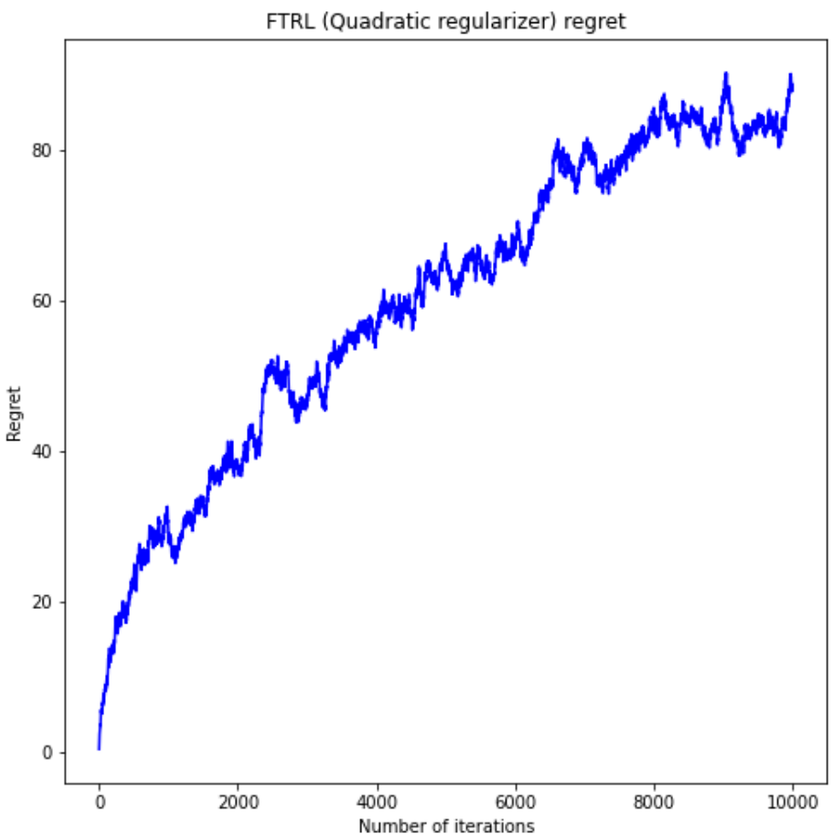
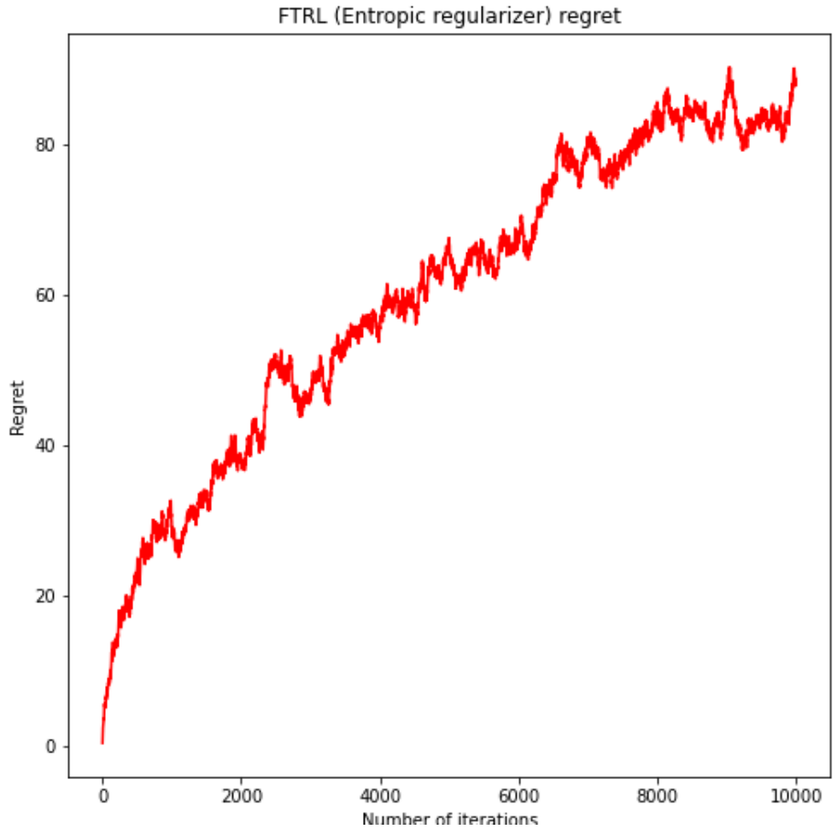
Assignment 2

