Reinforcement Learning and Dynamic Optimization

May 19, 2024
Assignment 2

Georgios-Marios Tsikritzakis
ID:2020030055

1 Description

In this assignment, your job will be to use the adversarial framework of bandit algorithms we learned (i.e., experts, adversarial bandit algorithms, OCO) in order to optimize your investment in stocks. The setup of the problem is a VERY simplified version of day-to-day trading:

- During the morning of each day, you invest exactly 1 euro per day on one of K stocks.
- At the end of the day you learn the closing price for this stock, and the percentage (%) of increase or decrease of that stock. You then sell your stock, and you profit exactly that percentage of your 1 euro.
  - e.g. if you invested in stock 3 at day t, and it closed with a gain of 5%, you get back your 1 euro plus 5 cents.
  - A stock can also lose value during the day. You still must sell, and pick a stock the next morning. E.g. if stock 3 lost 10% then you lose 10 cents (but you will still invest a whole euro the next day).
- The next day you can invest again 1 euro on any of the K stocks, and so forth, until the end of the horizon.

In the .csv file you will find the day-to-day % changes in 10 real stocks I have compiled from a global stock exchange, for a duration of 2000 days.

You need to complete the following tasks:

Task 1: (Experts Setup): Assume that at the end of each day you learn the price change percentage for all K stocks (not just the one you invested in). Implement the Multiplicative Weights algorithm to maximize the amount of profit you have collected at the end of the horizon. You should show two separate plots:

- **Cumulative regret** of your algorithm, from day 1 to the last day.
- **Cumulative profits** of your algorithm (i.e., how much you've made in total by day 2, day 3, etc.)

Task 2: (Experts with Transactions Fees): Assume the previous experts setup again (i.e., full feedback) but now each stock n has a fixed transaction cost c_i , that differs between stocks.

- E.g. imagine you invested again in stock 3. And let's say that stock 3 has a transaction cost $c_3 = 2\%$, and shows a price increase of 5%. Then, at the end of the day, you will have lost 2 cents for the transaction, and gained 5 from the price increase, for an overall gain of 3 cents.
- Assume that the transactions fees are: 0.5% , 1% , 1.5%... for stocks 0 to K, respectively.

Modify your MW algorithm to maximize the accumulated profit (gains minus the transaction costs). Show two plots again, the cumulative regret and the cumulative profit over time, but each one together with the respective plot from the previous scenario with no fees. What do you observe?

Task 3: (Bandits with Transaction Fees): Assume finally that you have a bandit setup, rather than an experts one. I.e. at the end of day t you only learn the % increase/decrease of the stock you invested in that day (but not the other ones). Modify your previous algorithm to be applicable in this bandit setup and maximize the accumulated profit (minus transactions costs). Plot the cumulative regret and cumulative profits for this scenario and compare with the respective experts plots.

2 Task 1 Implementation

We implemented the Multiplicative Weights algorithm to maximize the amount of profit collected at the end of the horizon.

2.1 Modification of the MW algorithm:

First of all , we have to adapt the losses/gains bounding them in the [0,1] interval.So we've done an exploration in the stocks file to find the best % gain and the worst % loss.Then we modified a bit the algorithm.The original algorithm when updating the weights , it decreases the weight of each expert if he has a loss , else keep it as is.Our modification , decreases the weight of the expert if he has a loss , but increases his weight if he has a gain.So the l_i in our algorithm takes place in the $[-1,1]$ interval.When l_i is from -1 to 0 it means that this expert has a gain and his weight is gonna be increased.When the l_i is from 0 to 1 it means he has a loss , so decrease his weight.So we feed the algorithm with the - value of the stock file.

