Solution of Chapter 6

6.1



6.2

Regarding for I have moved to a new building and a new parking lot, but I still enter the highway at the same place, so the difference of time cost is the way between the living building and the highway entrance. At beginning, I suppose the time costs are the same for all the legs, then after I drive to the highway enter, I will know the difference and update the time cost as TD suggested. Thus, I have already known the new estimated value. While the MC method need me to calculate the new value only when I arrive the office.

6.3

The first episode was terminated at the left.

The reason why only V(A) changes is that the rewards among the states are 0, so the update is 0, except for A whose next state is terminal state with value 0.

It changed 0.05.

6.4

Do you think the conclusions about which algorithm is better would be affected if a wider range of ↵ values were used?

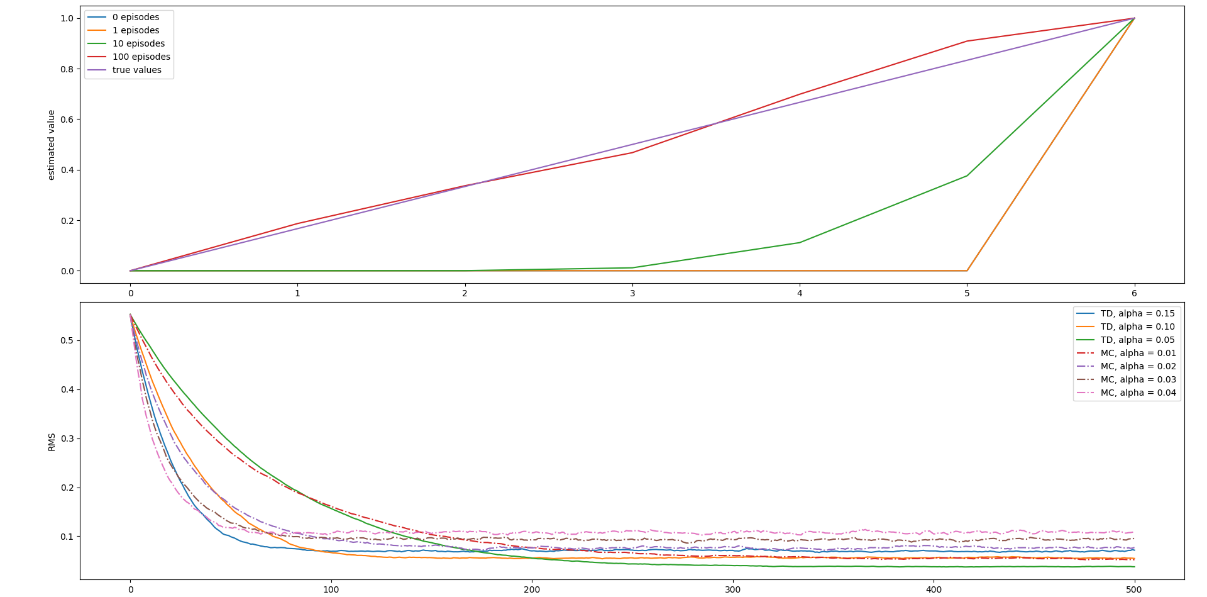
I think a wider range of step value doesn’t say more than current step value, because step value affects the rate of converge, which is shown clean in the diagram, and the accuracy of the converge value, and a wider range only can reproduce the tendency again.

Is there a different, fixed value of step at which either algorithm would have performed significantly better than shown? Why or why not?

I don’t think there is a fixed value can perform better, because there is a tradeoff between the converge speed and the accuracy of the converge value.

6.5

It seems this is because of the initial value. When I change the initial value, the curve always goes down.



Initial state value is 0.

6.6

Because the state transition probability is known, so we can use

1. Bellman equation to calculate the Value of state directly.
2. DP method to evaluate the value.

It is better to choose Policy evaluation method.

