Q0, Yes. Either anonymized or not is okay.

Q1.

(a) If we moved to a new building and a new parking lot but everything else remains the same, then TD method could update our estimation of remaining time as soon as we reach the highway entry, because we already have a convincing state value for highway, however in Monte-Carlo we have to wait til the end of the scenario to update our estimation of remaining time from the beginning.

It might not be the same in the original scenario, because in the original seenario we don't have any belief for all states, and TD will need to run many episodes to bootstrap and update every state. while monte-carlo can update all states within a single episode.

(b) In the environment where the agent only recieve +1 reward at the end of each episode, for example: maze run, using Monte-Carlo will be much better than TD method, since MC will update every state in the episode along the trajectory, while TD need much more episodes to achieve similar result.

Q2.

(a) Because the learning target that Q-learning use is $[R + \gamma \max_a Q(S', a)]$, and the maximizing over action is the target greedy policy and regardless of the behaviour policy (ε-greedy) that we use. Therefore, we are not using the behaviour policy to update action value. thus it's off-policy.

(b) Yes. If we make all action selection greedy, the policy we use will be greedy and thus the learning target $[R + \gamma Q(S', A')]$ will be the same as $[R + \gamma \max_{a'} Q(S', a')]$, which is the same learning target as Q-learning. Also, in Q-learning the soft greedy behavior policy will became greedy, then all action selection and weights updates will be all the same.

Q3.

(a) the first episode ends at the left most state. Since all the states have the same state value (0.5), most of the state values won't change except V(A), and the change equals to:

$$\alpha \cdot (r - V(A)) = 0.1 \times (-0.5) = -0.05$$

(b) No. In the long run, the smaller α value the more likely that it will converge to the optimal value. Now the two methods are both likely to converge ($\alpha = 0.01$ for MC and $\alpha = 0.05$ for TD), so there won't be a huge difference if we choose a smaller α. Nor does it will affect which algorithm is better. (TD always better than MC in this case)

(c) Larger alpha values (0.1, 0.15) could be a potential cause. and I think it does affected by the initial state value, because in practice cases ($n \neq \infty$) TD method will always have bias, and this situation may happen depended on the initial state values.

(d)

Because in 7.1 we are investigating n-step TD, and if we still use the original environment (5 states) then it will be hard to see the performance of different n value.

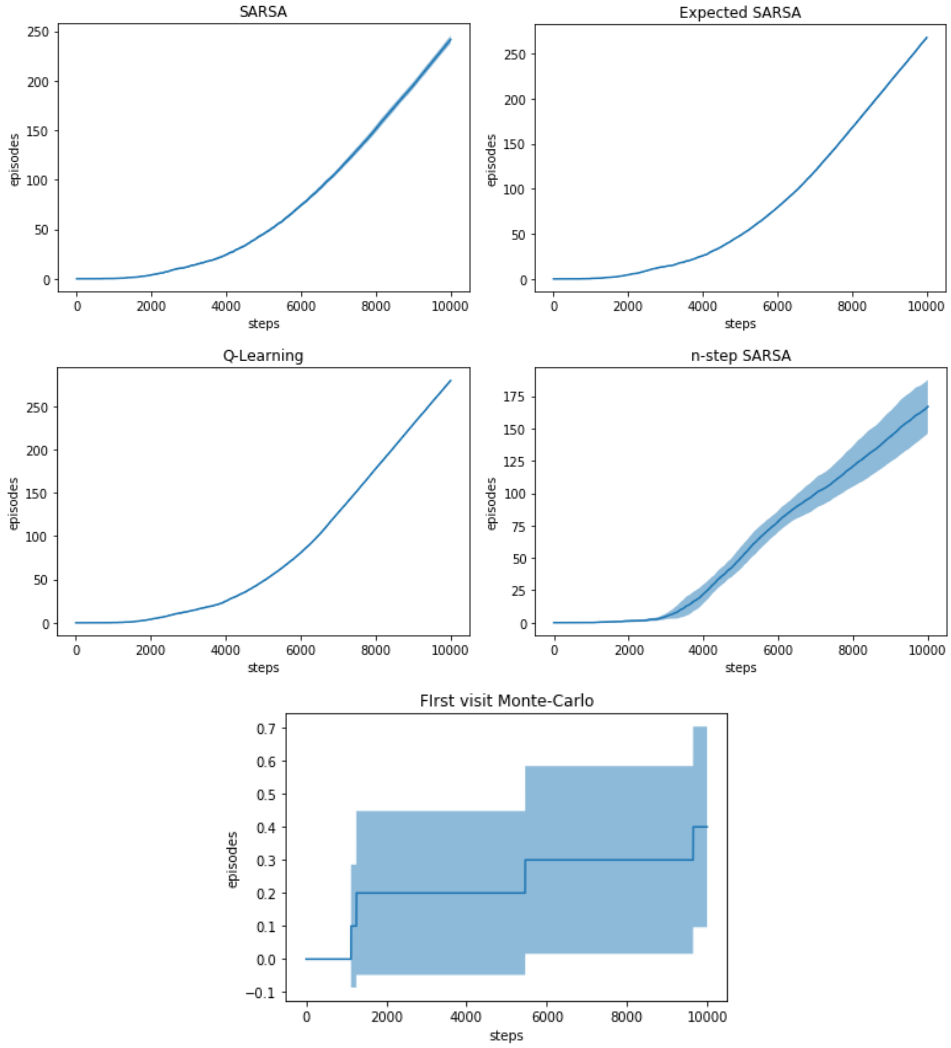
Difference on states number will affect the optimal n. Usually the optimal n is somewhere between monte-carlo and TD(0), and changing number of states will change the complexity of the environment, thus change the optimal n.