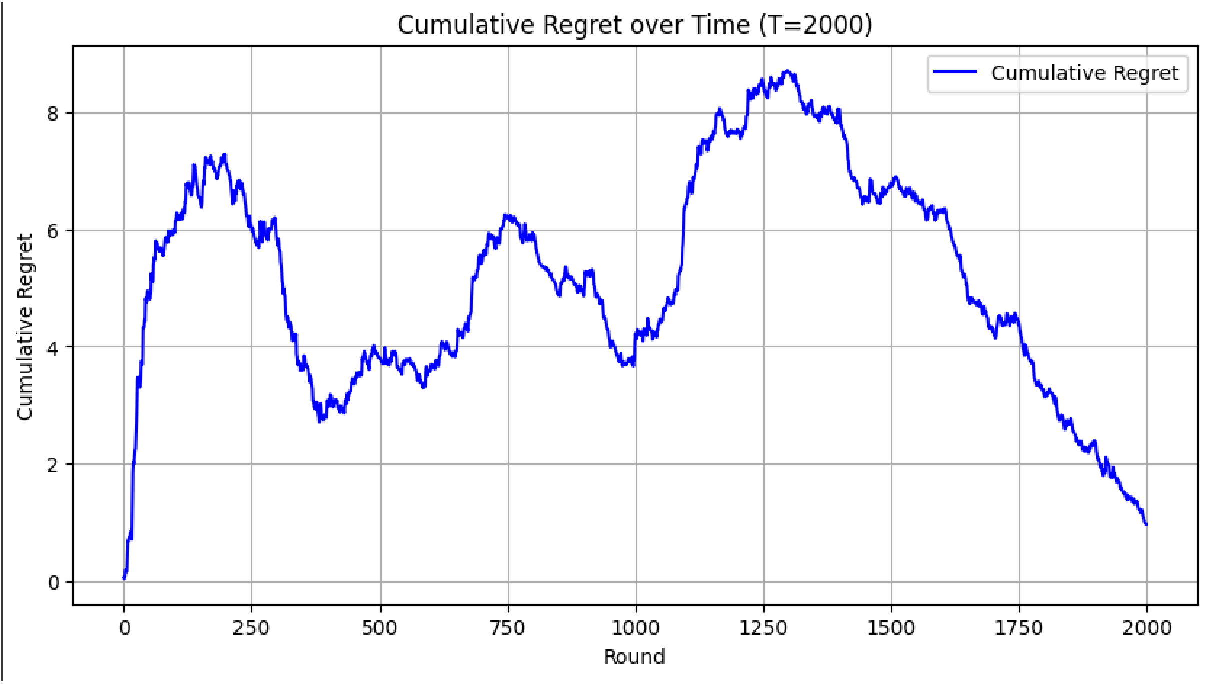

Figure 1: Cumulative Regret of Hedge algorithm with experts setup.

The regret(at any round) is defined as the $L_{\text{alg}} - L_{\text{opt}}$, where the L_{alg} is the average(expected) loss/gain at this round, and the L_{opt} is the loss/gain of the optimal expert at this round.Optimal expert-stock is the one that minimizes the losses or maximizing the gains at the end of the horizon(not in every round).So to find the optimal expert we sum all the %values of each expert , and we get the one with the maximum profit.This is the expert we compare with.

So when $L_{\text{alg}} < L_{\text{opt}}$, that means that optimal expert has more %gain or less %loss than the algorithms chosen expert , at this round , so we have a positive regret to add in the cumulative regret.

Similarly when $L_{\text{alg}} > L_{\text{opt}}$, that means that the algorithms chosen expert at this round has more profit or less loss than the optimal expert , so we have a regret to subtract from the cumulative one.

Basically the regret is defined as the suboptimality of the algorithm's chosen expert , compared to the optimal one , so Regret $=$ Reward$_{\text{opt}}$ $-$ Reward$_{\text{alg}}$ as defined in stochastic bandits.

In the plot we can see that the regret goes up and down , meaning that sometimes the algorithm is making more profit than the optimal expert and sometimes the optimal makes more profit .