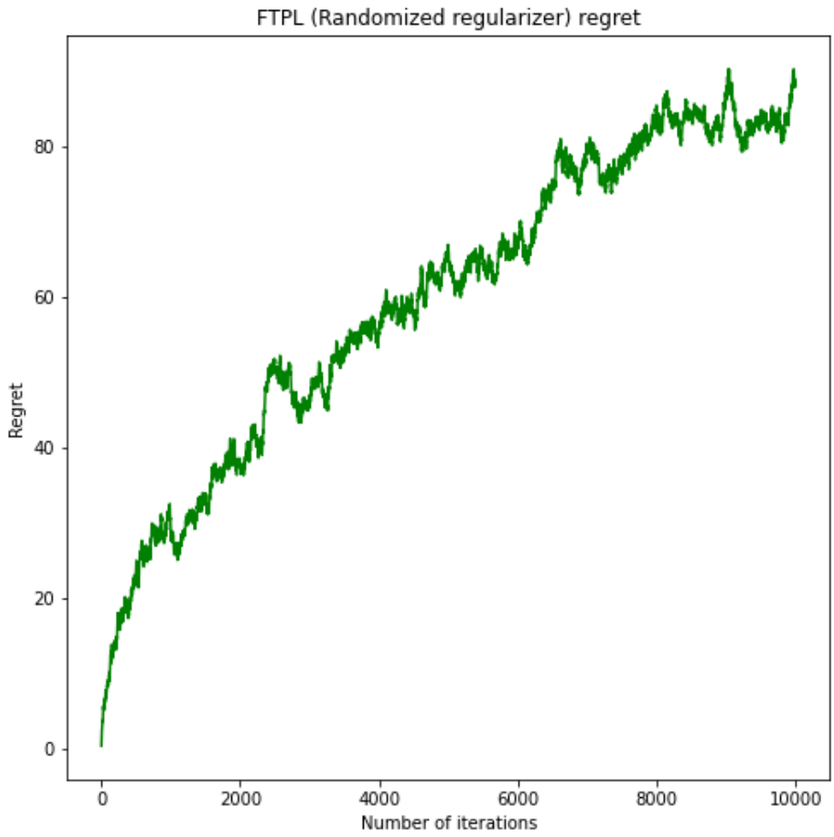
Kartik Bharadwaj CS20S020

May 10, 2021

1 Question 1







By adding random vector R to our objective function, we convert the FTL(Follow-the-leader) into FTPL (Follow-the-perturbed-leader) algorithm. The regret bound for FTPL is:

$$\mathbb{E}[\text{cost of FTPL}(\eta)] \leq \text{min-cost}_T + \eta RAT + \frac{D}{\eta}$$

