INFOF-409 — Learning Dynamics
Assignment 3 – Reinforcement Learning

Robin Petit
000408282
MA1-CS ULB
robpetit@ulb.ac.be

Monday 18 December 2017

1 Exercise 1

This first exercise is an N-armed bandit with Reinforcement Learning.

The selection method studied here are random, softmax, and ϵ-greedy (respectively with fixed τ and ϵ).

My ULB identification number is `000408282`, therefore the table of the exercise is table 3:

Action a_i	$Q^*_{a_i}$	σ_i
Action 1	1.3	0.9
Action 2	1.1	0.6
Action 3	0.5	0.4
Action 4	0.3	2

Figure 1 shows the distributions of probability of the reward for each action.

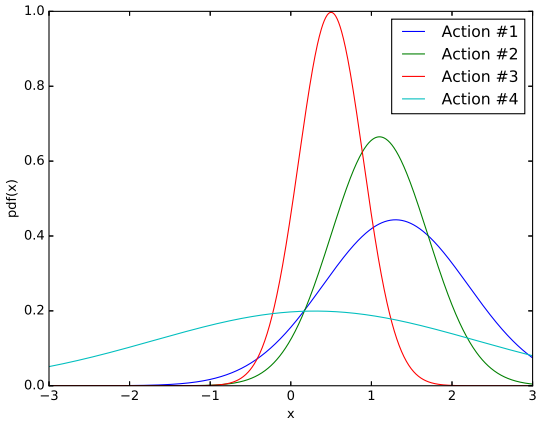


Figure 1: Probability density functions of the reward of each action.

1.1 Part 1

Figure 2 shows the evolution of the Q matrix of the agent on 1000 time steps. These results are averaged on 1000 different simulations with the same initial value of Q.

We can observe from the table above that the optimal action to take is the first one (best payoff in average). For the first arm, we can see that the softmax strategy with $\tau = .1$ has a very bad estimation of Q_{a_1} whereas all the other methods seem to converge towards the same value (which is also a bit off). Note also that the ϵ-greedy strategies find a value very quickly, and then stop updating it: it stays constant over time (at least in average).

On the two following arms, softmax with $\tau = 1$ seems to have a pretty good estimation of the Q values, but in general, the random strategy seems to be the more accurate for evaluation.

Figure 3 shows the average of the probability of playing any action for all the selection methods. The three first, i.e. the ϵ-greedy selection methods, have the highest probability to play the first action (which is optimal), and the higher ϵ, the higher the probability of the actions 2 to 4. Note that even for $\epsilon = 0$, the actions 2 to 4 have non-zero probability of being played. Reason is that as the Q matrix is initialized with only 0's, first call of $\arg\max_a Q_a$ can yield any of the four arms. Yet, the first arm is still the most played (probably because it is the arm that is the most likely to give negative reward).

On Figure 4, it is clear that the method giving the highest average payoff is ϵ-greedy, and the higher ϵ, the higher the average payoff.

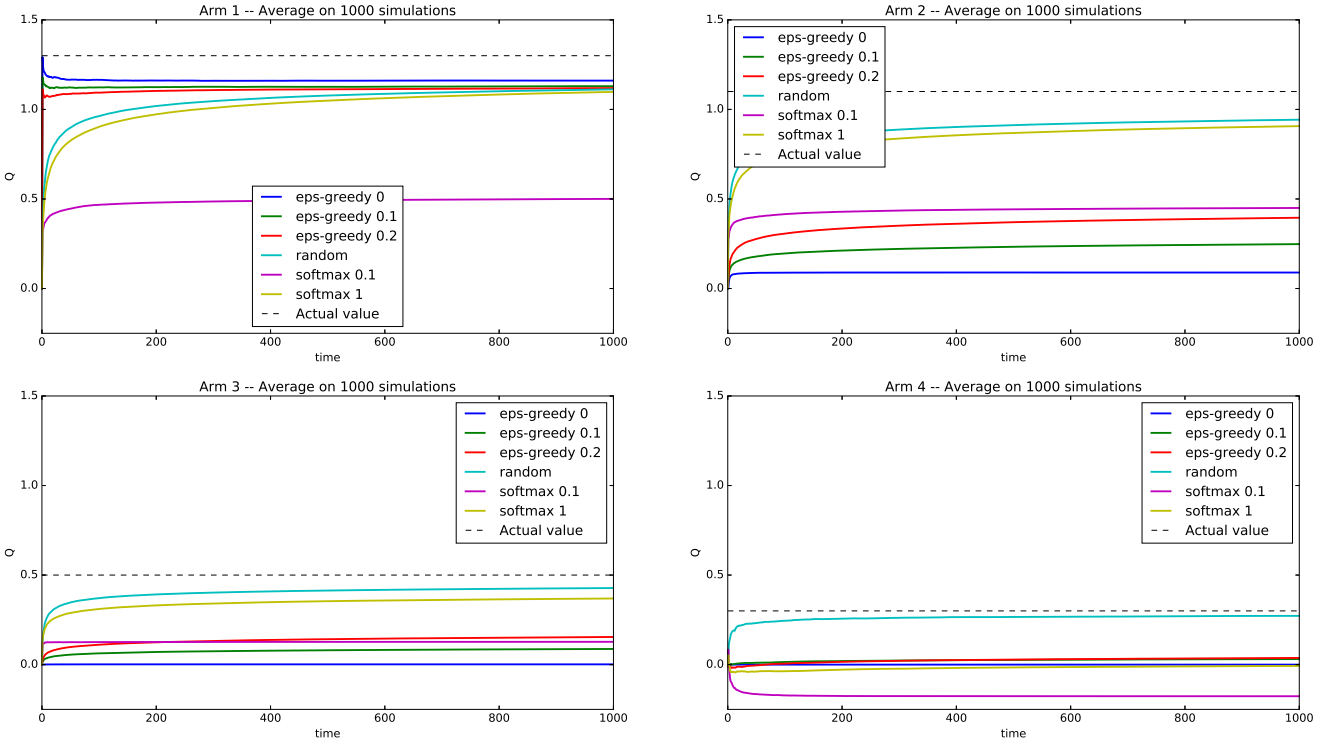


Figure 2: Mean evaluation of each Q value for each selection method performed on 1000 simulations.