6.7



6.8



6.9

Optimal policy is:

['R', 'D', 'D', 'UL', 'L', 'DR', 'DR', 'DR', 'L', 'D']

['UL', 'UL', 'D', 'UR', 'DL', 'DL', 'R', 'L', 'DL', 'DR']

['D', 'DR', 'D', 'R', 'DR', 'DL', 'DR', 'D', 'UR', 'DR']

['DR', 'D', 'DR', 'L', 'DR', 'UR', 'DR', 'G', 'DL', 'L']

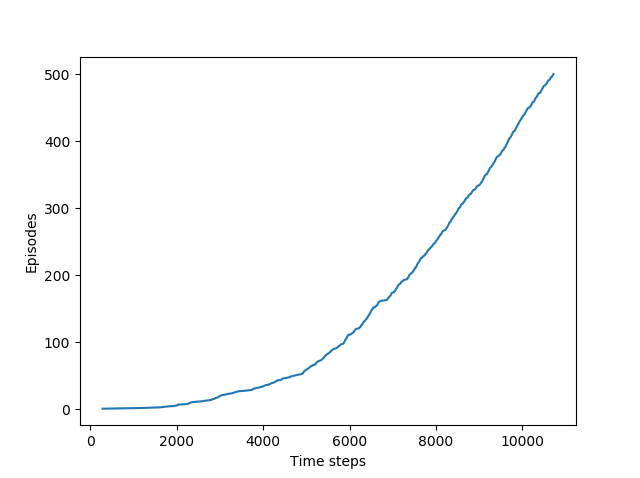
['DR', 'DR', 'DR', 'DR', 'DR', 'DR', 'DR', 'D', 'L', 'DL']

['R', 'DR', 'R', 'DR', 'DR', 'DR', 'R', 'DR', 'L', 'L']

['DL', 'R', 'DR', 'UR', 'DR', 'UR', 'UR', 'U', 'DL', 'L']

Wind strength for each column:

['0', '0', '0', '1', '1', '1', '2', '2', '1', '0']



6.10

Optimal policy is:

['R', 'R', 'R', 'R', 'R', 'R', 'R', 'R', 'R', 'D']

['R', 'R', 'U', 'R', 'R', 'R', 'L', 'U', 'R', 'D']

['R', 'R', 'R', 'U', 'R', 'R', 'U', 'U', 'L', 'D']

['R', 'R', 'R', 'R', 'L', 'D', 'D', 'G', 'R', 'D']

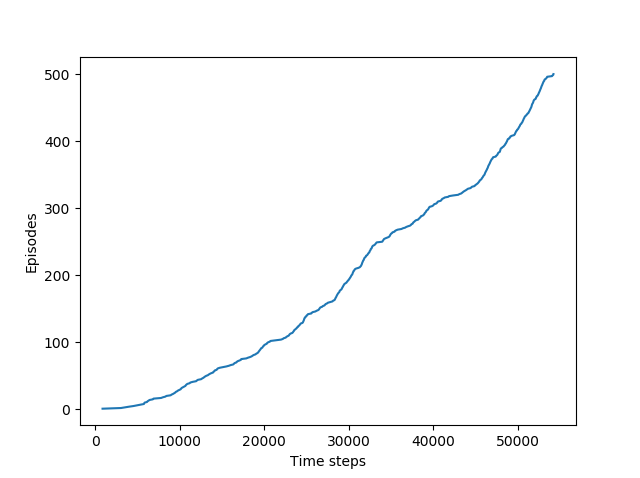
['R', 'D', 'D', 'L', 'R', 'R', 'U', 'R', 'U', 'D']

['D', 'U', 'D', 'L', 'R', 'D', 'U', 'D', 'L', 'L']

['D', 'R', 'R', 'D', 'D', 'R', 'U', 'U', 'R', 'L']

Wind strength for each column:

['0', '0', '0', '1', '1', '1', '2', '2', '1', '0']



6.11

Because the Q is independently updated of the policy be followed. In the evaluation process, it choose the max Q action to update the Q, while in the improvement process, it update it policy according to greedy policy.

6.12

Suppose action selection is greedy. Q-learning is exactly the same algorithm as Sarsa, if their initialized Q are same, the selection and update will be the same.

6.14

The task of Jack’s Car Rental is move cars from one place to another place, so several state-action pairs can lead to one state, such as A has 6 cars and 1 car is moved to A from B has the same action-after state with A has 7 cars and 1 car is moved to B. If we only consider the action-after, the state will decrease, which is likely to speed coverages.