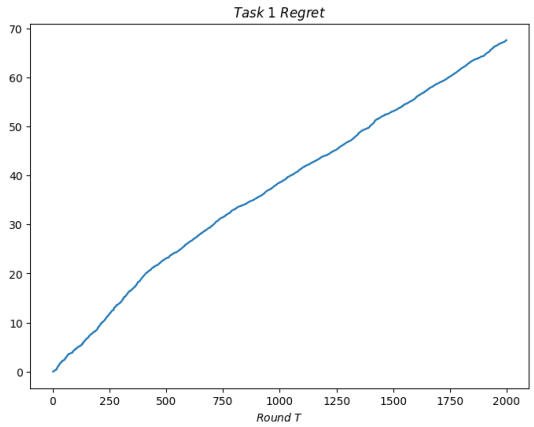
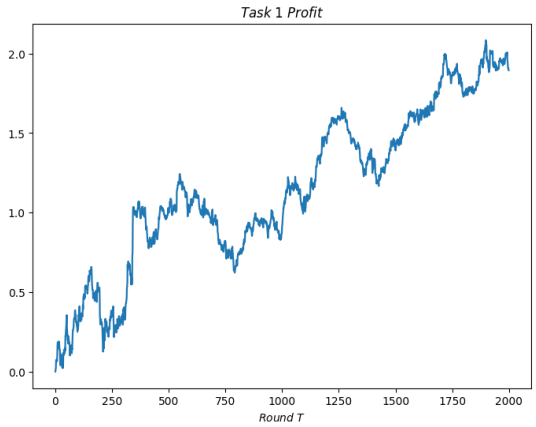
Technical University of Crete
School of Electrical and Computer Engineering
Course: **Reinforcement Learning and Dynamic Optimization**
Assignment 2
Angelopoulos Dimitris - 2020030038

In this exercise we will use the Multiplicative Weights(MW) algorithm in order to optimize our investments in some stocks. Every day t we choose to invest 1 euro in one of $k = 10$ stocks for $T = 2000$ days. At the end of the day we learn the percentage increase/decrease of the stock we invested in and calculate our profit. The goal is to maximize our profits, i.e. minimize our loss. The number of experts we set is equal to the number of stocks. The hyperparameter of the MW algorithm is $\eta = \sqrt{\frac{\ln k}{T}}$.

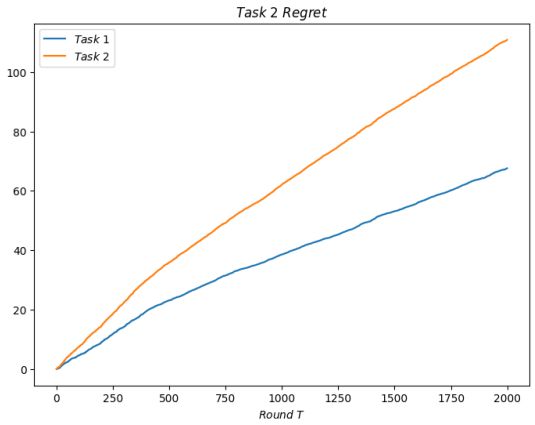
Task 1 For the first task we will assume that we learn the price change for all k stocks at the end of the day. That means that we can update the weight of every expert after calculating our profit for the day. Implementing this in code yields the following cumulative profit and regret plots.

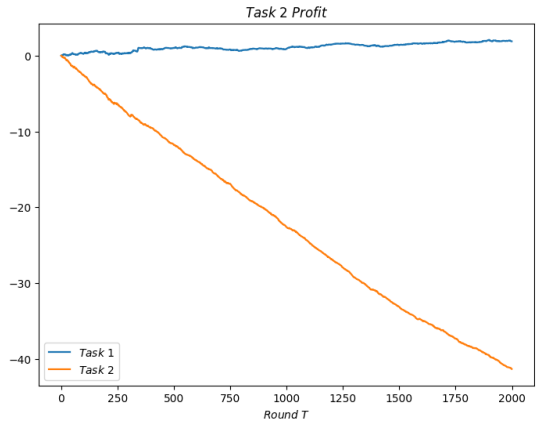




The MW algorithm can actually increase our profits.

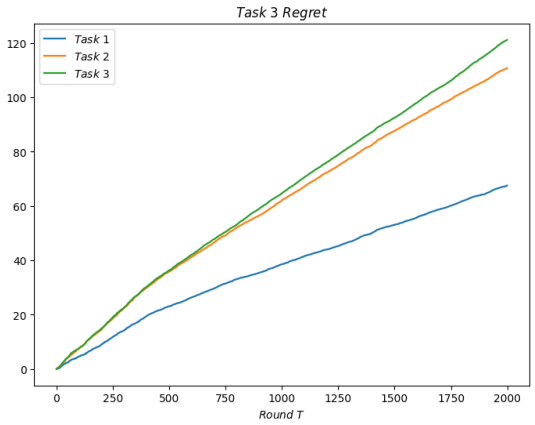
Task 2 For the second task we will implement the same algorithm with one change. This time our investment has a transaction fee. That means that we should subtract the transaction cost from our profit at the end of the day. The resulting plots are shown below. We observe that we stop making profit after adding that change. In fact we lose approximately 40 euros at the end of the time horizon. On the other hand the regret remains

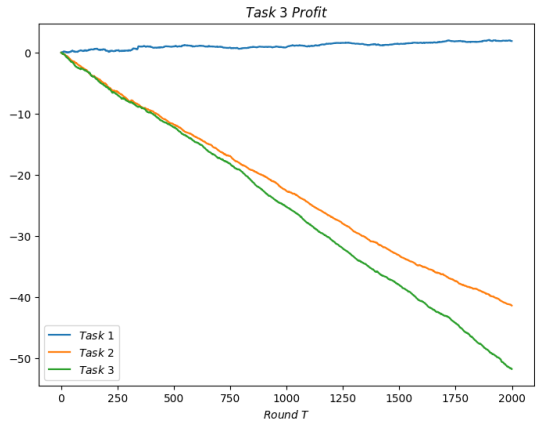




sublinear.

Task 3 Now we assume that we have a bandit setup, meaning that we only update the weight of the expert the algorithm chooses. This also means that we have balance exploration and exploitation. For this case on more hyperparameter is used, ϵ.





The bandit environment makes our algorithm have slightly worst performance than the previous task where we knew the price change of every stock. That results in a decrease of profit by 10 euros comparing to the previous but still maintaining sublinear regret, $Regret^T = O(\sqrt{Tk \ln k})$.

The Colab code for the assignment can be found in the following link.