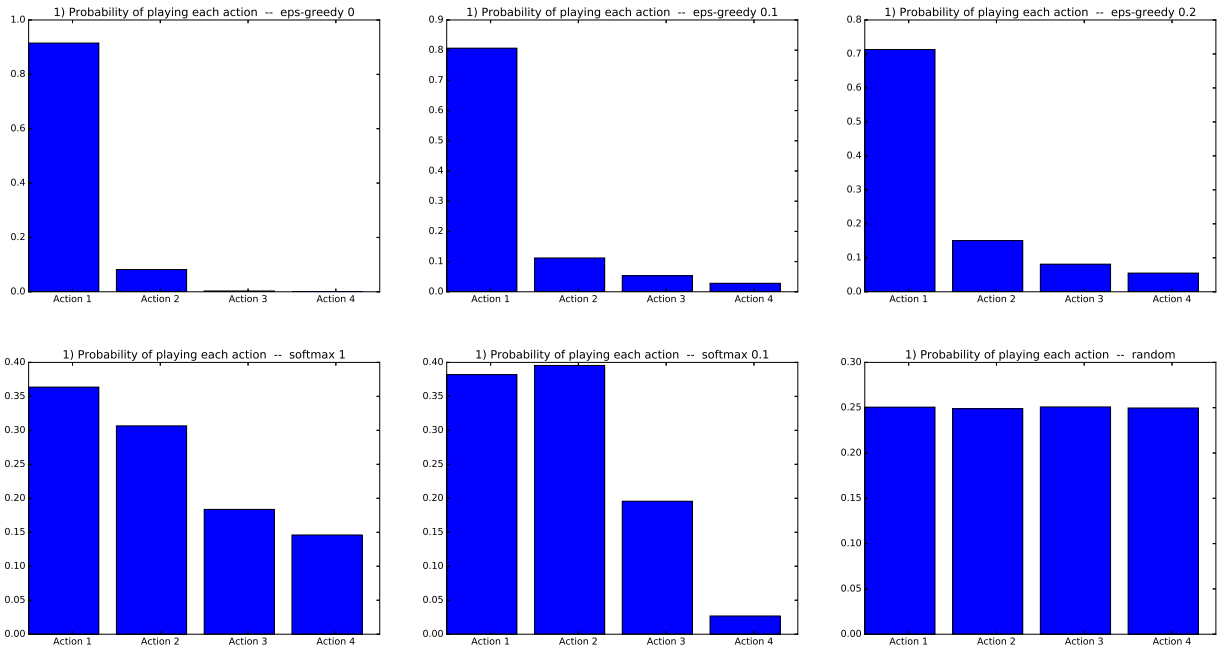


Figure 3: Probability of playing any action for every action selection method.

About softmax, the higher the temperature τ, the more exploring the strategy. Therefore it is logical that softmax with $\tau = 0.1$ gives a higher average reward than softmax with $\tau = 1$. Also, as can be seen on

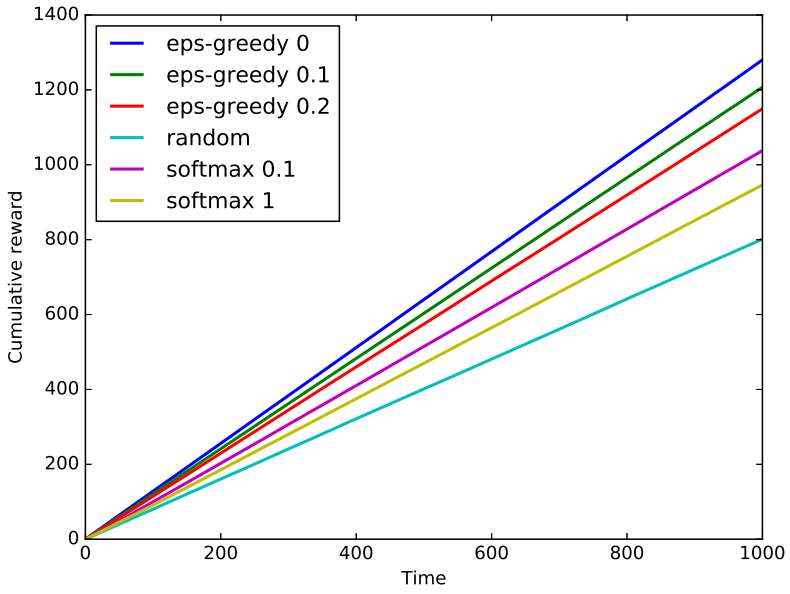


Figure 4: Cumulative reward of each algorithm.

Figure 3, softmax with $\tau = 1$ gives a higher probability for the second action, even though it is not the optimal one.

It is interesting to remark that the random strategy gives the lowest average cumulative reward (which is expected since there is no proper exploitation) but has the best **general** estimation of the Q matrix: as the random strategy plays each action with equal probability, there is no exploitation but only exploration which leads to a very good knowledge of the rewards in each situation.

This is the reason of the use of the ϵ-greedy strategy: using random play for exploration, and then arg max for the exploitation.

1.2 Part 2

In this second part of the first exercise, the only difference with the previous part is that the standard deviation of each action reward has been doubled, therefore the estimation of the Q matrix is blurred for the agent, yet the mean stays the same. So the end results should be identical, unless the processes are highly sensitive to high variance in the observation of the rewards.

