Part A

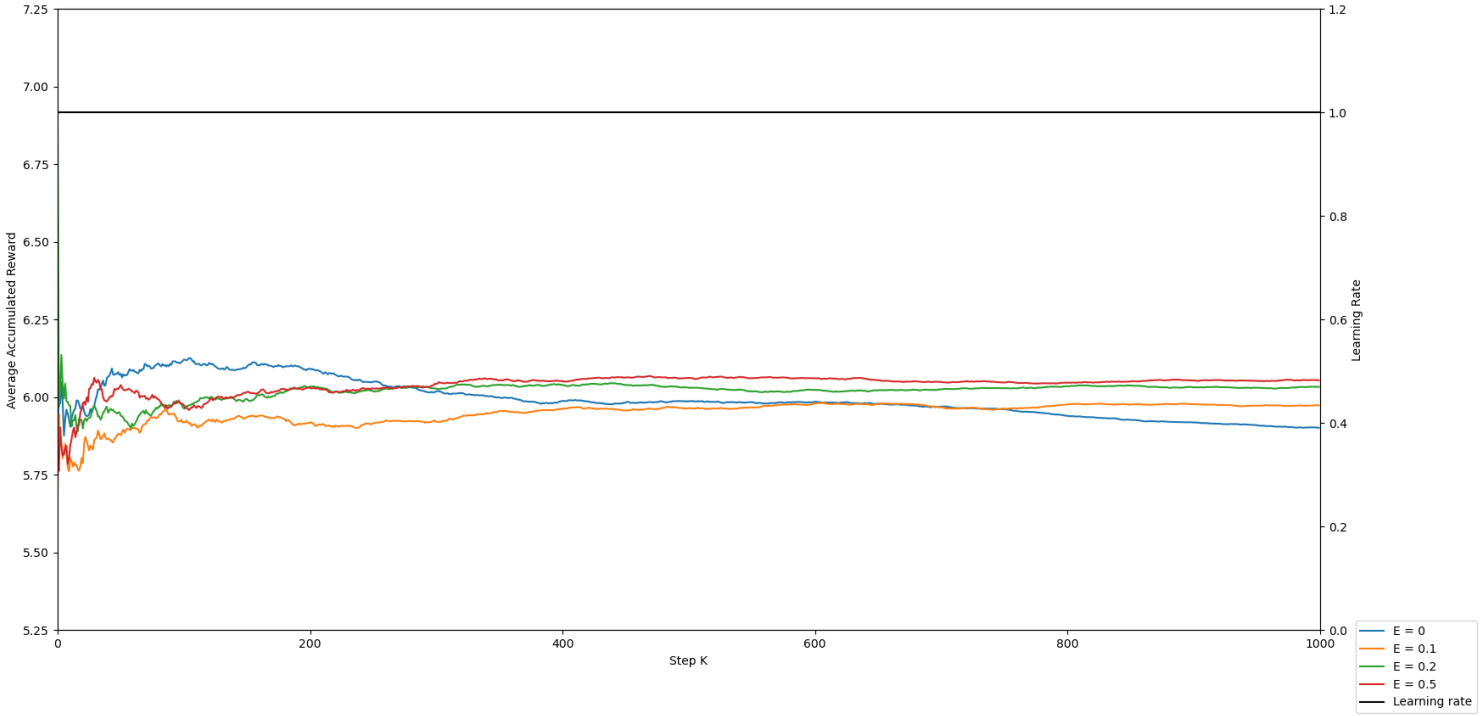

Figure 1: Average Accumulated Reward for α=1

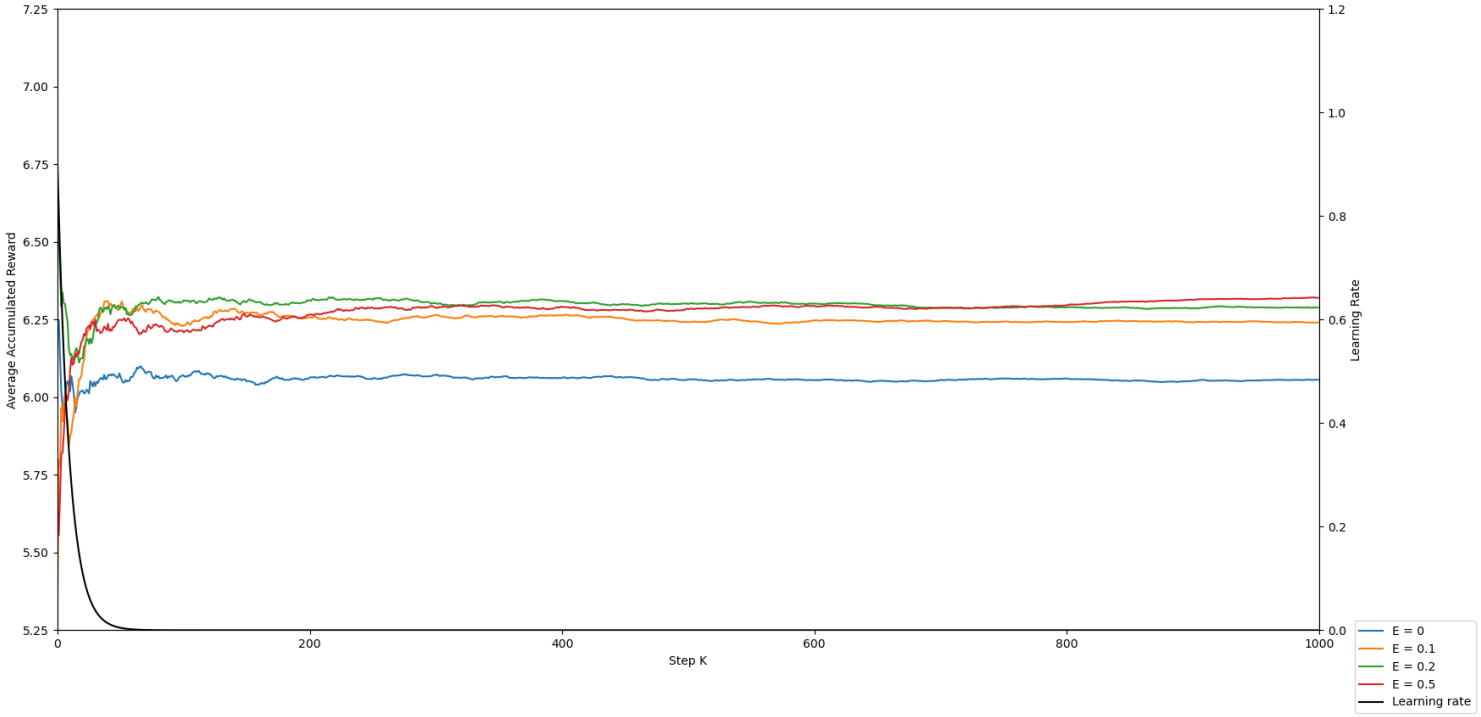

Figure 2: Average Accumulated Reward for $\alpha = 0.9^K$

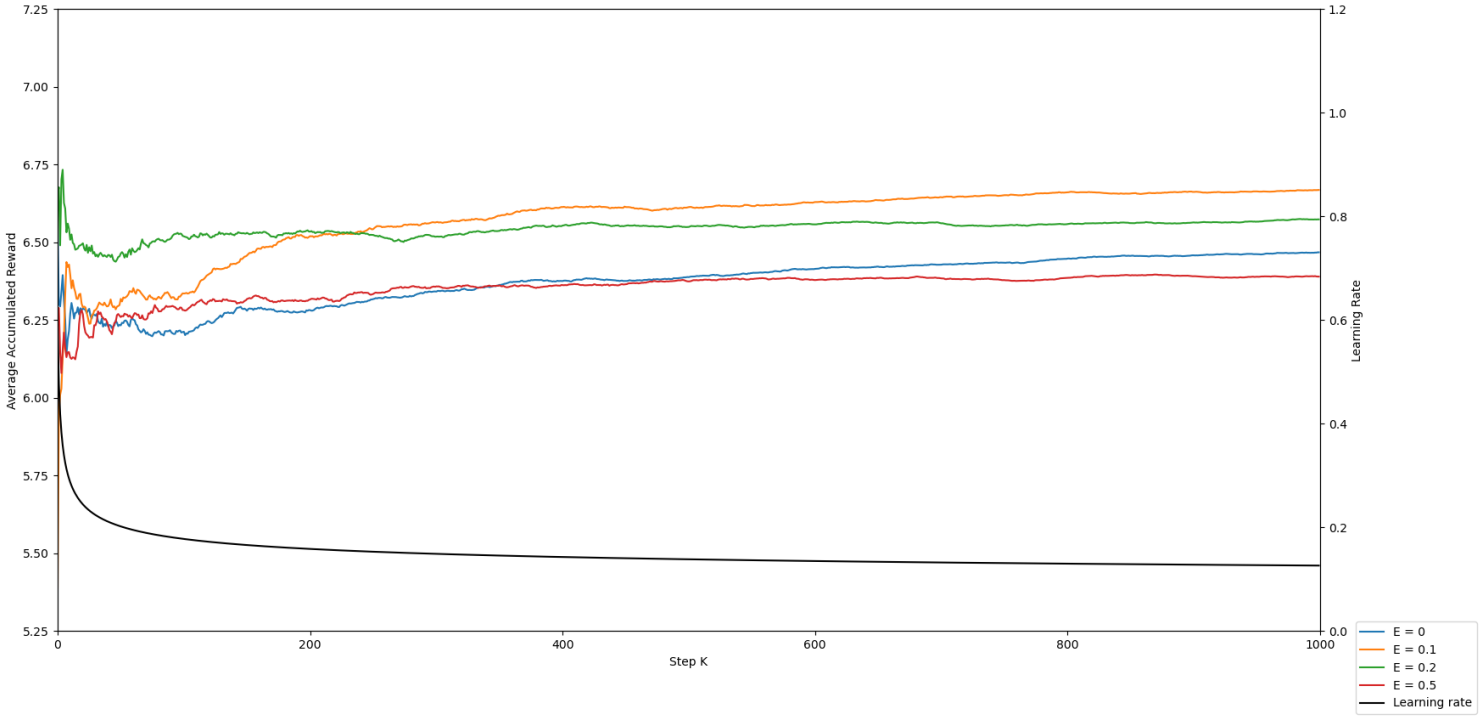

Figure 3: Average Accumulated Reward for $\alpha = \frac{1}{1 + ln(1 + K)}$

