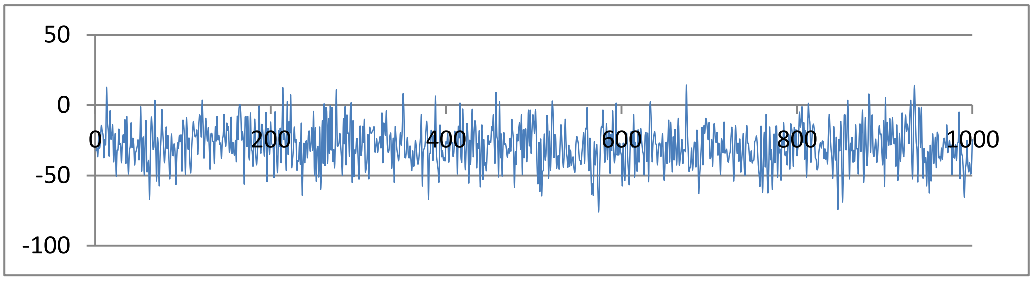


Average reward: 136.25

The robot goes to 1 first, and then goes to 2. On its way goes from 1 to 2, it does not go through the areas that have more chances to drop the packages. On its way back to the company, it walks through the areas that have higher slipperiness.

Compare to problem 1(a), instead of constantly select the actions following the greedy algorithm after the Q-table is stable, this problem has more flexibility to select different paths and also has more chances to update the Q-table.

(c) Discount factor=0.9,learning rate=0.1, ε =0.5

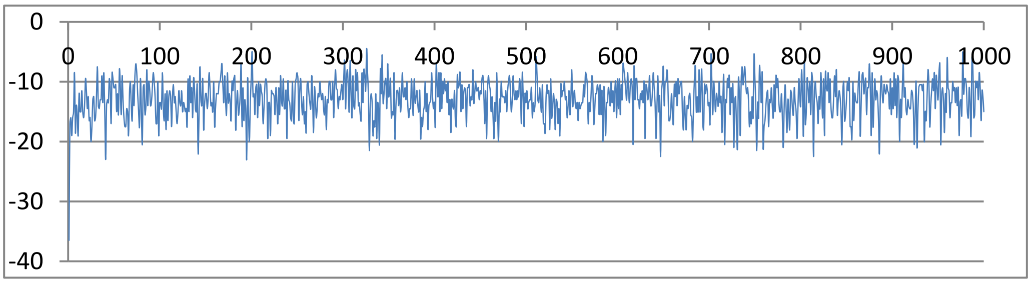


Average reward: -28.677

Compare to problem 1(b), the flexibility of the robot is too high that it has only half of chances in each step to follow the greedy path. The reward decreases when epsilon grows up, since the action of the robot may be more similar to walking randomly.

2. In WorldWithThief,

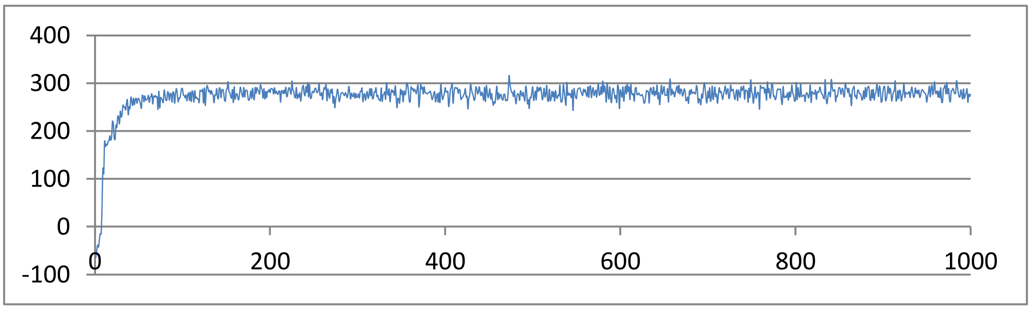
(a) Discount factor=0.9,learning rate=0.1, ε =0.05,Knows_thief=n



Average reward:-12.926

The robot does not know which column the thief is in. On the other hand, it does not have states that represent if there's a thief or not. So while it is walking on the third column, it may has chances to meet the thief.

(b) Discount factor=0.9,learning rate=0.1, ε =0.05,Knows_thief=y



Average reward:273.29

Compare to 2(a), this time the robot has states that represent if there's a thief. It can made decisions that prevent it from encountering with the thief. Also, the robot can reduce the steps walking on the third column in order not to bumping into the thief.

(c) I tried to change learning rate from 0.01 to 0.7, ε =0.05, the result is below:

Learning rate	0.01	0.03	0.05	0.07	0.09
Reward	262.73	269.4	268.84	268.98	268.84
Learning rate	0.1	0.2	0.3	0.5	0.7
Reward	272.08	270.35	261.57	237.93	208.45

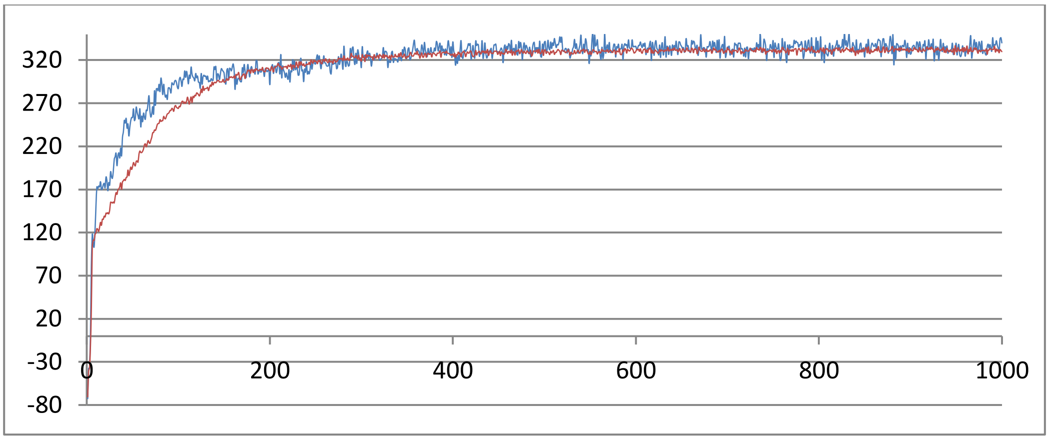
As a result, I select the learning rate as 0.1.

Then I change ε from 0.01 to 0.5, learning rate=0.1, the result is below:

ε	0.01	0.02	0.03	0.05
Reward	317.2895	309.3305	294.554	273.264
ε	0.1	0.2	0.3	0.5
Reward	212.759	92.224	-16.247	-180.67

As a result, I select ε as 0.01.

Discount factor=0.9,learning rate=0.1, ε =0.01,Knows_thief=y



The blue line is the first experiment, and the red line is the average of 10 experiments. It seems that the red line is much more stable than the blue line. I think the reason is that because the blue line has some probability involved (ε), it may has different reward values through all episodes. However, the red line is the averaging result of 10 experiments, and the instability caused by ε is reduced. By observing the red line, we can see the tendency that reward increases when the number of episode becomes larger.