Incentivized Exploration in Non-stationary Stochastic Bandits

by

Sourav Chakraborty

B.E., Birla Institute of Technology, Mesra, 2016

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Master of Science

Department of Computer Science

2022

Committee Members:

Lijun Chen, Chair

Rafael M. Frongillo

Bo Waggoner

Chakraborty, Sourav (M.S., Computer Science)

Incentivized Exploration in Non-stationary Stochastic Bandits

Thesis directed by Assistant Professor Lijun Chen

We study the incentivized exploration for the multi-armed bandit (MAB) problem with non-stationary reward distributions, where the players receive compensation for exploring arms other than the greedy choice and may provide a biased feedback on reward. We analyze the impact of the drifted reward feedback on two instances of non-stationary MAB environments: Piecewise-Stationary and Continuously-Changing. We show that our algorithms for both the environments achieve sub-linear regret and compensation under drifted reward, and are therefore effective in incentivizing exploration. Experimental results with synthetic data are provided to complement the theoretical analysis.

Dedication

To the memory of my father, *Kanai Lal Chakraborty*.

Acknowledgements

This work would not have been possible without my thesis advisor, Dr. Lijun Chen. I want to thank him for showing faith in me and encouraging me to be in-dependant and pursue this work. I want to thank Dr. Zhiyuan Liu who had provided me with initial context about the problem and for that insightful discussions we had regarding this topic. I would also want to thank Dr. Bo Waggoner for introducing me to the concept of online learning and bandits through his graduate classes, and Dr. Raf Frongillo for steering me into the right direction and introducing me to Dr. Chen. Lastly, I want to thank my family and friends for supporting me during the times of turmoil in my personal life.

Contents

Chapter

List of Tables

Table

List of Figures

Figure

Chapter 1

Introduction

1.1 Problem Introduction

The multi-armed bandit (MAB) problem (Figure 1.1) is a well-studied model for sequential decision making under uncertainty, with diverse applications in, e.g., clinical trials [16, 4, 34], financial portfolio design [7], recommendation systems [6, 26], search engine systems [35] and cognitive radio networks [14]. In the traditional MAB model, a decision-maker selects an arm to pull at each time step and receives a certain reward, and their objective is to maximize the long term accumulated reward. In this setup, the decision-maker (principal) and the player (agent) who pulls the arm are assumed to be the same entity who tries to achieve a good balance between exploitation and exploration. This, however, may not always be the case in the real world. There are many scenarios where the principal and the player are different entities with different interests, and the agent may select the best performing arm in face of uncertain reward (i.e., exploitation only). Taking the example of the Amazon product recommendation system: Amazon (principal) wants the customers (agents) to try out different products (arms) to identify the best product (exploration), while the customers are heavily influenced by the current ratings and reviews of the products and behave myopically, i.e., select the currently highest rated product (exploitation).

Such a greedy selection or myopic behaviour of the agent can lead to significantly degraded performance due to inadequate exploration, as shown in [8, 39].

Incentivized exploration (Figure 1.4) has been proposed to handle the greedy/myopic behaviour of the agent: The principal provides compensation to the agent so that the agent will

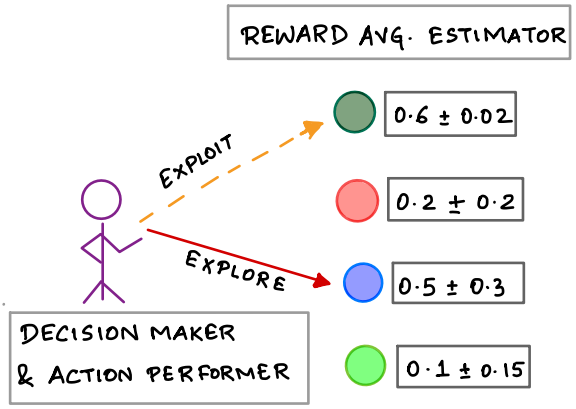

Figure 1.1: Multi-armed Bandit Problem

select the arms for effective exploration; see, e.g., [13, 31, 41, 20]. The goal of the principal is to maximize the cumulative reward while minimizing the compensation to the agents.

Early work on incentivized MAB models [20, 41, 19, 18, 28] assumed that the agents provide unbiased feedback or reward, independent of compensation received. This assumption does not always hold in the real world as shown by experimental studies in [33, 11]. These experiments show that the agents are inclined to give higher evaluations or rewards with incentives (such as discounts, coupons or gift cards in the case of Amazon). The compensation might even be the primary driver of customer satisfaction [33, 17]. This drift in reward feedback may have a negative impact on the exploration-exploitation tradeoff, as a suboptimal arm can be mistakenly identified as the optimal one because of the drifted rewards. In [29], the authors investigated the impact of the drifted feedback in the incentivized MAB problem and showed that their model for incentivized exploration based on upper confidence bound (UCB), ε-greedy or Thompson Sampling achieves optimal $O(\log T)$ regret and compensation.