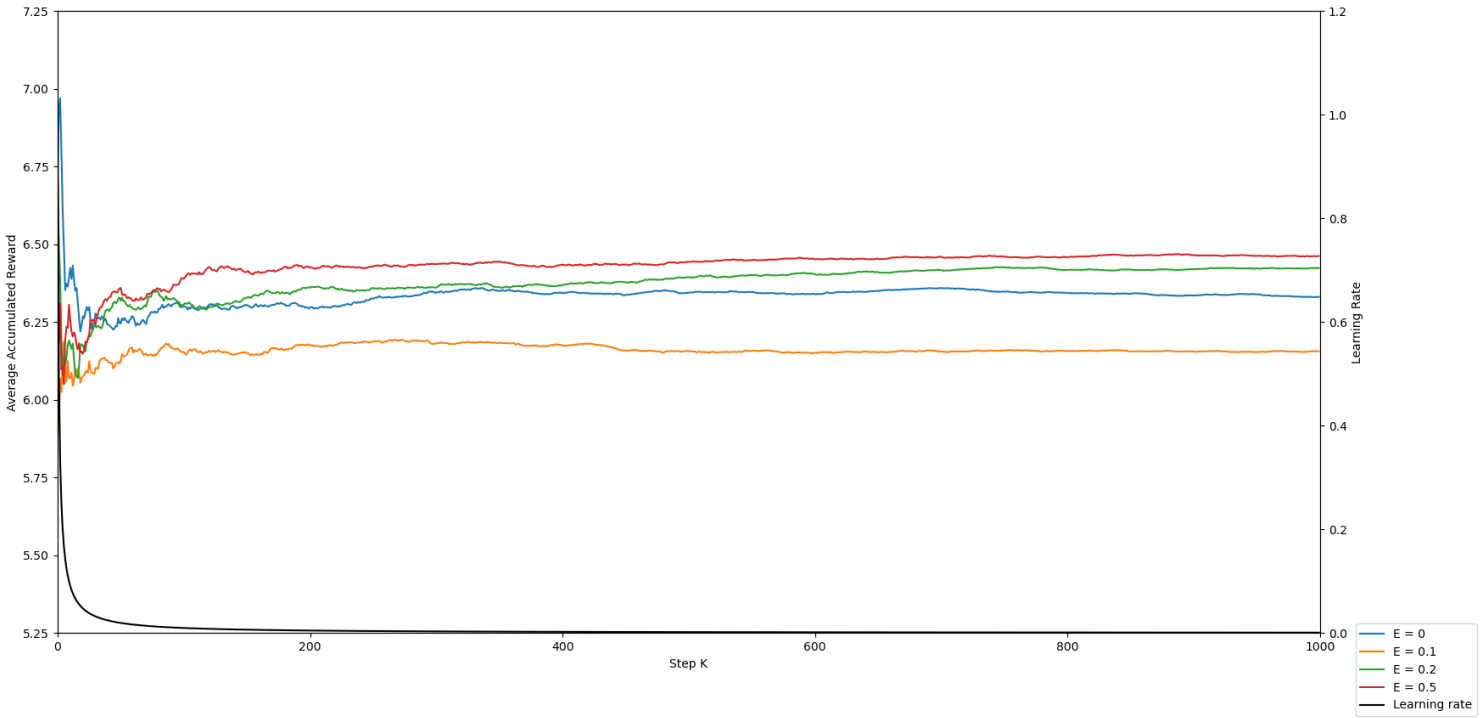

Figure 4: Average Accumulated Reward for $\alpha = \frac{1}{K}$

Epsilon-Greedy	Average of action value $Q(a^1)$ of 100 runs	True action value $Q^*(a^1)$	Average of action value $Q(a^2)$ of 100 runs	True action value $Q^*(a^2)$
ε = 0 (greedy)	1.248	5	-1.540	7
ε = 0.1	2.193	5	3.520	7
ε = 0.2	2.932	5	3.520	7
ε = 0.5 (random)	4.332	5	4.793	7

Table 1: Average final Q-values for α=1

Epsilon-Greedy	Average of action value $Q(a^1)$ of 100 runs	True action value $Q^*(a^1)$	Average of action value $Q(a^2)$ of 100 runs	True action value $Q^*(a^2)$
ε = 0 (greedy)	3.114	5	3.875	7
ε = 0.1	3.478	5	4.985	7
ε = 0.2	3.821	5	5.698	7
ε = 0.5 (random)	4.623	5	6.407	7

Table 2: Average final Q-values for $\alpha = 0.9^K$

Epsilon-Greedy	Average of action value $Q(a^1)$ of 100 runs	True action value $Q^*(a^1)$	Average of action value $Q(a^2)$ of 100 runs	True action value $Q^*(a^2)$
ε = 0 (greedy)	3.334	5	5.962	7
ε = 0.1	4.477	5	6.840	7
ε = 0.2	4.722	5	6.715	7
ε = 0.5 (random)	4.854	5	6.859	7