Change the left-most side to -1 won't affect the optimal value of n, since n is mostly depended on the environment. If the states and transitions remain the same in the environment, the best value of n is unlikely to be changed.

Q4.

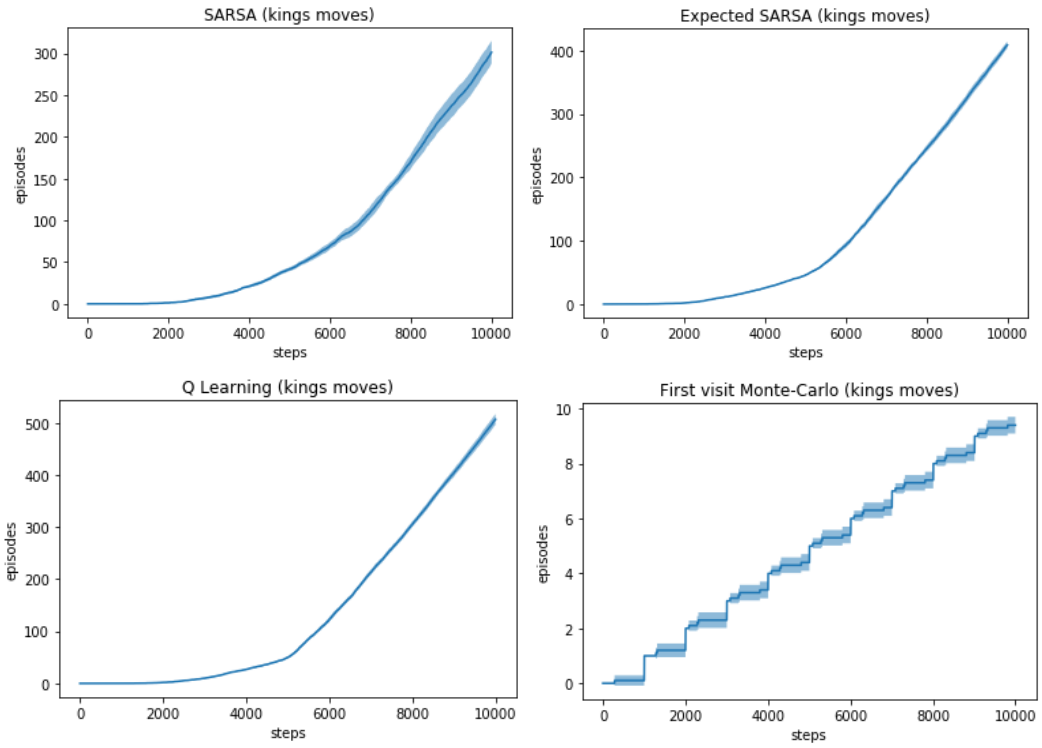
(a)

(b)



Performance: Q-Learning > Expected Sarsa > Sarsa > n-step Sarsa > First visit Monte-Carlo

(c)