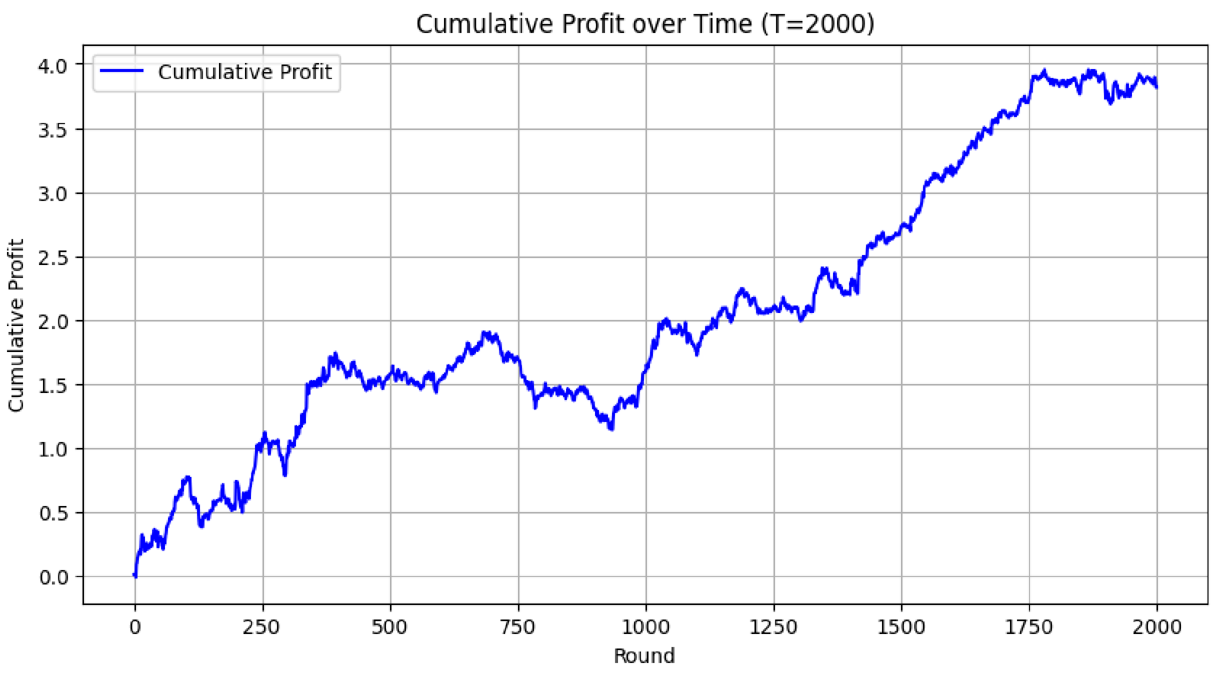

Figure 2: Cumulative Profit of Hedge algorithm with experts setup.

We can see that at the end of the horizon we can make some profit.The profit goes up and down because we have a probabillistic algorithm and an adversarial environment,so we dont know for sure that the decision we made is the optimal one.The optimal's stock profit is 4.028565203177035 and our's profit is close enough to 4.

3 Task 2 Implementation

We implemented the Multiplicative Weights algorithm of stocks(experts) with transaction fees environment ,to maximize the amount of profit(gains minus the transaction fees) you have collected at the end of the horizon. We found again the max %gain and the min %loss considering the transaction fees .We feed the algorithm with the normalized losses/gains considering the transaction fees of each expert every time.