Figure 5 shows the estimation of the Q value for each arm and each algorithm. We can clearly see that the agent has a harder time finding the actual value of the Q matrix, especially for the first arm (the optimal one). The ϵ-greedy strategy that was the best previously becomes pretty inefficient. The random strategy

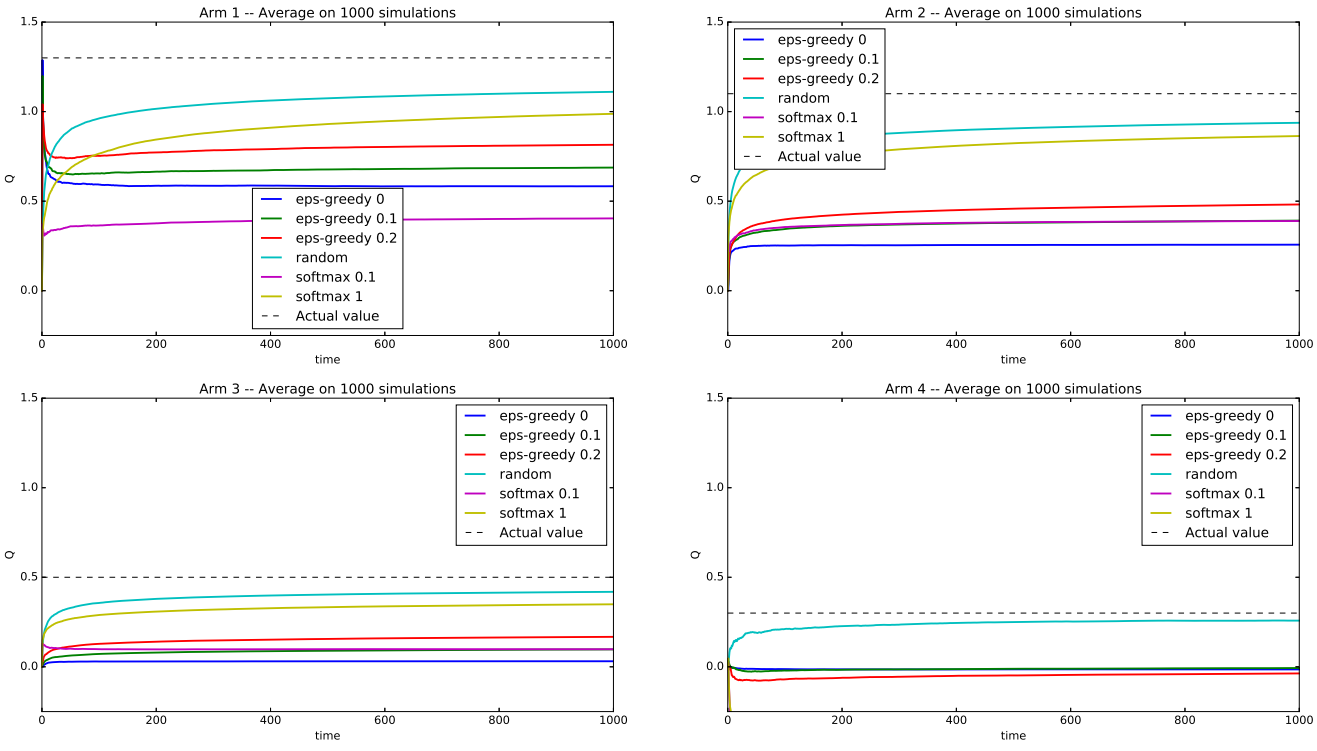


Figure 5: Mean evaluation of each Q value for each selection method performed on 1000 simulations.

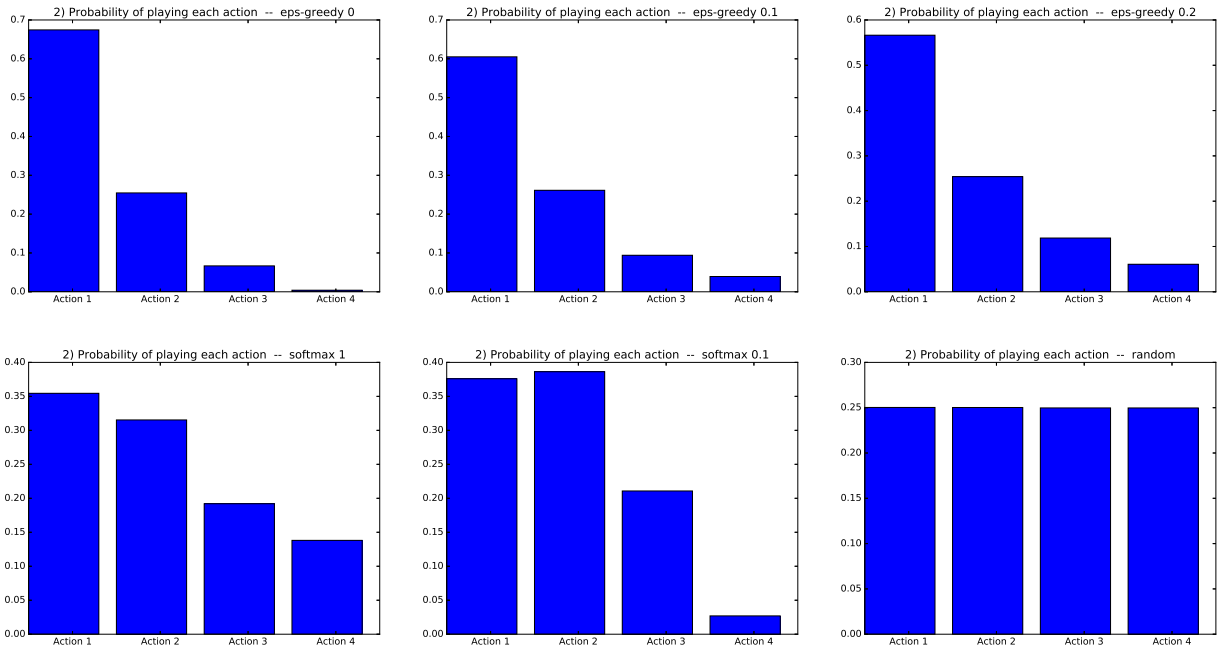


Figure 6: Probability of playing any action for every action selection method.

doesn't seem shaken for its part: it stays pretty good at estimating each individual value, and becomes the best estimator. Regarding the ϵ-greedy algorithm, we can see that the higher ϵ, the best the estimation, and such for each arm.

Also, unless about the fourth arm, the softmax strategy with $\tau = 1$ has become the second best strategy at estimating Q. This shows that softmax, despite being equally efficient at estimating the best action (but better at estimating the other ones) compared to ϵ-greedy, is more robust to high standard deviations than ϵ-greedy.

Figure 6 shows the probability of playing each action for each separate algorithm. First observation is that even though the ϵ-greedy strategies have a clearly less accurate evaluation of each Q_a, it still finds the right order. Yet, even though the best one is the first one, the number of times the other actions have been played is greater than in the previous situation. This means that more steps have been needed to the agent in order to achieve a steady state.