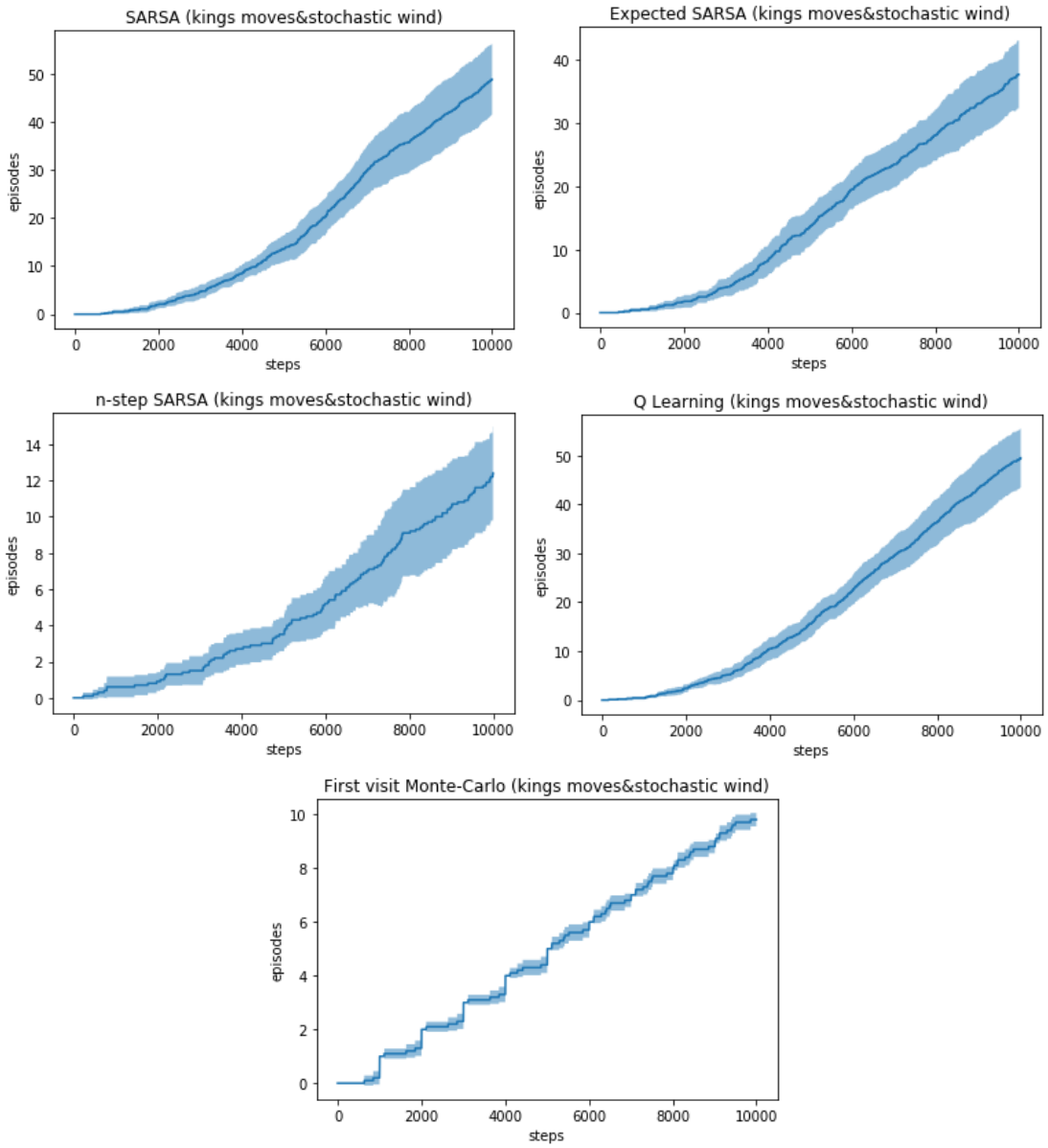


After changing action space to king's moves, all the methods achieved much better performance (Sarsa by 20%, Expected Sarsa by 55%, q-Learning by 90%, and MC by 30 times, although it's still very low and basically learned nothing). Before changing, the minimum distance that from the start to the goal is 17 steps and after changing the minimum distance is 7 steps. They still follow the pattern:

Q-Learning > Expected Sarsa > Sarsa > Monte-Carlo

I don't think including a ninth move will further improve the performance, because staying at the same location will always result in a longer episodes (at best you may be transit up by one or two steps by the wind and approach the goal, but you can always move left, right, or up to make it even closer).

