Decomposing and Optimizing Regret in Classical Multi-Armed Bandit Algorithms: ETC, UCB, and Thompson Sampling

Yixuan Yao*

Brunel London School, North China University of Technology, Beijing, China

Abstract. Define and decompose the regret values of three classic algorithms, Explore-then-commit, Upper confidence bounds, and Thompson Sampling. First determine which part of the algorithm, function or equation yield the regret, then decompose the regret to find the source of the regret, usually into two parts: exploration phase and exploitation phase. For Explore-then-commit, the regret comes from fixed trial rounds in exploration phase; for UCB, the regret comes from running trial rounds to separate optimal arm from sub-optimal arms; for Thompson Sampling, the regret comes from the differences between posterior distribution and actual distribution. Later optimize the corresponding part of the function or equation to reduce the regret from different types of the total decomposed regret. Python coding will be used to construct original algorithm and optimized algorithm, including plotting the image of the cumulative regret of each algorithm. The effectiveness of the optimized algorithm will be verified through the comparison of original and optimized cumulative regret image.

1 Introduction

Multi-armed bandits is an arithmetic model used in statistic and probability. Its application is used to decide which arm or action yield the most reward or profit. Multi-armed bandit has a variety application, such as deep reinforcement or decision making in low frequency fluctuation [1, 2]. Multi-armed bandits include stochastic stationary bandits and non-stationary bandits. For stochastic stationary bandits, the reward distribution remains constant, while for non-stationary bandits, the reward and its distribution vary along time. This essay will be focused on stochastic stationary bandits, especially in the three classic algorithm of stochastic stationary bandits: ETC (explore-then-commit), UCB (upper confidence bond) and TS (Thompson sampling), and evaluate the limit and shortage in the form of cumulative regret of these algorithms, and decompose the composition of each algorithm's regret, followed by the optimization of specific part of each algorithm based on the source of its cumulative regret. The essay will evaluate the cumulative regret of each algorithm, ETC, UCB and TS, through their detailed function and give corresponding optimization strategy to reduce regret. The entire calculation will be conducted through python coding and the

* Corresponding author: 23190010202@mail.NCUT.edu.cn

outcome of regular algorithm and optimized algorithm will be illustrated through python image. By compare and contrasting the image outcome of regular algorithm and optimized algorithm, conclusion of the optimization strategy will be given and further inspirations regarding future research direction will be inferred.

2 Method

2.1 Definition of MAB algorithms: ETC, UCB and TS and their regrets

Multi-armed bandit, MAB, is a mathematical frame used to solve exploration-exploitation trade-off in sequential decision. Arm stands for optional decisions, for each arm there is a correlated reward with unknown distribution. Reward stands for the feedback from test subjects of each arm, which normally has a dynamic distribution. Regret stands for the reward difference between chosen arm and optimal arm, cumulative regret stands for the sum of the regret in total rounds, regret is considered to be the key index to measure the performance of decisions. After the first publication of MAB research in 1933, an enormous body of work has accumulated over the years, covered in several books and surveys [3] and applied MAB in several modern applications. Multi-armed bandit was designed to choose the best arm while minimizing the cumulative regret or maximizing the cumulative reward. Multi-armed bandit can be divided into two kinds: stochastic stationary bandits, in which the distribution of reward remains unchanged, and non-stationary bandits, where the reward distribution varies with time or actions or arms taken. For classic MAB algorithm, or stochastic stationary bandits, there are three representative algorithms: the explore-then-commit algorithm, ETC, the upper confidence bond algorithm, UCB, and the Thompson sampling algorithm, TS. For each algorithm, during their exploration phase to find the optimal arm, they yield both a different cumulative reward and cumulative regret. Although each algorithm's regret is composed of different parts from different sources, they are mostly related to the number of bandit arms and the variance of the rewards [4]. To reduce the cumulative regret based on the original algorithm: ETC, UCB and TS, one possible way is to decompose the cumulative regret into different component and apply modifications on each part of the equations from the algorithms and hence lead to a total reduction on the cumulative regret.