Let's also note that the difference between the ϵ-greedy strategies is that the optimal action (first arm) is played less often, and that the lost proportion is set mainly on action 4.

Regarding the softmax strategies, as mentioned above, it is more robust to the variance in the observations since the frequencies displayed in Figure 6 are very similar to those presented in Figure 3: for $\tau = 1$, the probability of playing each action is decreasing from 1 to 4, and for $\tau = .1$, it is again the second action that has the highest probability, despite not being the optimal one.

And obviously, the random strategy has the same probabilities as before since it is not sensitive to the variance: it does not depend on the observations...

It is also worth noting that the average cumulative reward (Figure 7)of ϵ-greedy has lowered a bit, which can be explained by the increased number of plays of the non-optimal actions, leading to a lower reward. As these rewards are taken in average over a big number of simulations, the variance shouldn't have influence

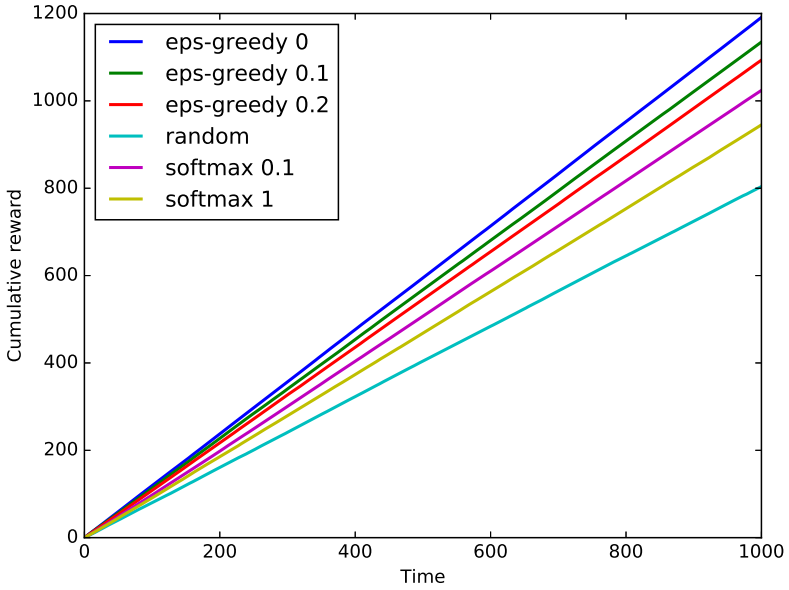


Figure 7: Cumulative reward of each algorithm.

on the rewards in Figure 7.

Also, as the proportion of each action doesn't change drastically for the softmax and random strategies, it is normal to not see a difference in the average cumulative rewards of these strategies.

1.3 Part 3

This third part of the first exercise is about dynamic parameter: the strategies implemented are the same (i.e. ϵ-greedy and softmax) but here, the parameter (i.e. ϵ or τ) is not fixed but decaying with time: $\epsilon(t) = t^{-1/2}$ and $\tau(t) = 4\frac{T-t}{T}$ with T the number of steps (i.e. 1000). The decaying of ϵ is going slowly (from 1 to .03) but with a decreasing derivative (in absolute value) and goes *asymptotically* towards 0 whereas the decreasing of τ goes faster (from 4 to 0) with constant derivative, and reaches 0 at $t = T$.

Figure 8 shows the evaluation of the Q values for both these methods. We can see that for the three first arms, the softmax with decreasing temperature performs similar than softmax with fixed temperature, and for the last arm softmax performs slightly better at evaluating Q_{a_4}. Decreasing ϵ for its part doesn't perform as good as fixed ϵ.

Also, in the first part, we remarked that the ϵ-greedy methods reached stability in their Q evaluation (especially of the optimal action) very quickly. Here, we can observe in the first subplot of Figure 8 that the ϵ-greedy strategy with decreasing ϵ takes longer to find stability (probably also due to the longer phase of exploration).

When looking at Figure 9, we can observe that softmax has the same distribution as in Figure 6, which is coherent with the fact that the evaluation of Q is the same as well. On the other hand, we can see that for ϵ-greedy, the distribution is widely different. This is probably due to the fact that $t \mapsto t^{-1/2}$ decreases very slowly, and then $\epsilon(t)$ stays quite high for a long time (drops below .1 at time $t = 100$) leading to a long phase of exploration and not exploitation. Yet, when the agent becomes more prompt to exploitation, the learning rate (which is set to be $t \mapsto t^{-1}$) is too low for wide updates in the Q values.