Table 3: Average final Q-values for $\alpha = \frac{1}{1 + ln(1 + K)}$

Epsilon-Greedy	Average of action value $Q(a^1)$ of 100 runs	True action value $Q^*(a^1)$	Average of action value $Q(a^2)$ of 100 runs	True action value $Q^*(a^2)$
ε = 0 (greedy)	4.619	5	5.473	7
ε = 0.1	4.515	5	5.350	7
ε = 0.2	4.487	5	6.241	7
ε = 0.5 (random)	4.812	5	6.895	7

Table 4: Average final Q-values for $\alpha = \frac{1}{K}$

Learning Rate:

I decided to include the evolution of the learning rate in the plots, since its value is relevant in relation to the exploration/exploitation dynamic.

Given the update of $Q_{(a)} = Q_{(a)} + \alpha * (r - Q_{(a)})$ a learning rate that is always close to 1 will mostly consider the last reward and not the history of rewards (it will update faster but with "worse" information, and the exploitation will be similar to the exploration), while a learning rate that is always close to 0 will mostly consider the history of rewards and not the last reward (it will update slower but with "better" information, and the exploitation will be more distinct to the exploration).

Given that initially we don't have a history of rewards and we need to learn it, we want to have a learning rate that is initially high. As time goes by and we learn about the reward statistics of each arm (while we update the Q values), we want to decrease the learning rate so that we exploit this knowledge and we get rewarded from it. However, we don't want the learning rate to go too close to 0, since we want to keep on learning meanwhile.

Looking at the plots of the four tested learning rates, the one that seems to better follow this reasoning is $\alpha = \frac{1}{1 + ln(1 + K)}$ (the third case). This makes sense, since it is the learning rate that achieves the highest average accumulated reward.

The first figure, showing the case for learning rate equal to 1, also follows this reasoning, since the four epsilon cases are less distinctive from one another (compared to using the other learning rates values) meaning that exploration and exploitation are more similar in this case. It is also the one that achieves the lowest average accumulated reward, since the exploitation results in lower rewards than in the other cases.

Epsilon and final Q-values:

Looking at the average accumulated reward for the third learning rate case (probably the best one), we can see that the greedy (E=0) and random (E=0.5) curves are the ones that achieve the lower average accumulated rewards. This is because in general, as we learned in class, the greedy (no exploration) and random (lots of exploration) cases result in lower accumulated rewards than intermediate epsilon values that lead to a better exploration/exploitation ratio.

An interesting observation is that for all three other learning rate cases, the random (E=0.5) is the one that achieves the highest accumulated reward. This clearly indicates that the learning rate value modifies the effect of the epsilon value on the accumulated reward. This is probably related to what can be seen in the final Q-values tables, where we can see that those final values are closer to the true Q-values for the third and fourth learning rate values tested. Also, a constant learning rate of 1 (first case) achieves the worse final Q-values, since it is close to a constant exploration with no exploitation.

Best case:

The maximum accumulated reward was achieved with the third learning rate ($\alpha = \frac{1}{1 + ln(1 + K)}$), and with an epsilon of 0.1. Looking at the plot for this case, it seems to have a slight positive slope, which means that if we left the algorithm run for a longer time, it would achieve an even higher accumulated reward. This makes sense since, if we look at the final Q-values in the table, we can see that they could still get closer to the true Q-values of 5 and 7.