2.1.1 ETC regrets and optimization

ETC, explore-then-comment algorithm is composed of two parts: fist, in its exploration phase, the algorithm tries each arm same rounds, gather reward statistics and assume the average reward or mean reward of each arm to find the optimal arm, then apply or commit on the optimal arm in its exploitation phase. The cumulative regret of ETC algorithm can be divided into two parts: samples used in test rounds to find out which arm is the optimal arm, all the samples that were wasted on the sub-optimal arms are considered to be cost, which is regret. The other kind is when the ETC algorithm chooses the wrong arm, mistakenly take the sub-optimal arm for the best arm, such situation occurs when the trial rounds are not enough, and decides to commit on the wrong arm in exploitation phase. Before switching to the optimal arm, all sample used on the current sub-optimal arm are considered to regret. The second kind of regret happens under a possibility, which declines as the test rounds grow bigger. When having infinite trial round, the chances of choosing the wrong arm is approximately zero, but when in finite trial rounds, the arm that has the seemingly highest mean reward may not actually be the optimal arm [5]. The equation of ETC cumulative regret is as follows:

$$\mathrm{Rn} = \mathrm{m}\sum_{i=1}^{k}\Delta_i + (\mathrm{n}-\mathrm{mk})\sum_{i=1}^{k}\Delta_i\, p\left[\widehat{\mu_i}(mk) \geq max\widehat{\mu_j}(mk)\right], \quad (1)$$

where Rn is the total cumulative regret, mk is the total round number in all the rounds when optimal arm k is chosen, $\widehat{\mu_i}$is the difference of the reward between optimal arm and sub-optimal arm.

The idea of optimizing ETC algorithm is to run a basic pre-process on all the arms. Then there will be an initial evaluation of the performance of each arm, then the arms will be separated into two groups based on their performances. Two groups will be designed as high-potential and low-potential group. After the pre-process, the rest of the test samples will be distributed unevenly, where more test samples will be concentrated on the high-potential group to determine which arm is the optimal one. Thus, reducing the cost of samples wasted on the arms that reward poorly and reduce the probability of committing on the wrong arm in exploitation phase. The optimized part of the equation is as follows:

$$m_a = \begin{cases} m_{high}, & if\ \hat{\mu} > \theta \\ m_{low}, & otherwise \end{cases}, \quad (2)$$

where m_a is the remaining test sample to be distributed unevenly, m_{high} is the test samples for high-potential arms and m_{low} is the test samples for low-potential arms, θ is a threshold artificially set up to distinguish high-potential arms.

2.1.2 UCB regrets and optimization

UCB algorithm, upper confidence bounds algorithm, is an algorithm that chooses arm based on the upper confidence bounds of the reward distributions rather than choosing the precise arm. Upper confidence bound is also a standard tool for MAB algorithms. The index of the upper confidence bound is as follows:

$$\hat{\mu} + \sqrt{\frac{2log\frac{1}{\delta}}{n}}, \quad (3)$$

where $\hat{\mu}$ is mean reward, n is the total number of samples. When $\delta = \frac{1}{n^2}$, the index can be transformed into the following equation:

$$\hat{\mu} + \sqrt{\frac{2log\, n}{n}}. \quad (4)$$

In comparison with ETC algorithm, UCB algorithm has an advantage of changing the intensity of exploration without the need of knowing the gap between optimal arm reward and sub-optimal arm reward, which avoid the possibility of committing on the wrong arm during exploitation phase. In muti-arm situation, UCB has a priority of exploring high-potential arms, a more efficient strategy comparing to evenly exploring all arms strategy of ETC. To some extent, the optimization of ETC algorithm is an attempt of ETC mimicking UCB algorithm.

The cumulative regret of UCB algorithm is the total sum of rounds when sub-optimal arms are chosen times the gap between optimal arm reward and sub-optimal arm reward. The function of UCB algorithm's cumulative regret is as follows:

$$R_n = \sum_{i=1}^{k}\Delta_i E[T_i(n)], \quad (5)$$

where Δ_i is the difference between optimal arm reward and sub-optimal arm reward, $\mathrm{E}[T_i(n)]$ is the expected round when sub-optimal arms are chosen. The cumulative regret of UCB is composed of two parts: exploring too much on sub-optimal arms and not enough exploration

on optimal arm. When the horizon of test samples is too small, UCB algorithm may chose the sub-optimal arm, when the horizon is too big, the algorithm would waste too much test rounds on sub-optimal arms, both situations would contribute to the cumulative regret.