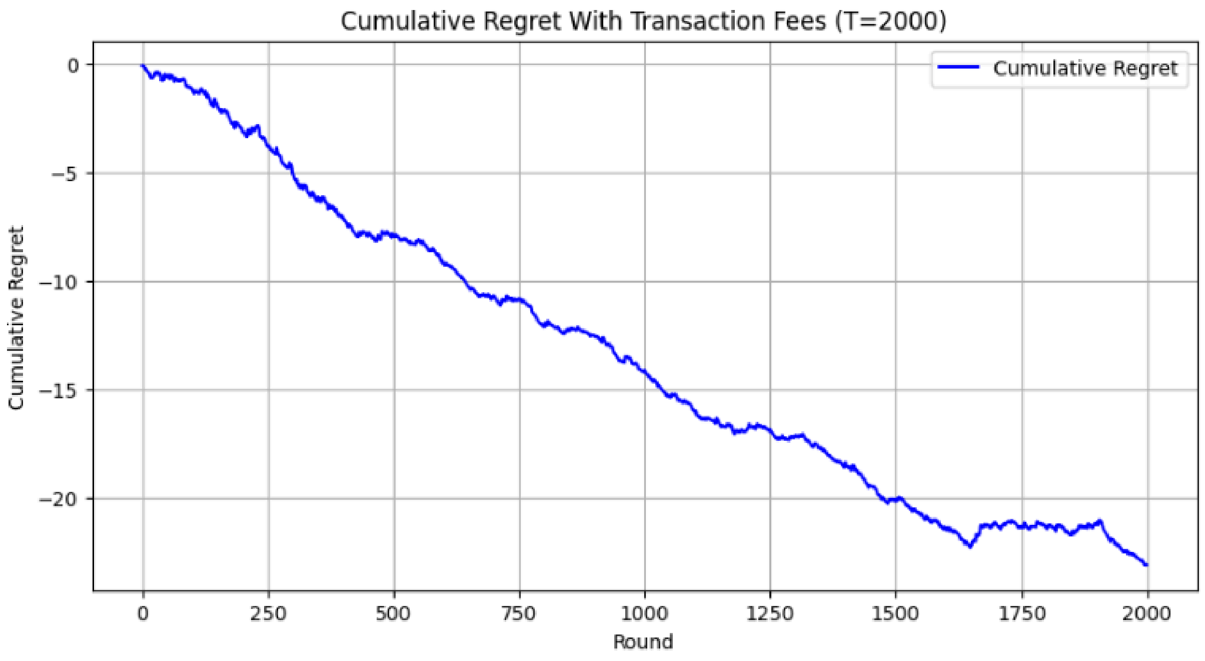

Figure 3: Cumulative Regret of Hedge algorithm with experts setup with transaction fees.

Explanation:
In the above graph we can see that our algorithm does better “profit” than the optimal expert(Because the regret is decreasing).What does it mean? That means that the algorithm chooses experts with less loss than the loss that occurs from the optimal expert.The most values of the %values in the array are negative , because the transaction fees.So whatever choice we do , we are losing money.

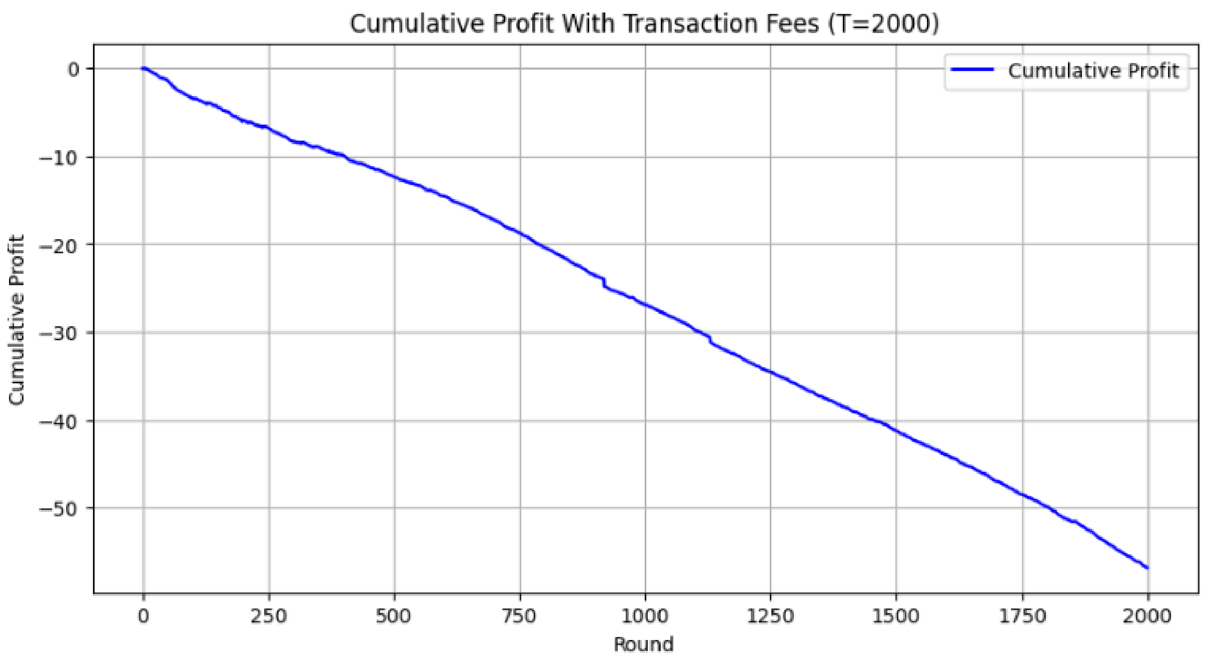

Figure 4: Cumulative Profit of Hedge algorithm with experts setup with transaction fees.