Part B

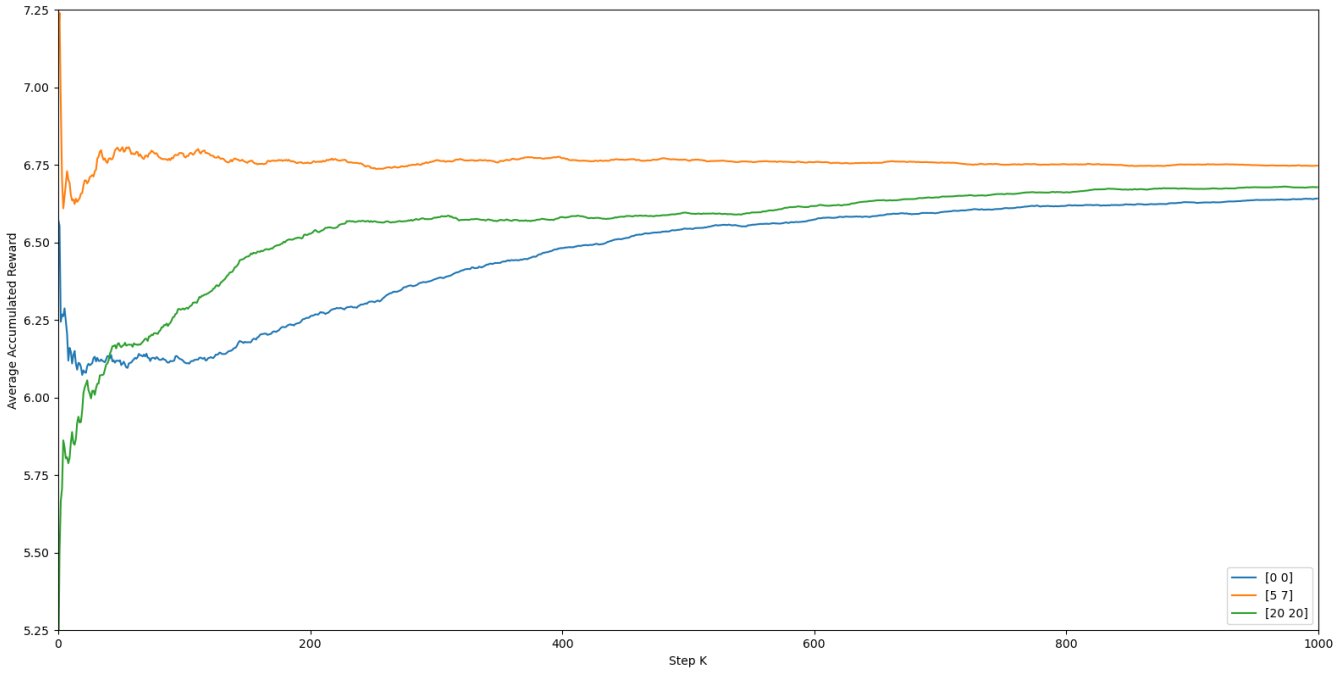

Figure 5: Average Accumulated Reward for 3 optimistic initial Q values

Epsilon-Greedy	Average of action value $Q(a^1)$ of 100 runs	True action value $Q^*(a^1)$	Average of action value $Q(a^2)$ of 100 runs	True action value $Q^*(a^2)$
Q = [0 0]	4.577	5	6.856	7
Q = [5 7]	4.6	5	6.811	7
Q = [20 20]	4.729	5	6.762	7

Table 5: Average final Q-values for $\alpha = 0.1$ and $\varepsilon = 0.1$ for different optimistic initial Q-values

Comparing the E-Greedy (with initial Q=0) and the optimistic-Greedy (with initial Q=20) curves, we can see that they follow exactly what we learned in class. This is, the optimistic one starts to increase the accumulated reward slower, but then at a certain point, it quickly surpasses the reward of the E-Greedy curve. This is because higher initial Q-values (higher than the real Q-values) lead to initially more exploration, but as time passes the exploration rate is reduced, leading to higher exploitation rewards.

We can also note that the optimistic values help get to high rewards faster, but they don't make a big difference after about 500 steps. Therefore, we can conclude that optimistic values might be useful in problems where we need to arrive to the solution in short time.