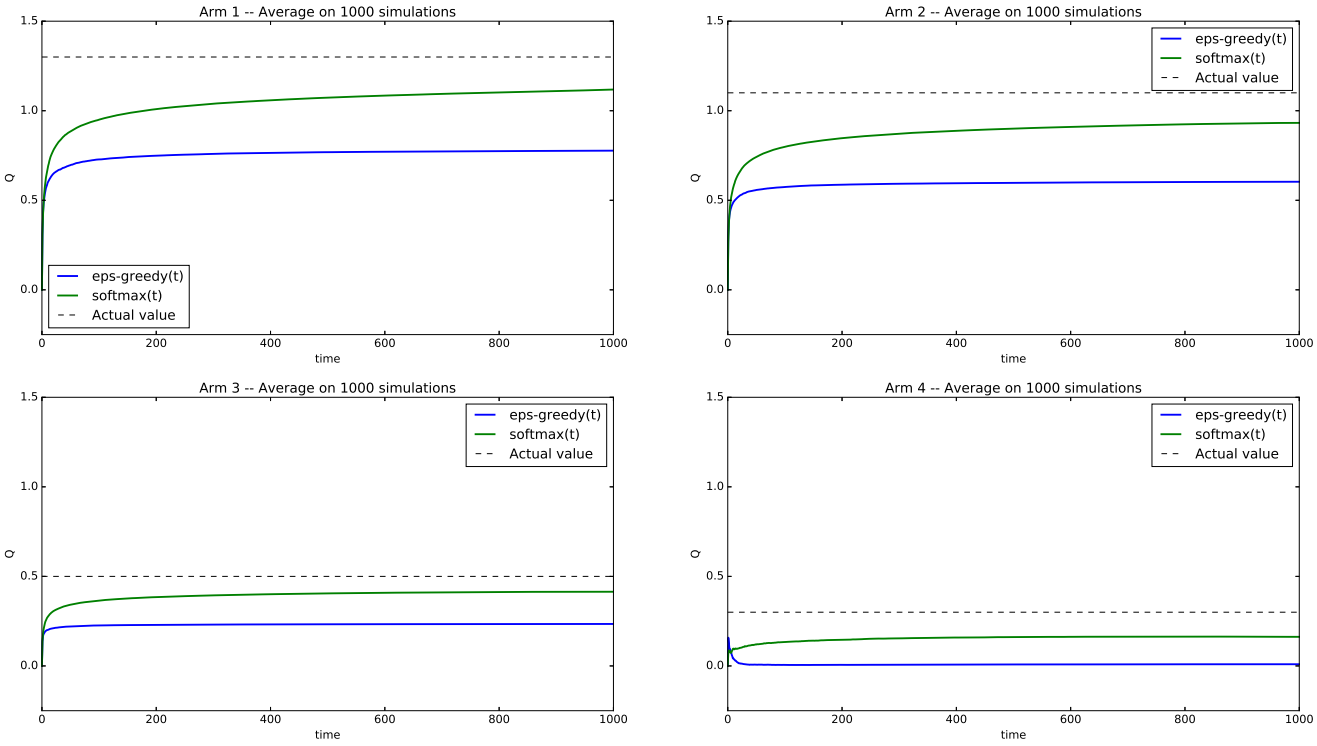


Figure 8: Mean evaluation of each Q value for each selection method performed on 1000 simulations.

This would also explain why even though the estimation of the Q values are off, the average cumulative

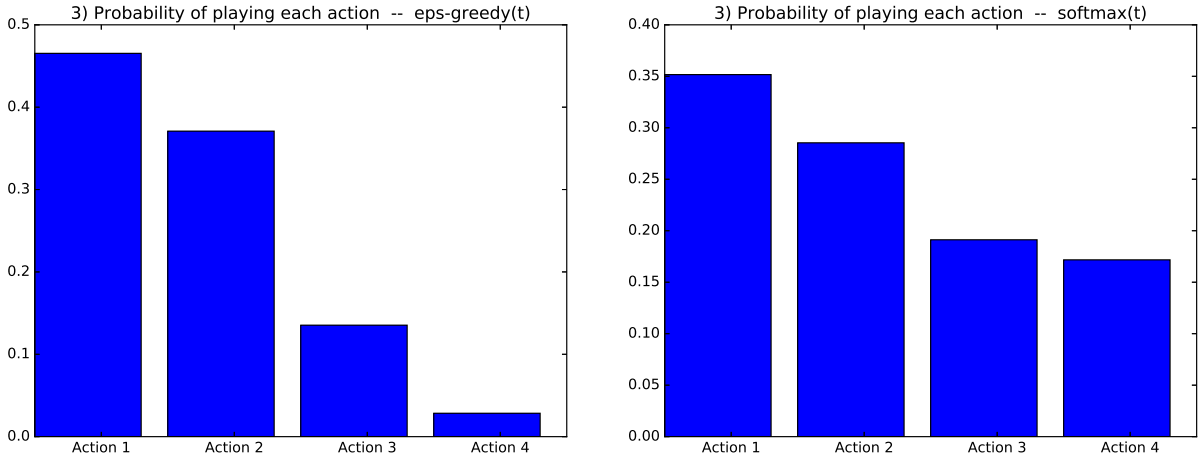


Figure 9: Probability of playing any action for every action selection method.

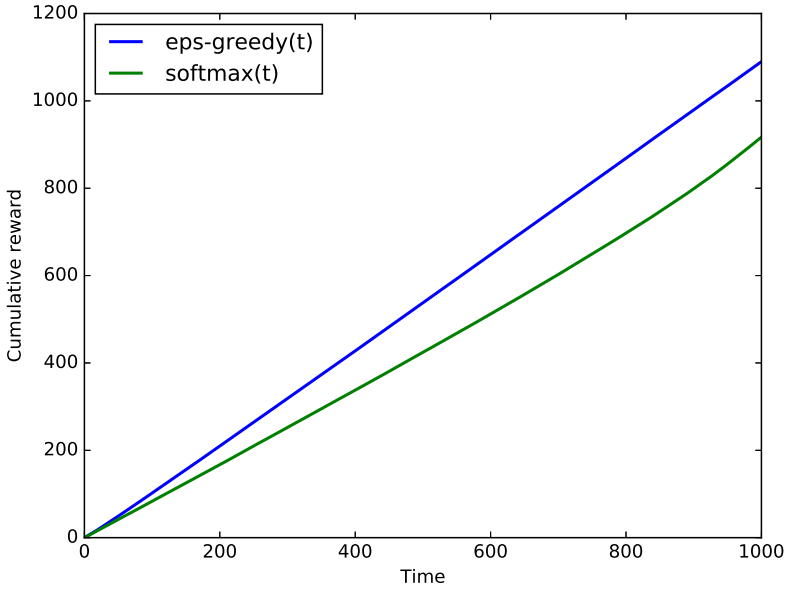


Figure 10: Cumulative reward of each algorithm.

rewards are very close to those of the first part of this exercise.