We can see that whatever the algorithm does , we are losing money and make no profit because the application of the transaction fees .

3.1 Plots compared to the previous Plots

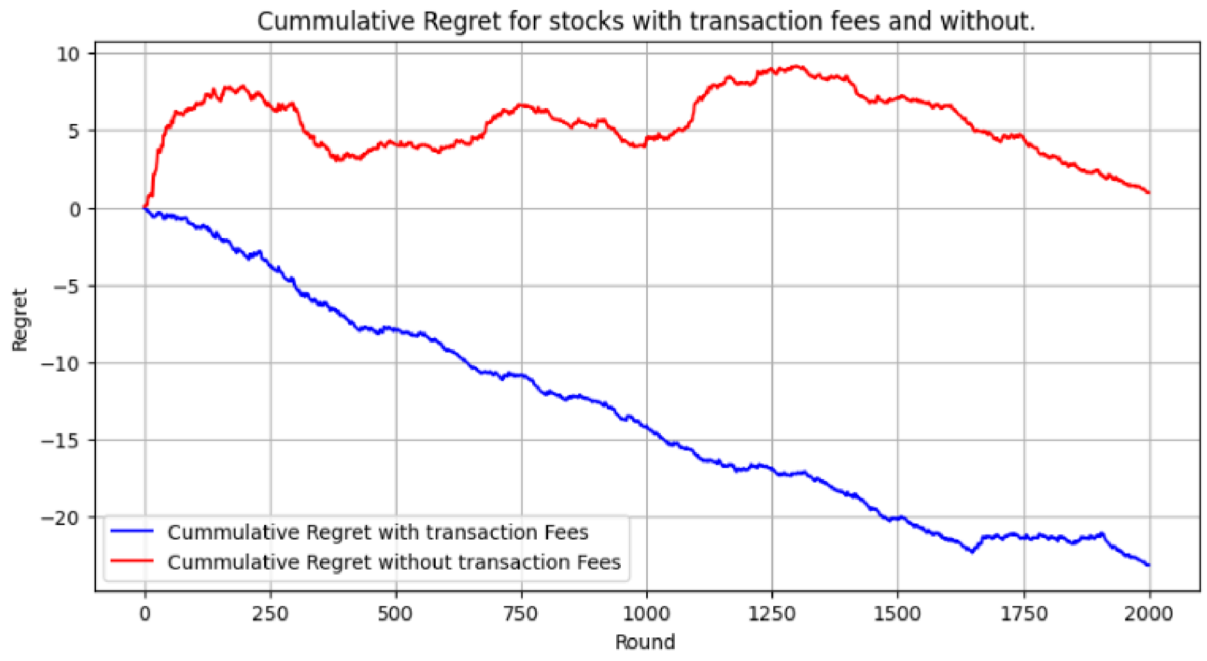

Figure 5: Cumulative Regret of Hedge algorithm with experts setup.

In both scenarios sometimes the algorithm occurs more profit / less loss (when the slope is negative) and sometimes occurs less profit/ more loss (when the slope is positive).