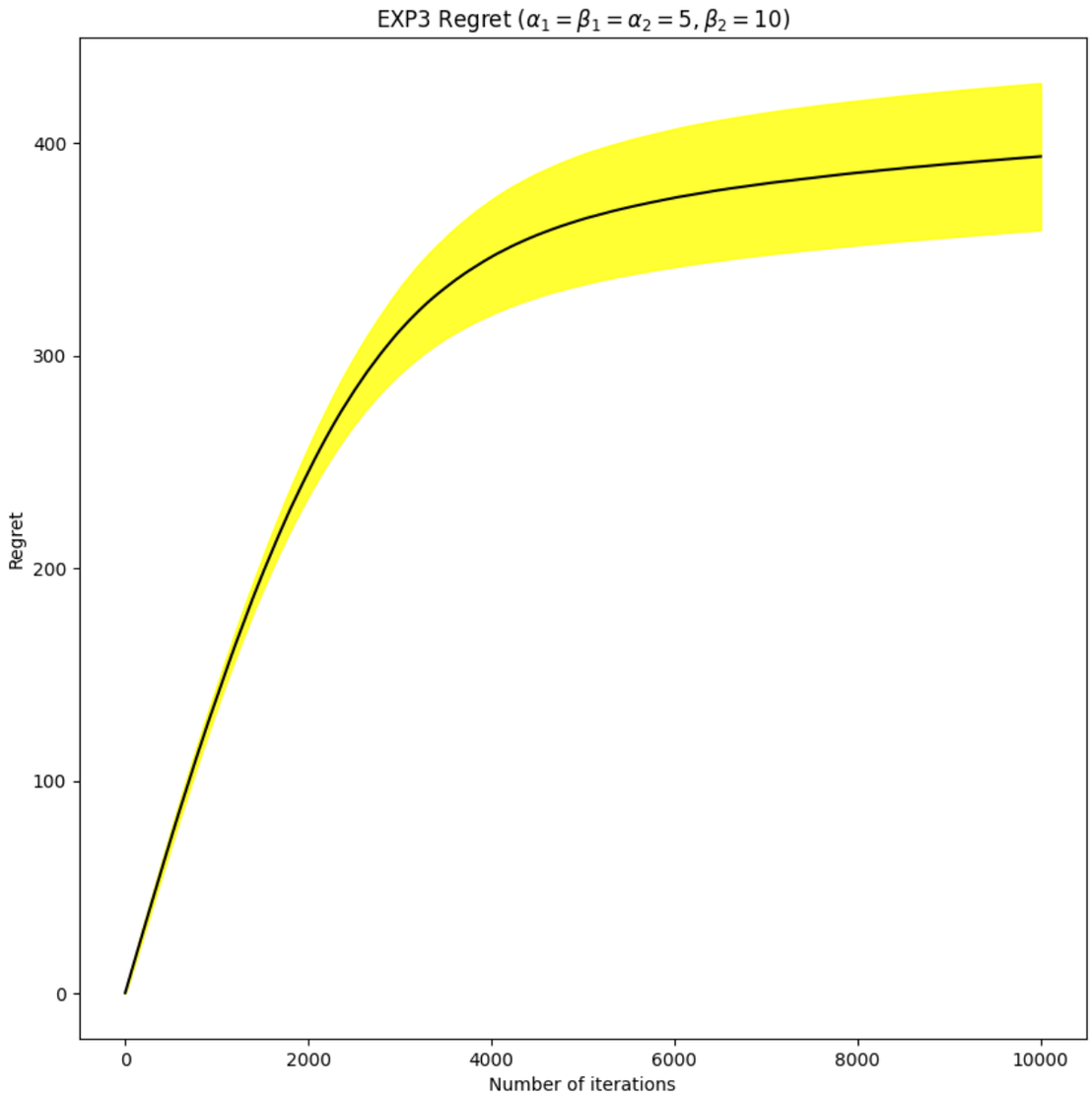
Hence, the optimal value of $\eta = \sqrt{\frac{D}{RAT}}$ where:

1. Diameter $D \geq |p - p'|_1, \ \forall \ p, p' \in \Delta^d$
2. $R \geq |p \cdot z|, \ \forall \ p \in \Delta^d, z \in R^d$
3. $A \geq |z|_1, \ \forall \ z \in R^d$