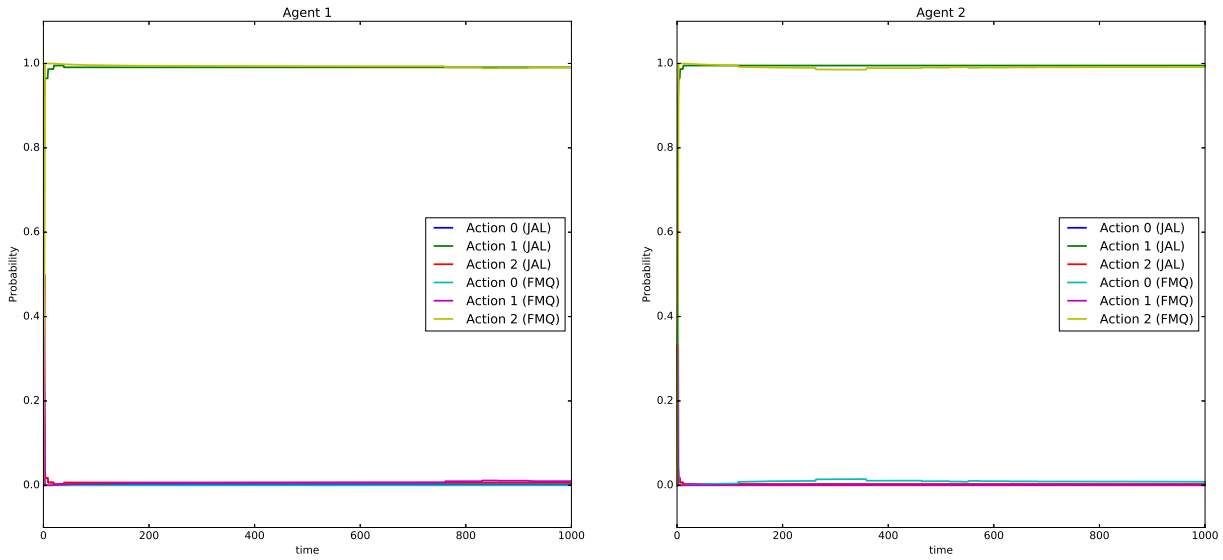
The major conclusion of this last part of the first exercise is that softmax is very robust, and that the decay in τ does not bring much to the study of **this particular case**, and that the ϵ-greedy strategy suffered quite a bit of the change from fixed ϵ to decaying ϵ, probably because of the difference in decreasing rate between $\epsilon(t)$ and $\alpha(t)$ the learning rate. If the decay of ϵ was faster, results would have probably been closer to the ones with fixed ϵ (even better).

2 Exercise 2

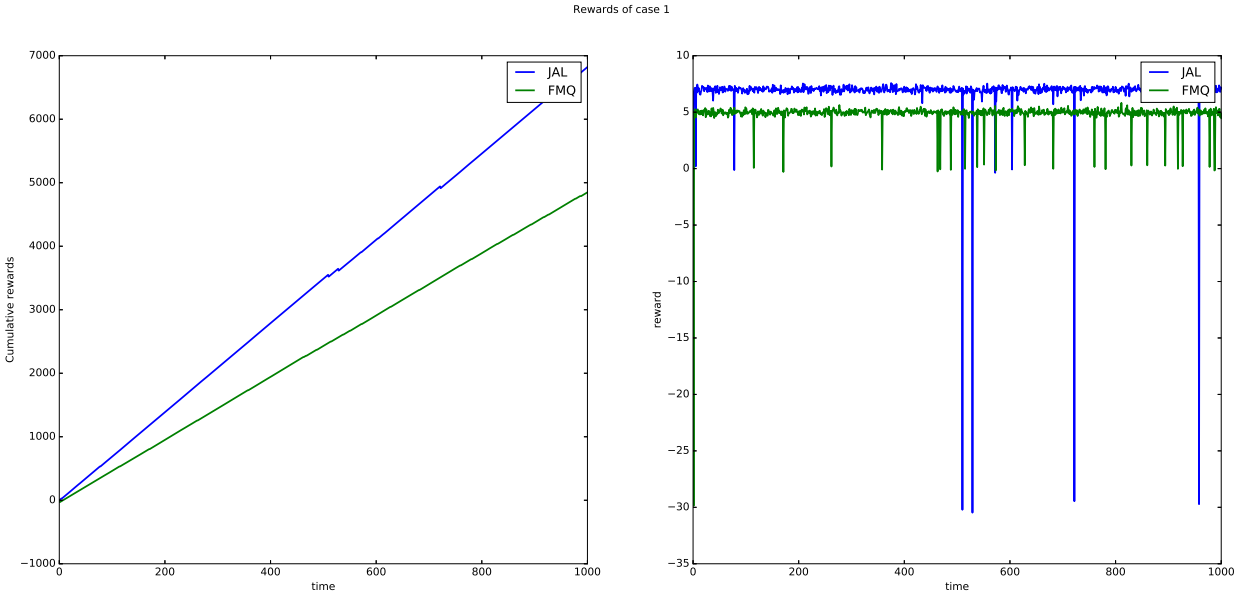
This second objective is a stochastic climbing game with two agents and Reinforcement Learning. First part of this study is when both agents learn as JALs (Joint Actions Learners), so they try to evaluate the Q value of each couple of action, and the second part is when both agents learn individually using FMQ heuristic.

The JAL agents have a Q matrix initialized at 1 for each available action and with a belief vector for the opponent's actions (the term *opponent* is used here, even if the two agents don't have conflicting interest and don't truly act as opponents) initialized at 0 for each opponent action.

2.1 Part 1



(a) Probability of playing each action for each agent in both JAL vs JAL and FMQ vs FMQ.



(b) Immediate and cumulative rewards of both agents.

Figure 11: State of both agents in first situation of standard deviations.

2.1.1 $(\sigma_0, \sigma_1, \sigma) = (.2, .2, .2)$

Figure 11a shows the probability of each agent to play each action in both situations JAL vs JAL and FMQ vs FMQ.

We can see that for JAL vs JAL, the reached equilibrium is action 1 for both agents but for FMQ vs FMQ

the equilibrium is action 2 for both agents.

This explains why on Figure 11b the general mean of the reward per step is around 7 for JAL agents and around 5 for FMQ agents. FMQ is heuristic and then tries to play the best action with the restricted knowledge available. As no huge loss is possible with 3rd action of each player, it seems reasonable to the FMQ agents to play this one.

Also, we can see on Figure 11b that the JAL agents have a negative reward of -30 from time to time due to the fact that if one agents plays 1 and the other 0, then the payoff suddenly drops, whereas for FMQ agents, as they play 2, if only one player plays another action, the only possible reward is 0 (which we can also observe on the figure).

As well, we can see that agents sometimes don't play their best action (non-zero probability for softmax) and they play action 0, they didn't play it together (leading to reward of -30 but not of 11).