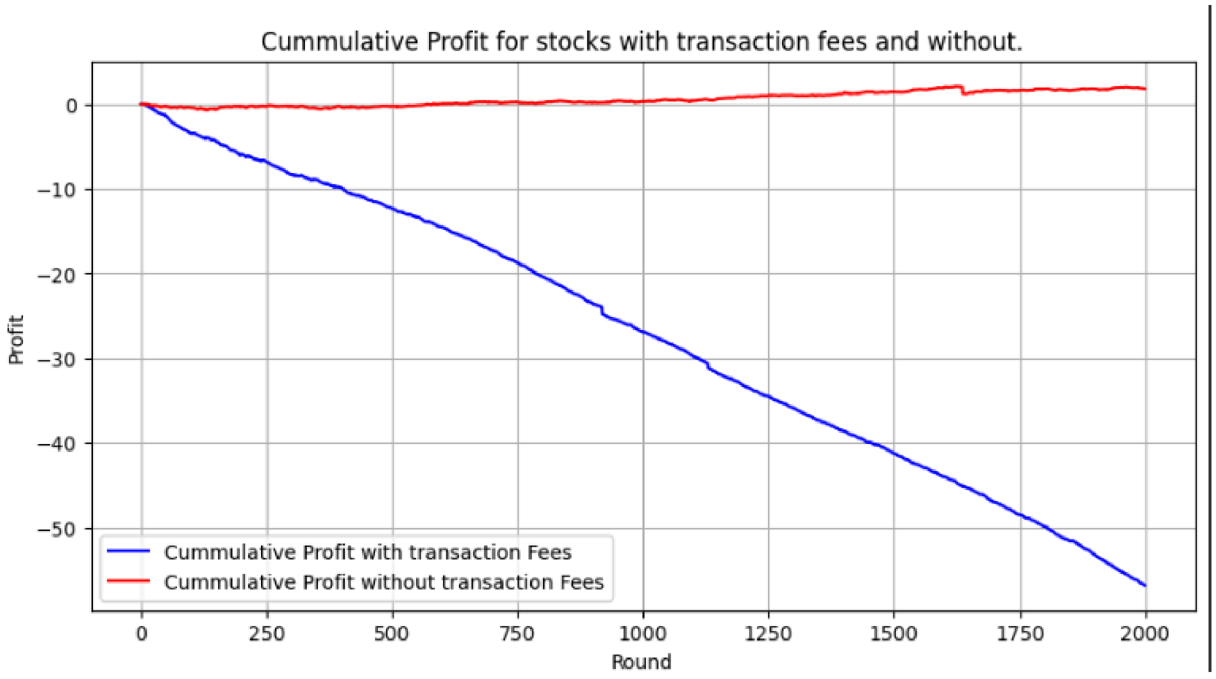

Figure 6: Cumulative Profit of Hedge algorithm with experts setup .

Here we can see that with the application of transaction fees we cannot have any profit compared to the case without fees , where we were pretty close to the optimal's expert profit.

4 Task 3 Implementation

We implemented the Multiplicative Weights algorithm of stocks with transaction fees on a bandits environment(partial knowledge of the losses , you know the loss-gain only for the stock you invested) ,to maximize the amount of profit(gains minus the transaction fees)collected at the end of the horizon.