(d)

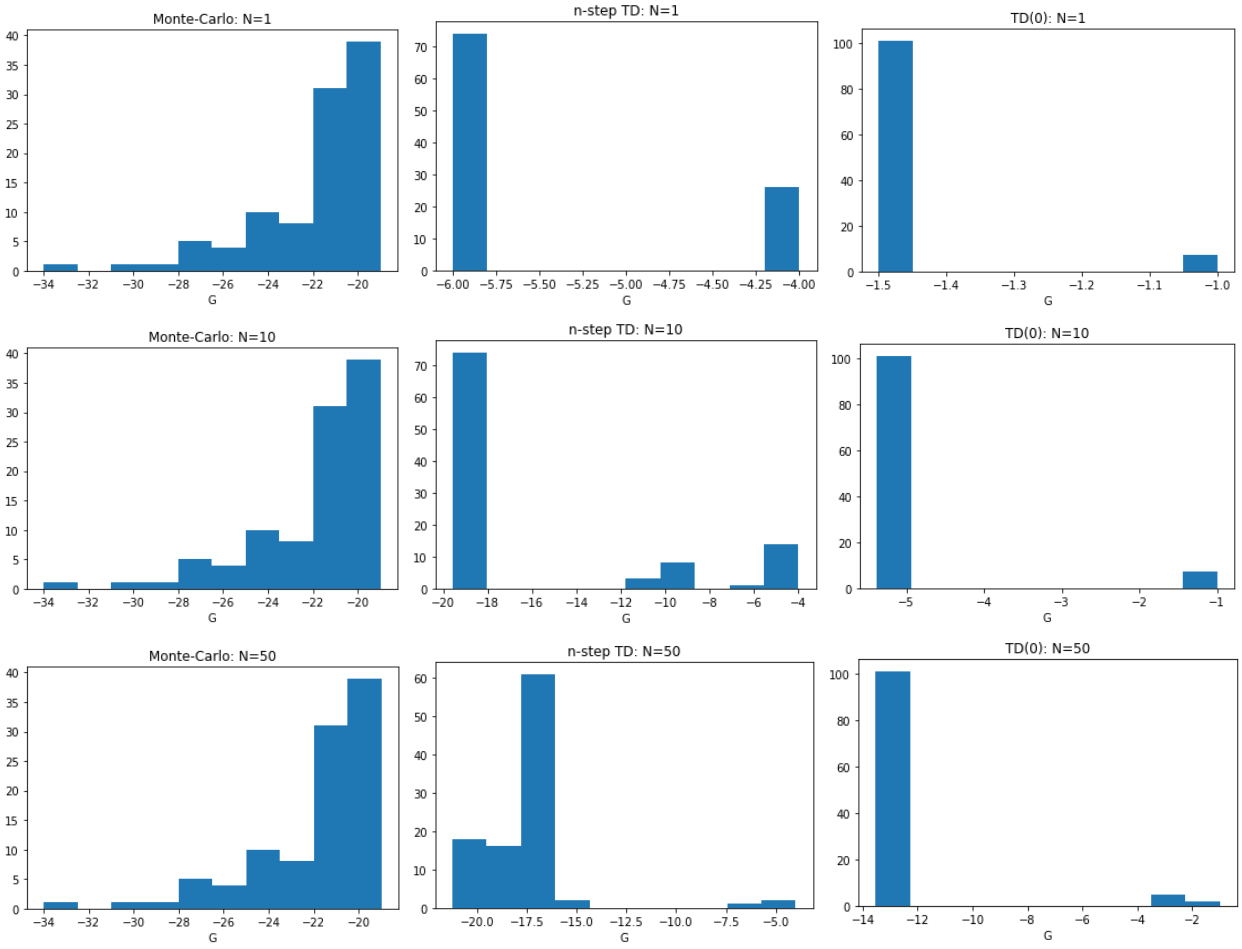


Variance increased, and performance dropped in all methods, caused by the stochastic wind, while Monte-Carlo remains the same because it still learns nothing.