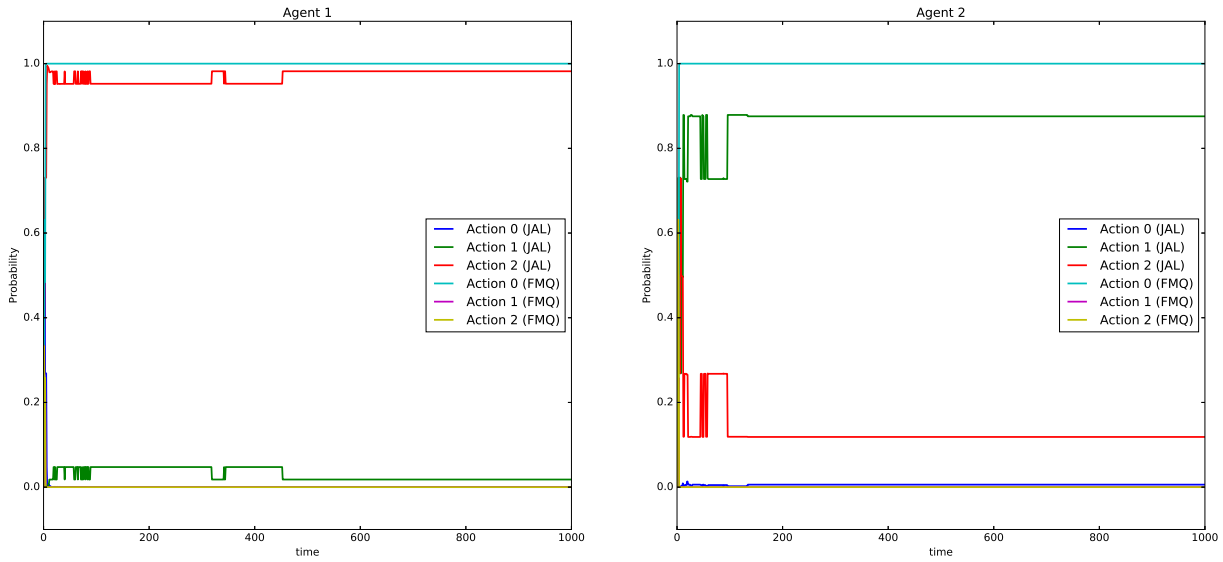
Note that as the standard deviation is very small for each reward, the reward seems pretty constant around 7 for JAL and 5 for FMQ.

2.1.2 $(\sigma_0, \sigma_1, \sigma) = (4, .1, .1)$

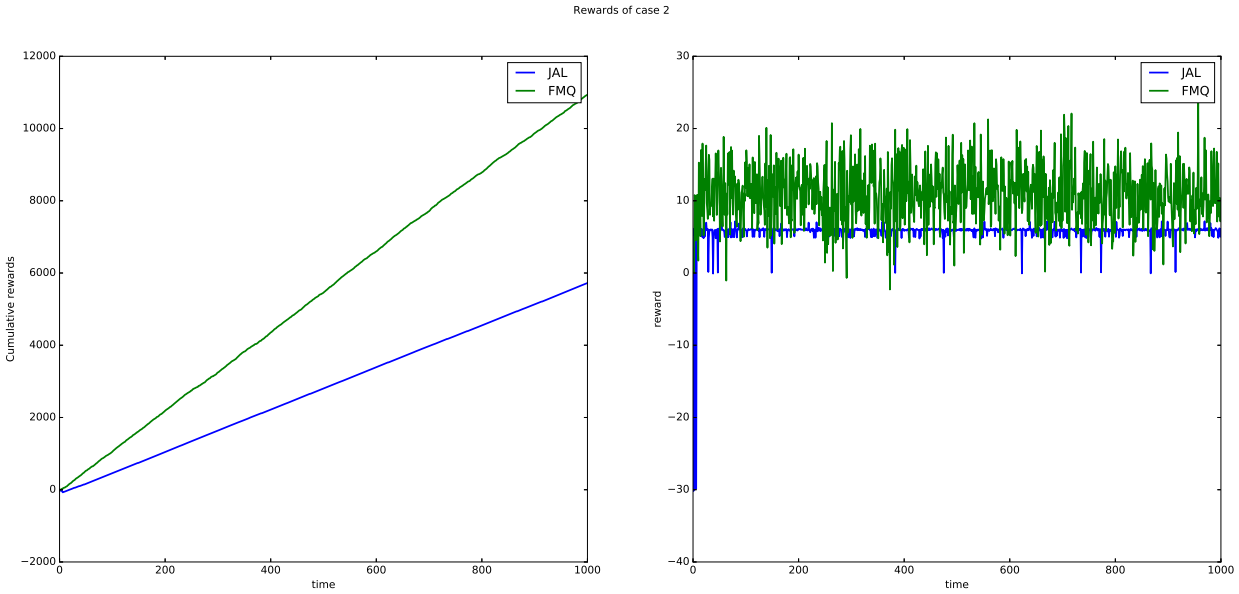
In the second case of standard deviations, σ_0 (i.e. the std of the reward of joint action $(0, 0)$) is way bigger than the standard deviation of the other rewards.

We can see on Figure 12a that in this case, the equilibrium reached for FMQ agents is the optimal Nash Equilibrium: both agents play action 0, leading to an immediate reward around 11 at each step, yet it looks a lot more spread than previously due to the higher standard deviation. Also, JAL agents have reached strategy where first player plays 2 and second player plays 1 which correspond to an average payoff of 6 (as the reward matrix has been transposed).

As opposed to the previous case, we can then see on Figure 12b that the FMQ agents perform way better and have a bigger cumulative reward than the JAL agents, which is the opposite of what was observed on Figure 11b.



(a) Probability of playing each action for each agent in both JAL vs JAL and FMQ vs FMQ.