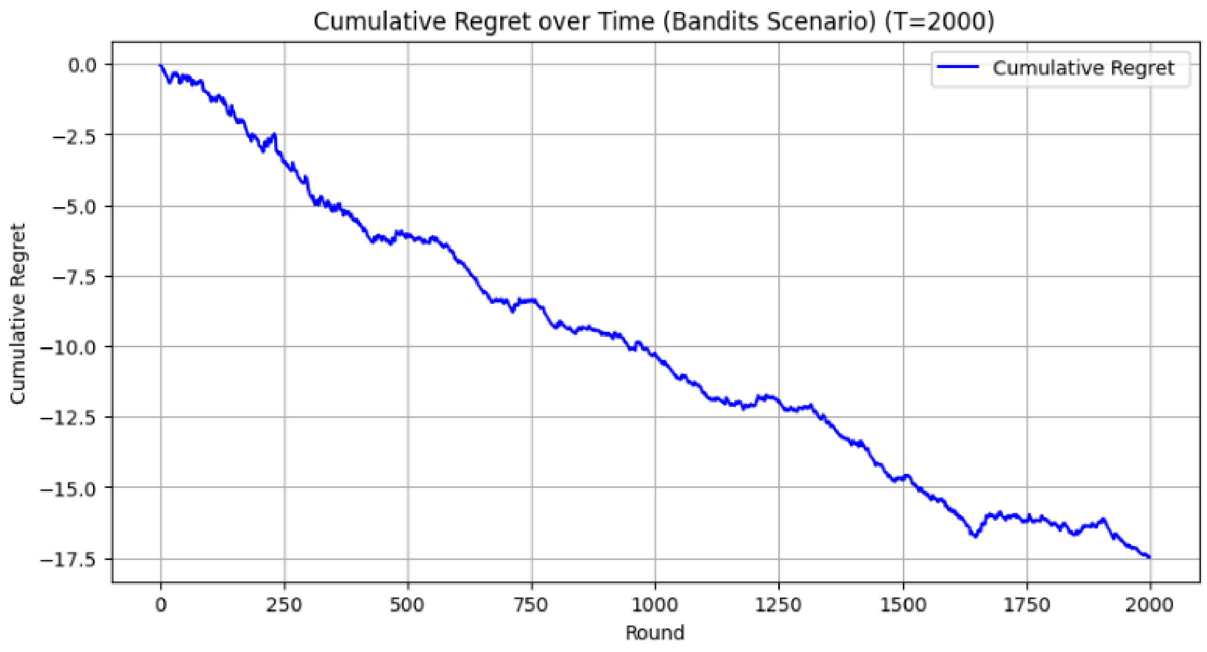

Figure 7: Cumulative Regret of Hedge algorithm with bandits setup and transaction fees.

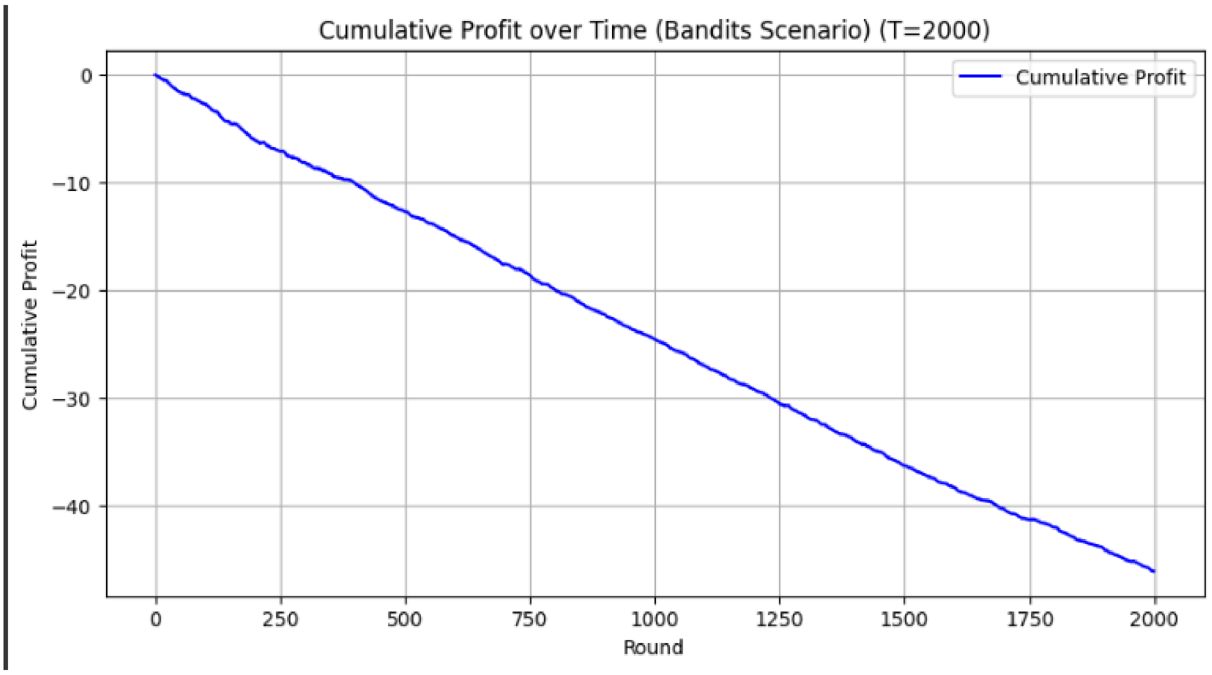

Figure 8: Cumulative Profit of Hedge algorithm with bandits setup and transaction fees.

Again we can see that due to the transaction fees we cannot make profit . Although we can see that the exploration parameter we add in the probabilities can make some difference.We have less loss in the profit compared to the experts scenario, and that's because we have partial observability.We do not change the weight of every expert every round , and so in the next round we may choose an expert which gave a loss previously(so the probability of him to be choosed in the experts scenario is less) but now occurs a profit or less loss than previous round.In an adversarial environment the partial knowledge of the losses can be more effective than an adversarial environment where we know all the losses(experts setup).