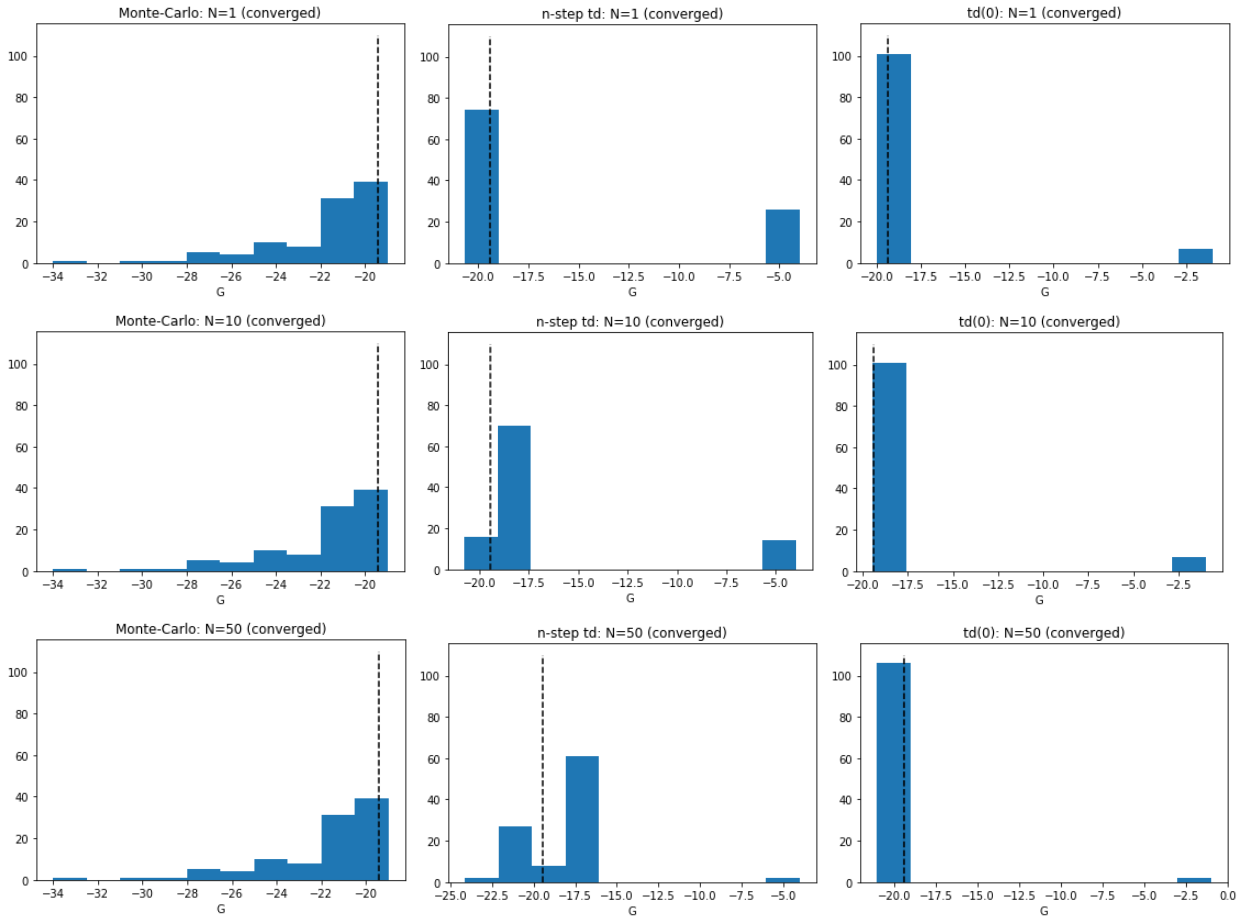
Q5.

(a)

Evaluation graphs after one single run on N episodes:



Evaluation graphs after the state values converged, based on N episodes (dashed line indicates true value of the state, calculated by Dynamic programming, V(s) = -19.42):



(b)

During the training phase, Monte-Carlo provides the most accurate value estimation (lowest bias) while the largest variance, and TD(0) provides the most stable one (lowest variance) but highest bias, and N-step TD is between these two. After the state value is converged, the bias of TD(0) and N-step TD decreased while Monte-Carlo is still the one with largest bias, and its variance remains the same.

Monte Carlo is unaffected by training set, because it doesn't need state value to do bootstrapping, whereas the other two methods performs better when the training sets grow larger, because TD methods always have bias in practice and providing larger training dataset will make the value converge to the true value asymptotically.