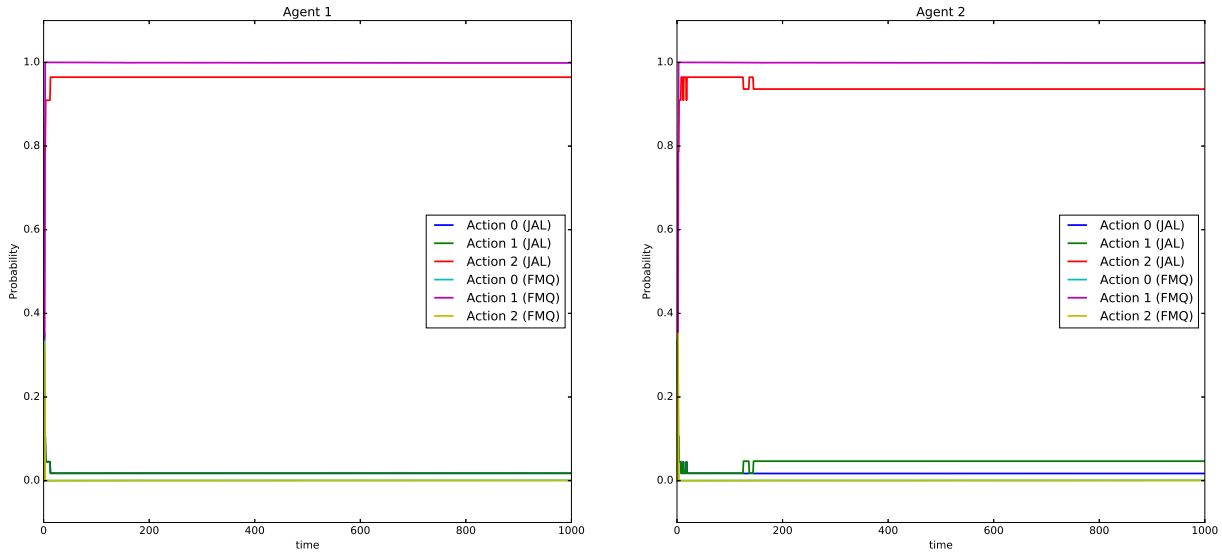


(b) Immediate and cumulative rewards of both agents.

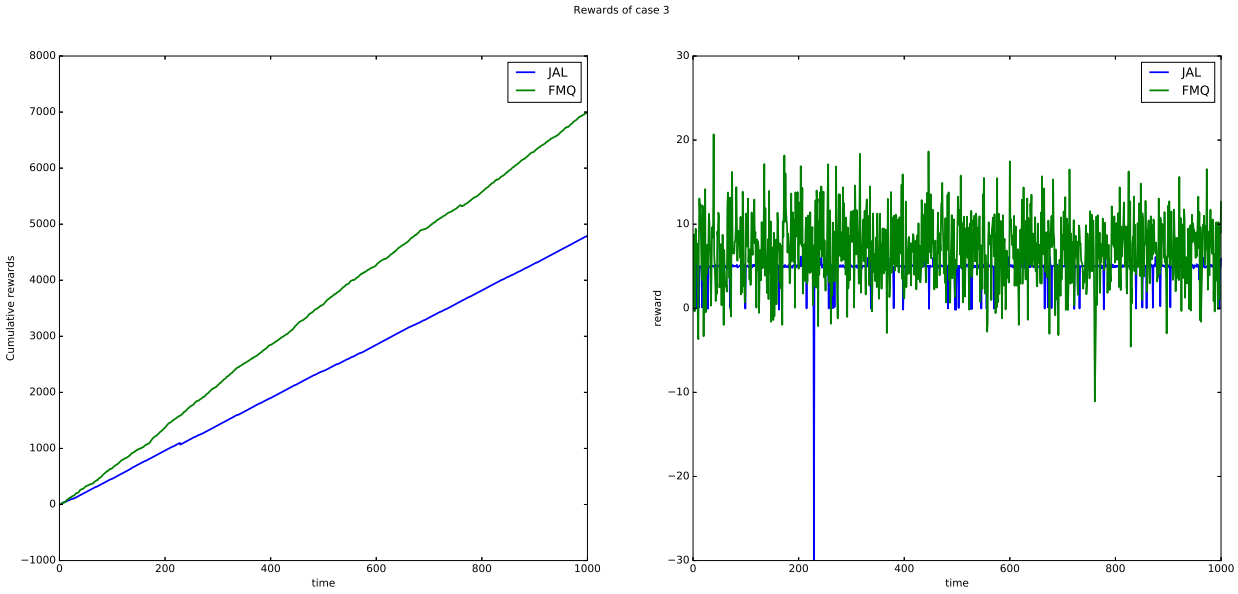
Figure 12: State of both agents in second situation of standard deviations.

2.1.3 $(\sigma_0, \sigma_1, \sigma) = (.1, 4, .1)$

In this last std situation, the reached equilibrium is action 2 for both players for JAL agents, and action 1 for both agents for FMQ agents, as can be seen on Figure 13a. Then, as in the previous case, the FMQ agents perform better than the JAL agents regarding to the cumulative rewards (and the immediate rewards).



(a) Probability of playing each action for each agent in both JAL vs JAL and FMQ vs FMQ.



(b) Immediate and cumulative rewards of both agents.

Figure 13: State of both agents in third situation of standard deviations.

Even though the standard deviation in the case of joint action $(1, 1)$ that can lead to a small reward (as well as a big one) in this situation, this is still the solution accessed by FMQ agents.

2.2 Part 2

If agents were made independent learners, they would have had a harder time finding a stable strategy, but the behaviour after a sufficiently long time would have probably been the same. FMQ heuristic method can be partially considered as independent learning since the Q values that are estimated are not the Q values of the joint actions but the Q values of the individual actions; yet the Q values use knowledge on the other agent's actions. Totally independent learners should only consider their own rewards and actions, without observing any information regarding the other agent.

If only the arg max was used to find the action to play (i.e. ϵ-greedy strategy), the exploration phase would either not truly exist *per se* (if $\epsilon = 0$) or be highly reduced (if $\epsilon \neq 0$ but is sufficiently small to consider the strategy to be ϵ-greedy and not random). But this phase is crucial for each agent to have basic knowledge on their opponent actions, and on the resulting rewards.

2.3 Possible improvements

In JAL, the counting step only took the opponent's actions into consideration. It would be possible to count not only the actions of the other agent $C_i^j(a)$ (number of steps when player i observed player j play action a), but the actions of the opponent regarding the actions of the players $C_i^j(a|b)$ (number of steps when player i played action b and observed player j play action a).

This is a larger assumption of knowledge of each agent about its opponents but could bring better confidence in the actions to take since the final knowledge can give a better insight of the opponent's strategy.