The optimization strategy for UCB algorithm is to distribute the test rounds in exploration phase based on each arm's standard deviation. The optimized function is as follows:

$$A_t = \text{argmax}\left(\widehat{\mu_a}(t) + c * \frac{\widehat{\sigma_a}(t)}{\sqrt{T_a(t)}}\right), \tag{6}$$

where $\widehat{\sigma_a}$ is the standard deviation of action a, which can be used to measure the fluctuation of reward; c is the adjustive parameter. By using such strategy, when the standard deviation of an arm ($\widehat{\sigma_a}(t)$) is big, the algorithm will tend to explore more on this arm, similarly, when $\widehat{\sigma_a}(t)$ of an arm is small, the algorithm would reduce running test rounds on this arm and therefore reduce the cumulative regret.

2.1.3 TS regrets and optimization

Unlike the other algorithm, where ETC evenly explore every arm, or UCB chooses optimal arm by examine their upper confidence of the reward, Tompson Sampling, or TS algorithm is a MAB algorithm that is based on Bayesian Learning. Bayesian Learning allows MAB algorithm to deal with one problem: How to choose the optimal arm in all admissible arms? For an arm that seems suitable for one circumstance may turn out to be sub-optimal in another situation. Bayesian Learning enables the algorithm to choose the algorithm that yield not the highest reward, but the smallest regret, by examining the prior probability distribution. This method has an extensive usage in many modern deep-learning algorithm [6]. In MAB's case, such method is used to conduct the minimax policy, which is to choose the arm that has the least regret, such policy can also be a research direction for UCB algorithm [7]. By using Bayesian Learning, TS algorithm can automatically adjust balanced exploration and commit on the arm that has the highest sample mean without the need of setting exploration parameter artificially [8]. TS algorithm can also be considered as a natural Bayesian algorithm [9].

In comparison with ETC algorithm and UCB algorithm, where ETC cumulates a high regret by exploring a fixed number on all arms and may commit on the wrong arm, Tompson Sampling can adjust itself through posterior distribution, which is more effective in multi-armed situation, and the latter may commit an excessive exploration by exploring through upper confidence bounds, where Tompson Sampling, based on probability sampling, has a higher tendency towards more potential arms and avoid invalid exploration. The function of TS algorithm's cumulative regret is as follows:

$$R_n = \sum_{i=1}^{k} \Delta_i \, E[T_i(n)]. \tag{7}$$

The function of TS algorithm's cumulative regret is homogenous to that of UCB's cumulative regret. The cumulative regret would accumulate when there are differences between posterior distribution and actual distribution, causing an excessive sampling on sub-optimal arm; the changing of environment would also lead to raising cumulative regret for the fixed posterior distribution cannot keep up with the fluctuation of mean, then lead to an increase on the number of sub-optimal arms [10].

The strategy of optimizing Tompson Sampling is to free the algorithm of posterior distribution of arms and allow it to automatically adjust exploration intensity. This goal will be achieved by adding a disturbance term that's proportional to the posterior variance, and thereby enhance exploration on high variance actions and focus on exploitation on smaller variance actions. The original TS algorithm sampling method is as follows:

$$\theta_a \sim \text{Beta}(\alpha_a, \beta_a). \tag{8}$$

The optimized sampling method is as follows:

$$\theta_a \sim \text{Beta}(\alpha_a, \beta_a) + \lambda * \sqrt{Var(\theta_a)}, \quad (9)$$

where $Var(\theta_a)$ is the variance of Beta distribution, λ is the adjustment parameter (usually λ =1). In the new function, when α_a *and* β_a are too small, i.e. data deficiency, variance $Var(\theta_a)$ will increase and the exploration term $\lambda * \sqrt{Var(\theta_a)}$ will increase, and encourage exploration; when α_a *and* β_a are big enough, i.e. data sufficiency, variance $Var(\theta_a)$ will shrink and encourage exploitation.