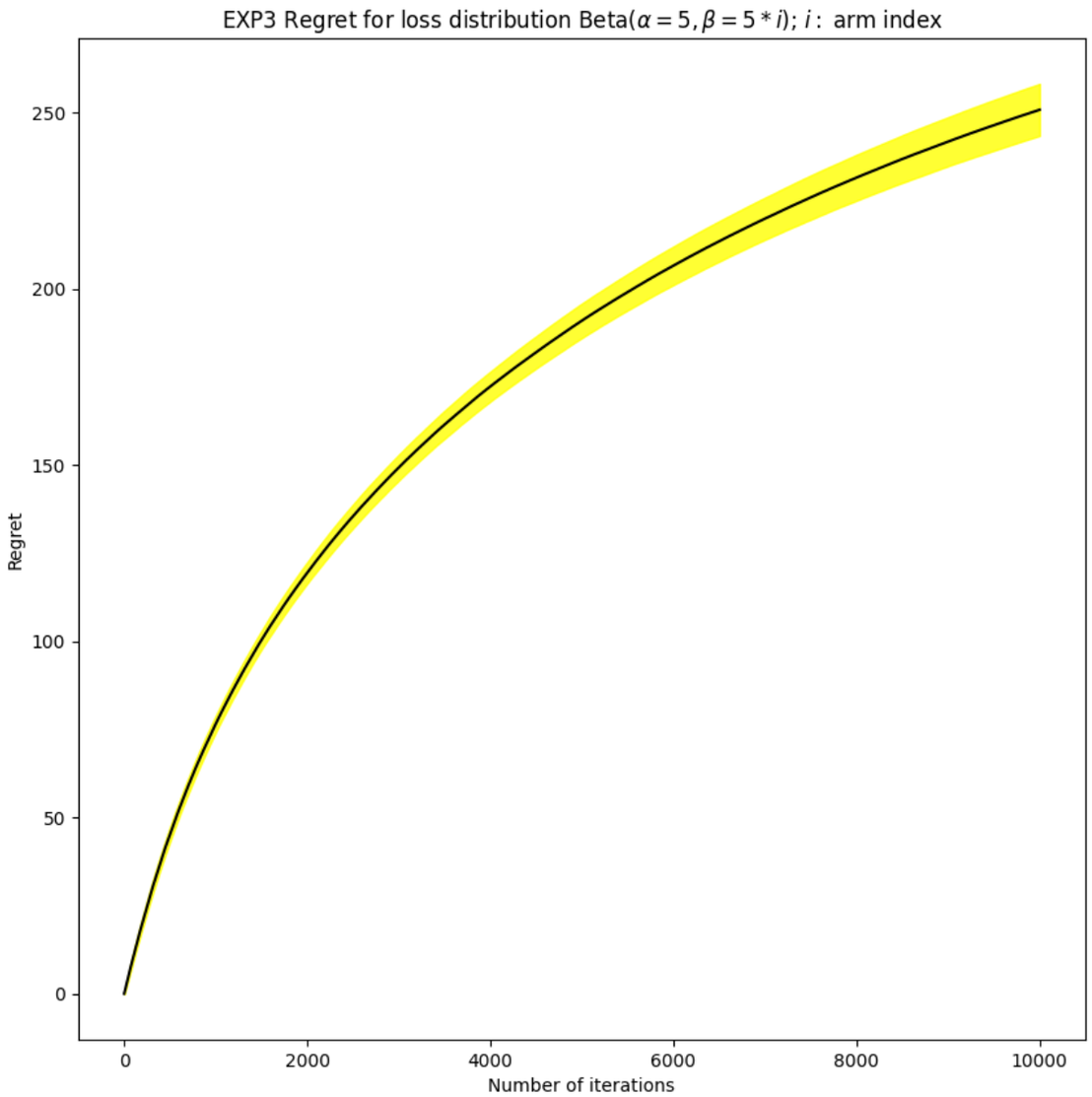
By plotting the regret bound and comparing FTPL with FTRL (Quadratic) and FTRL (Entropic), we see that they all perform very similar to each other. This means that adding randomization acts like a regularizer and that there is some inherent relationship between randomization and regularization of ML algorithms.

2 Question 2

We know that $\mathbb{E}\hat{l_i^t} = l_i^t$. Similarly, $Var(\hat{l_i^t}) = \frac{1}{p_i^t}$. This means that at any given round, if the probability of sampling an arm is very low, $Var(\hat{l_i^t})$ increases. This means that regret will increase due to higher expected loss for sub-optimal arms with low probability. This trend can be seen when we vary shape parameters.



By varying shape parameters, we vary how the expected loss of each arm affects the regret of the algorithm. For instance, the expected loss of sampling from $Beta(5, 5)$ is more than $Beta(5, 10)$. In the graph above, we sample losses from $Beta(5, 5)$ for 9 arms. Hence, when EXP3 reduces the probability across these 9 arms, variance in the fake loss for each of the 9 arms increases, thereby, adding to our regret.



On the other hand, if we sample losses from $Beta(5, 5 * i)$, where i is the index of the arm, the expected loss over each arm decreases as we increase the arm index. Hence, variance of the losses of sub-optimal arm decreases our effect on the algorithm's regret.