Looking at the figure, we can see that the best case is when the initial Q-values are equal to the true Q-values, which intuitively makes sense. This case has as a good estimation of the action values from the very beginning, and therefore doesn't require much exploration.

Finally, looking at the final Q-values table, there is no big difference between the three cases. Therefore, the initial Q values don't seem to be very relevant for the value of the final Q-values (at least after 1000 steps).

Part C

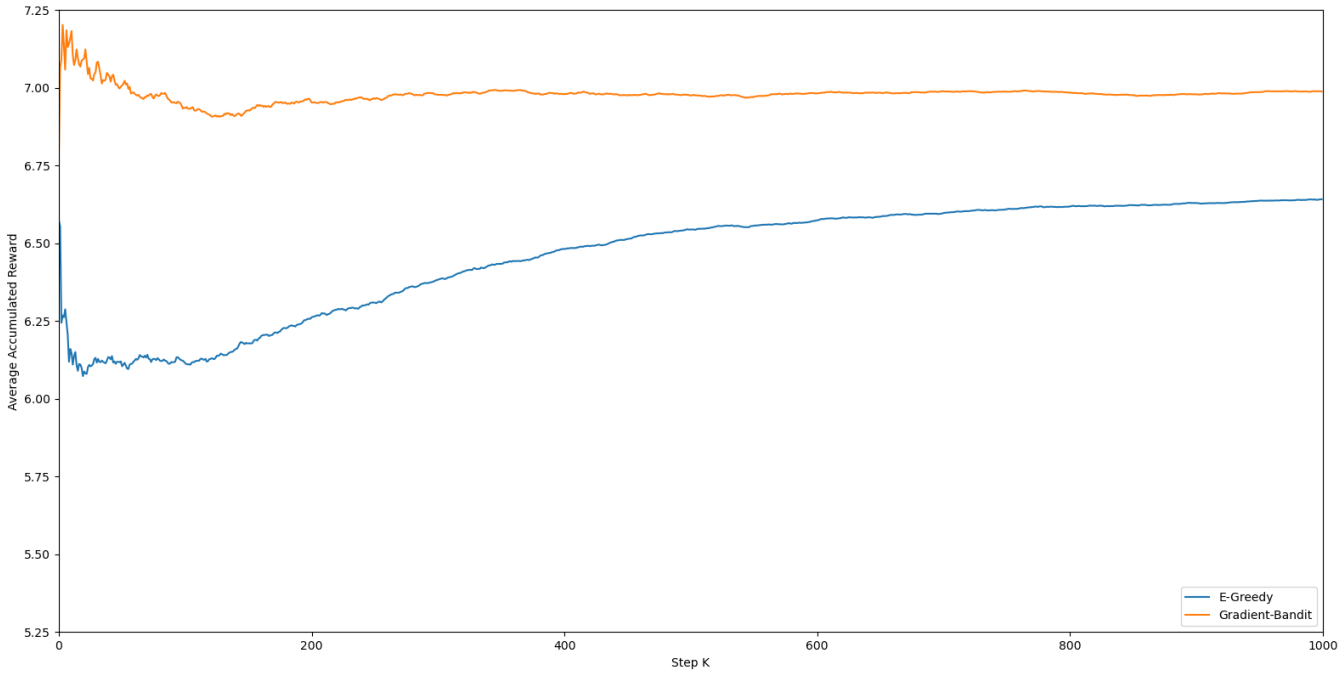

Figure 6: Comparison of Average Accumulated Reward between ε-Greedy and Gradient-Bandit policies

We can clearly see that the Gradient-Bandit case achieves higher rewards, both initially and in the long term. Also, this reward is higher than all previously analyzed cases, and it is around the expected reward value of action 2. This probably means that the algorithm quickly finds a preference for action 2, and sets a probability of selecting action 2 over action 1 which is reinforced every time action 2 provides a reward that is higher than the average reward.