3 Experiment results

3.1 Dataset

The data set used in the experiment is a set that contains 10,0000 movie ratings of 9 movie genres, the algorithm aims to determine which movie genre is most popular among reviewers, genres are considered as arms, ratings are considered as rewards. The website of this dataset is as follows: *https://grouplens.org/datasets/movielens/1m/*

3.2 Original algorithm and optimized algorithm comparison.

By comparing original algorithm cumulative regret figures and optimized algorithm cumulative regret figure, the comparison shall prove the effectiveness of the optimization strategies.

3.2.1 ETC optimization and contrast

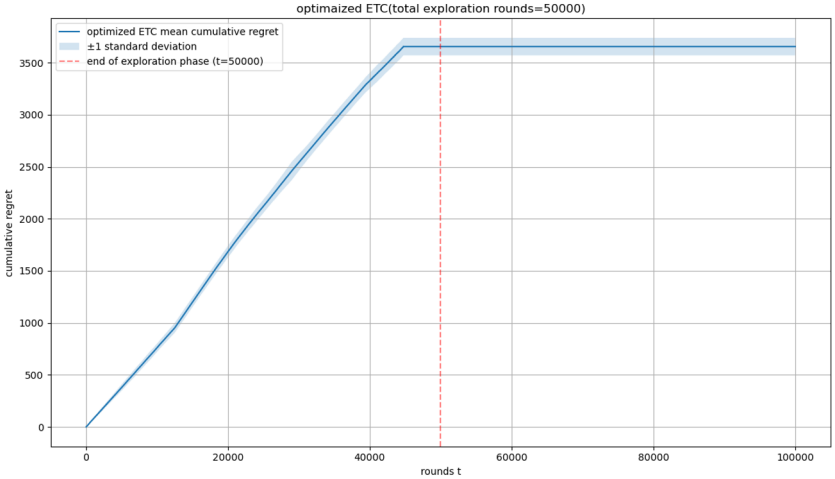

Fig.1. The outcome of optimized ETC algorithm

In this test, total trial rounds were 50,000 rounds, which is 50% of total test samples, 25% of total test rounds are used for pre-process to divide arms into low-potential and high-potential groups. As Fig.1 shows, by using the optimized ETC algorithm, cumulative regret rises all the way to the cumulative regret is approximately 3700, and the reward of optimal arm stops the cumulative from rising.

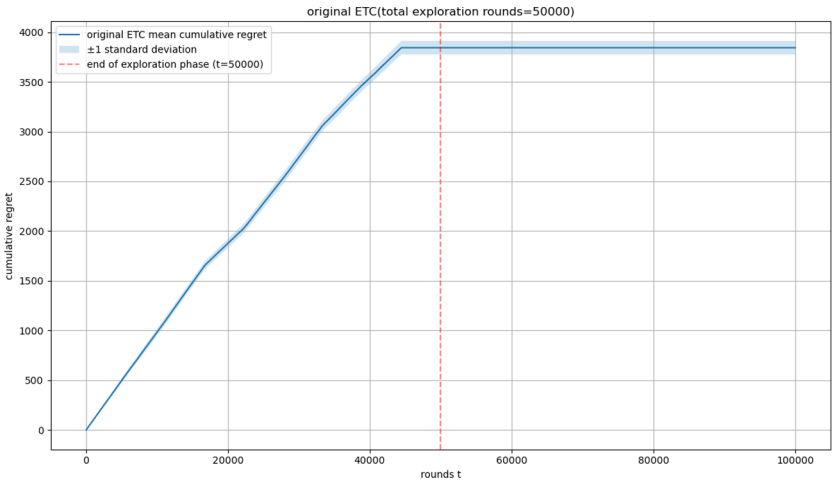

Fig.2. The outcome of original ETC algorithm

Fig.2 shows the cumulative regret of original ETC algorithm, the cumulative regret reaches the platform at approximately 3800, which is at least 100 more than optimized ETC. The comparison between optimized ETC and original ETC validates the optimization strategy.