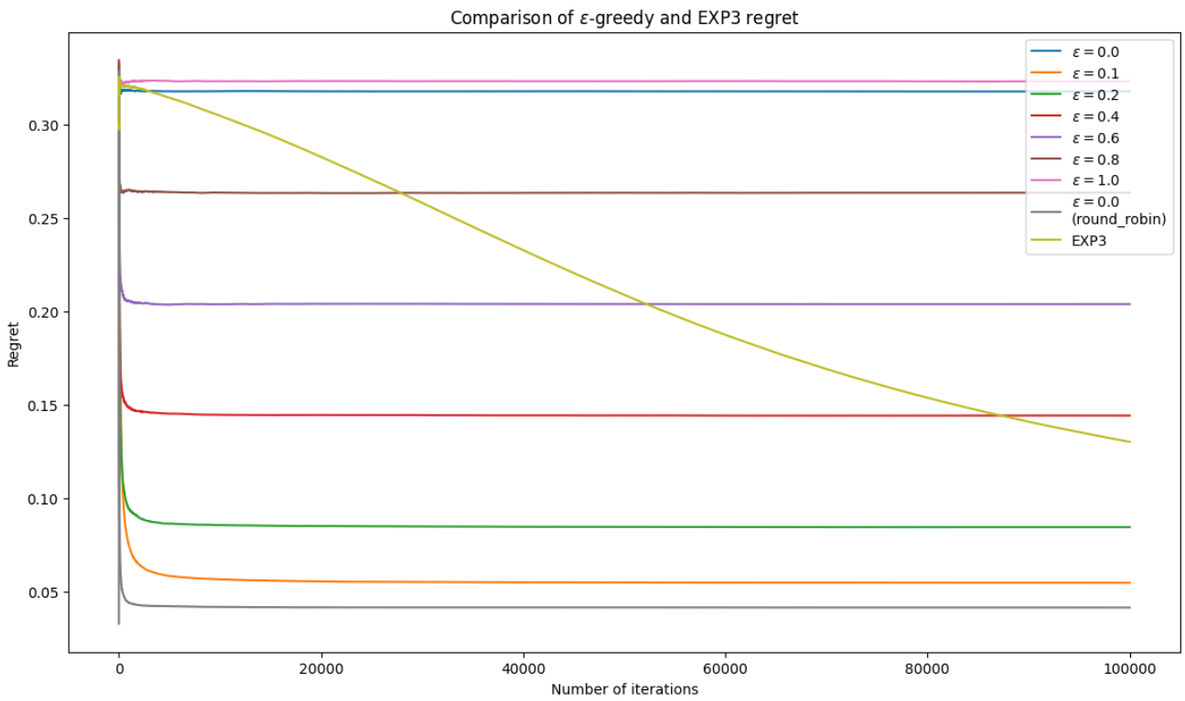
One way to mitigate the issue of variance is to introduce a constant term to p_i^t in the following manner:

$$p_i^t = (1 - \gamma)\frac{w_i^t}{\sum_i w_i^t} + \frac{\gamma}{\#\text{arms}}$$

This will ensure that there is a constant probability attached to each arm so that the value doesn't decrease to 0 and in turn increase variance of the estimator. The γ can be thought of as mixing (or biasing) sampling probabilities with uniform distribution (or $\frac{\gamma}{\#\text{arms}}$ term).

3 Question 3

Plot for required experiments:



The above experiments are run for 100k iterations for 100 times and then mean is taken.

From the graph, we can see that with $\epsilon = 1$, we perform pure exploration, thereby, giving us worst regret. On the other hand, with $\epsilon = 1$, we do pure exploitation which gives us second worst regret. This close relationship between pure exploration and pure exploitation says that none of them are good strategies to achieve lowest regret. As we decrease the value of ϵ, the regret decreases. For $\epsilon-$greedy algorithm, our most optimal (or lowest) regret is at $\epsilon = 0.1$, which means we explore 10% of the time and exploit rest of the time.

If we analyze our reward distribution, we can see that as values of β increases over each arm, the expected reward value over each arm decreases. This means arm 1 gets the most rewards. Hence, when we run round robin exploration of arms once, and then act greedily ($\epsilon = 0$), arm 1 is chosen most often in expectation, thereby, leading to zero regret. Hence, round robin algorithm gives the best regret in this scenario. A similar analogy could be considered for $\epsilon = 0.1$ as it only explores enough to find the optimal arm and then exploit it. Arms with other ϵ values explore more than what is necessary and hence suffer greater regret.

Finally, we can see that EXP3 decreases as we increase the number of time steps. This means as $T \to \infty$, EXP3 will most often choose the best optimal arm (in this case, arm 1) as it will have the highest probability of being picked. Its regret will come very close to round robin algorithm.

One issue with round robin method is that exploration is "concentrated" at the beginning of the experiment. If the reward distribution of each arm changes over time, RR might not perform well. Performing uniform exploration over time might help alleviate this issue.