3.2.2 UCB optimization and contrast

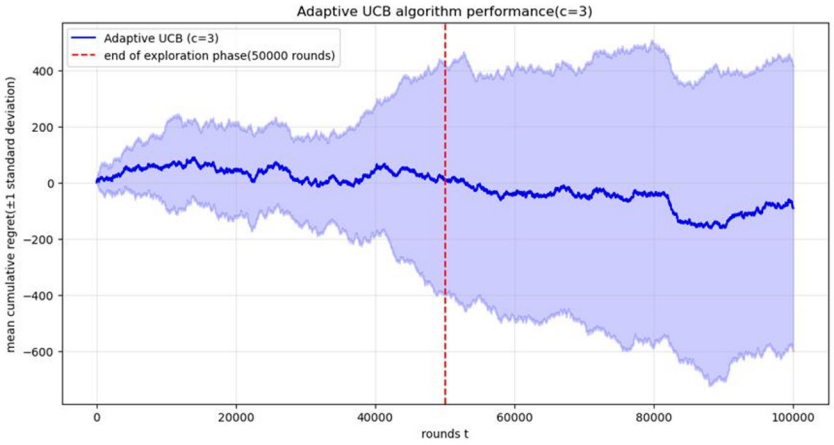

Fig.3. The outcome of optimized UCB algorithm

Fig.3 is the outcome of optimized UCB algorithm. Total sample horizon and total trial rounds remains unchanged. The cumulative regret shows a mild growth from 0 to 20000, which indicates that the algorithm chose the wrong arm at first and later switched to optimal arm, resulted in small fluctuation from 20000 to 80000 rounds, from 80000 to 100000 cumulative regret reaches stable with small fluctuation and tends to drop, this suggests the reward of the optimal arm during exploitation phase was beginning to make up for the loss in the exploration phase.

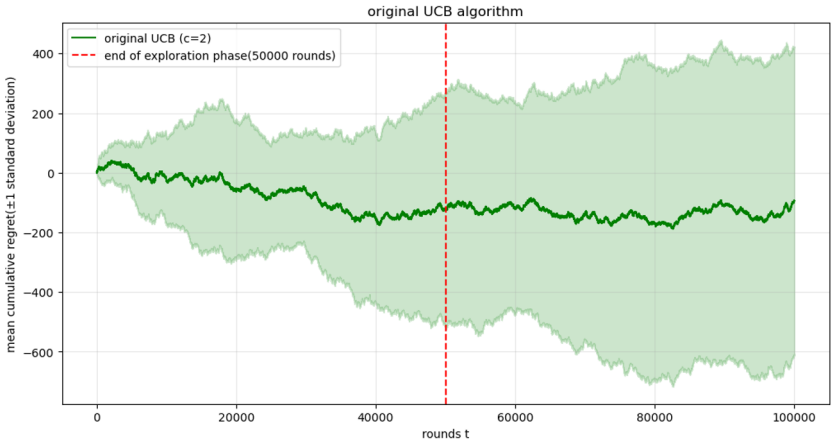

Fig.4. The outcome of original UCB algorithm

Fig.4 is the figure of the cumulative regret of original UCB algorithm. Although the curve seemingly yields an outcome of smaller cumulative regret, but at the end of this curve, figure shows a tendency of gradually rising, which implies that the original UCB algorithm may have chosen a sub-optimal arm and begin to have an increasing cumulative regret. Even though from the figure it seems that the optimized UCB algorithm has no improvements, while in fact it promised a stable and shrinking cumulative regret in comparison of a gradually growing cumulative regret of original UCB algorithm.

3.2.3 TS optimization and contrast

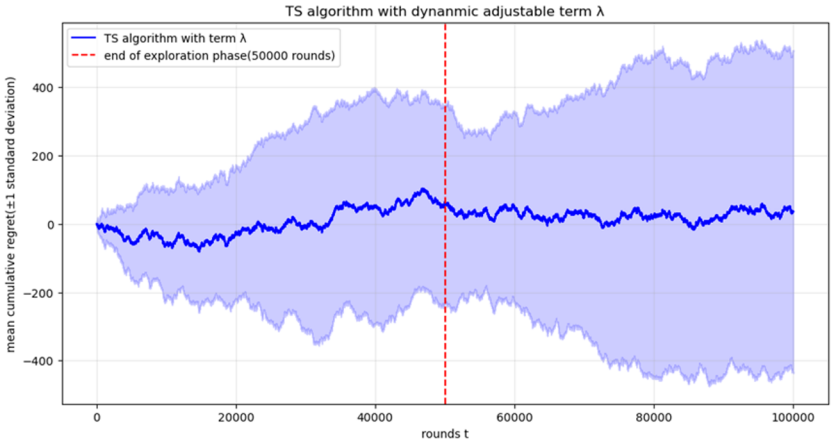

Fig.5. The outcome of optimized TS algorithm

Fig.5 is the outcome of optimized TS algorithm. Total sample horizon and total trial rounds remains unchanged. As shown in Fig.5, from 0 to 20000 rounds cumulative regret increases, which indicates that the algorithm was constantly choosing sub-optimal arms and contributing to cumulative regret; from 20000 to 50000 cumulative regret begin to drop, which suggests that the dynamic exploration term began to take place and identified the optimal arm, the curve of cumulative regret remains approximately below 0 after round 20000, this shows that the reward of optimal arm in exploitation phase outweigh the regret cumulated in exploration phase.

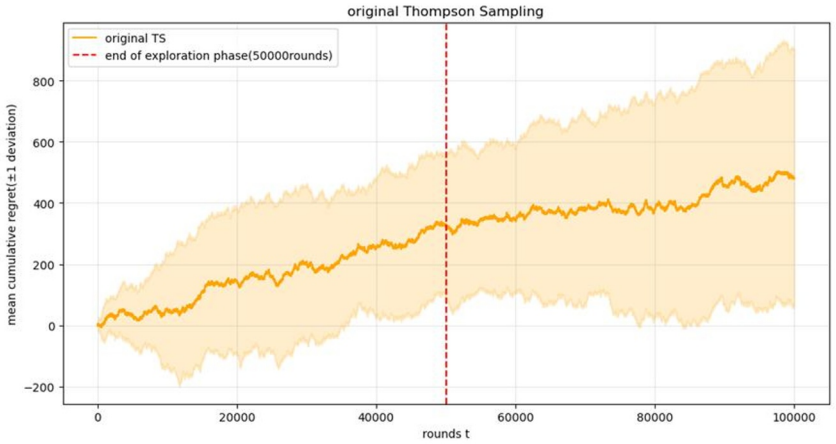

Fig.6. The outcome of original TS algorithm

Fig.6 is the figure of the cumulative regret of original TS algorithm. Fig.6 shows a fluctuating and rising curve, which indicates that the cumulative regret had been rising since exploration phase, and brought the cumulative regret to more than 400, which is much more than the outcome of optimized TS algorithm of approximately 0 cumulative regret. The comparison between optimized ETC and original ETC validates the optimization strategy.

3.3 Optimized ETC, UCB and TS algorithm comparison

By comparing original algorithm cumulative regret figures and optimized algorithm cumulative regret figure, the comparison shall prove the effectiveness of the optimization strategies.

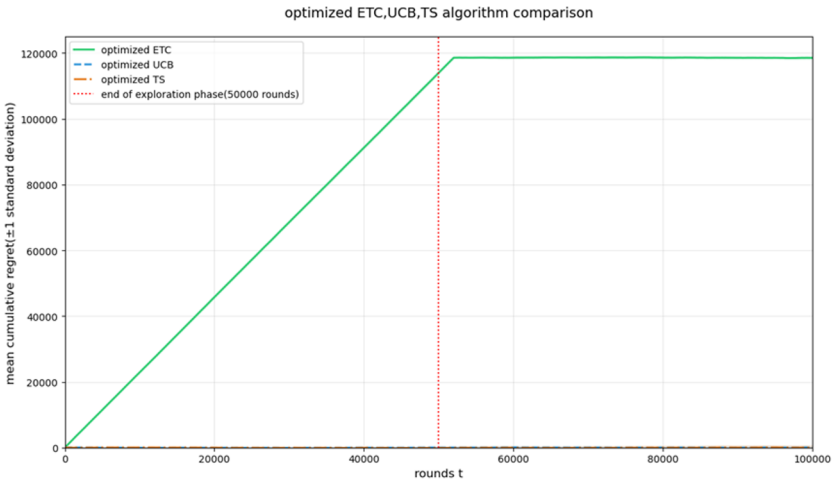

Fig.7. The cumulative regret of all three optimized algorithms

Fig.7 is the figure where the cumulative regret of all three optimized algorithm in the same form, it is clear that despite optimized ETC can reduce cumulative regret by 100, the remaining cumulative regret is still quiet high comparing to that of optimized UCB and TS, whose cumulative regrets are both close to zero.

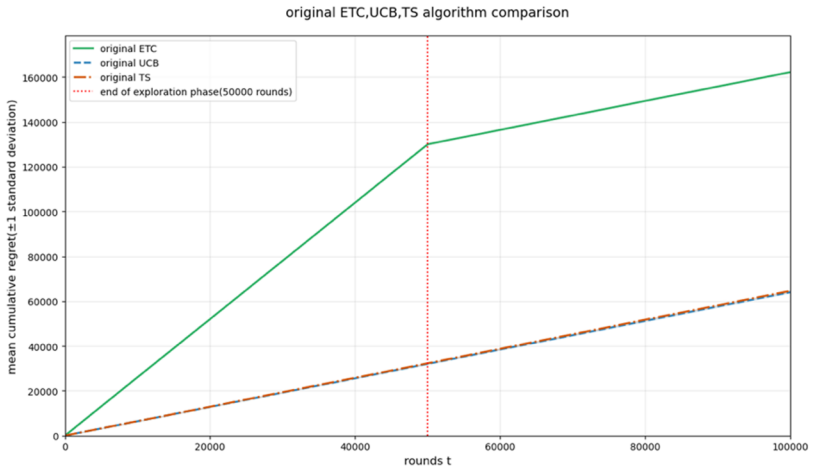

Fig.8. The cumulative regret of all three original algorithms

Fig.8 is the figure where the cumulative regret of all three original algorithm in the same form, Fig.8 shows that all three algorithm's cumulative regret are gradually growing, which indicate that all three un-optimized algorithms have chosen a sub-optimal arm and lead to an increasing regret, among them, ETC has the most rapid growing cumulative regret, suggesting its limitations and deficiencies against other two algorithms.

4 Conclusion

In summary, this paper is focused on stochastic stationary bandits, especially in the three classic algorithm of stochastic stationary bandits: ETC (explore-then-commit), UCB (upper confidence bond) and TS (Thompson sampling), and evaluate the limit and shortage in the form of cumulative regret of these algorithms, and decompose the composition of each algorithm's regret, followed by the optimization of specific part of each algorithm based on the source of its cumulative regret.

References

1. A.N. Khiabani, B. Macq, Deep reinforcement learning for the expert advice multi-armed bandit. In Proc. IEEE EUROCON 2025 – 21st Int. Conf. Smart Technol. (IEEE, 2025).
2. K. Sasaki, T. Mihana, K. Kanno, M. Naruse, A. Uchida, Experiment on decision making for multi-armed bandit problem using chaos and low frequency fluctuations in laser network. In Proc. Conf. Lasers Electro-Optics Pacific Rim (CLEO-PR) (IEEE, 2022).
3. A. Slivkins, Book announcement: Introduction to multi-armed bandits. ACM SIGecom Exchanges 18, 28–30 (2020).
4. V. Kuleshov, D. Precup, Algorithms for multi-armed bandit problems. CoRR abs/1402.6028 (2014).
5. A. Yekkehkhany, R. Nagi, Risk-averse equilibria for vehicle navigation in stochastic congestion games. IEEE Trans. Intell. Transp. Syst. 23(10), 18719–18735 (2022).
6. M.E. Khan, H. Rue, The Bayesian learning rule. J. Mach. Learn. Res. 24, 46 (2023).

7. J.Y. Audibert, S. Bubeck, Minimax policies for adversarial and stochastic bandits. In Proc. Conf. Learn. Theory (COLT), Montreal, Canada, June 18–21, pp. 217–226 (2009).
8. S. Bubeck, N. Cesa-Bianchi, Regret analysis of stochastic and nonstochastic multi-armed bandit problems. Found. Trends Mach. Learn. 5, 1–122 (2012).
9. S. Agrawal, N. Goyal, Analysis of Thompson sampling for the multi-armed bandit problem. arXiv:1111.1797 (2011).
10. P. Auer, Using confidence bounds for exploitation-exploration trade-offs. J. Mach. Learn. Res. 3, 397–422 (2002).