A stationary bandit setting (Figure 1.2) is assumed in [29], where the reward distribution of the various arms does not change with time. In this thesis, we consider the challenging setting of non-stationary bandits (Figure 1.3) where the reward distributions change over time.

Considering the Amazon example: In the stationary setting, a product (arm) will have the

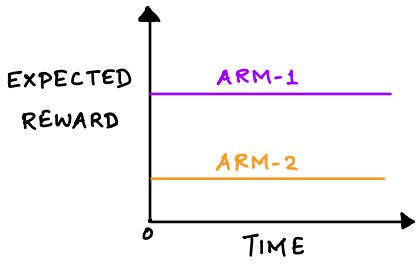

Figure 1.2: Stationary Bandit Problem

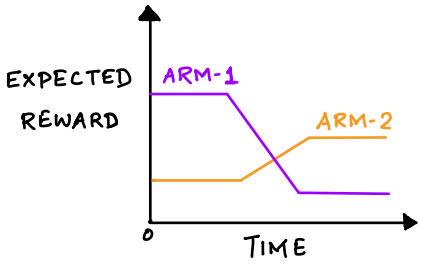

Figure 1.3: Non-stationary Bandit Problem

same value to Amazon (principal) throughout the whole time horizon, say, over a year; while in the non-stationary setting we can model a situation where some product becomes more (or less) popular because of the time of the year, (e.g., summer or a holiday season) and promises to generate higher revenue for Amazon. To take a concrete example, winter boots will be more popular in the winter and Christmas decoration items will be during the holiday season of Christmas.

Specifically, in this thesis, we study the impact of drifted reward feedback on the non-stationary incentivized MAB problem. We consider a general incentivized exploration algorithm, inspired by [29], where the agent receives compensation from the principal that is equal to the difference in estimated mean rewards between the principal's choice and the greedy choice and provides biased feedback which is equal to the sum of the true reward of an arm and a drift term that is a non-decreasing function of the compensation received for pulling the arm, but with changing reward distributions. We consider two non-stationarity models and study the robustness of the proposed algorithms in terms of regret and compensation. The first model assumes a piecewise-stationary environment (Section 3.2) where the rewards of various arms change abruptly at certain breakpoints. We show that by employing discounted UCB (D-UCB) [22] and sliding window UCB (SW-UCB) [15] as the principal's arm selection procedure, we can achieve $\tilde{O}(\sqrt{T})$ regret and compensation for D-UCB and $\tilde{O}(\sqrt{T})$ regret and $\tilde{O}(T^{1/4})$ compensation for SW-UCB. The second model considers a continuously-changing environment (Section 3.3) where the rewards can change continuously within a variation budget. We show that by employing the restarting mechanism proposed by [5] on policies UCB1, ε-greedy and Thompson Sampling as the principal's arm selection policy, we can

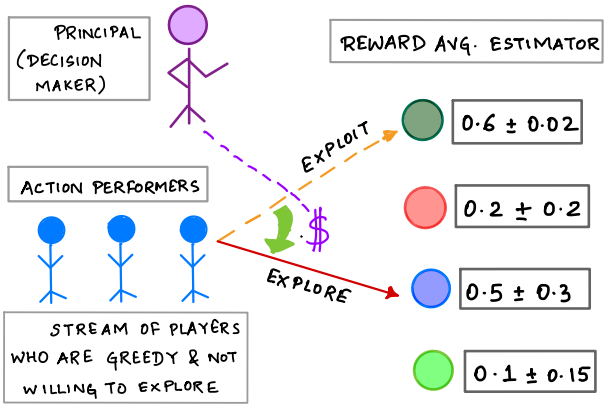

Figure 1.4: Incentivized Multi-armed Bandit Problem

achieve the optimal regret for all the policies and the $\tilde{O}(T^{2/3})$ compensation for UCB1 and $\tilde{O}(T^{1/3})$ compensation for ε-Greedy and Thompson Sampling. As all the regret and compensation bounds are sub-linear, we conclude that the proposed algorithms are effective in incentivizing exploration in non-stationary bandit environments.

1.2 Thesis Outline

This chapter is intended to give an overview of the problem to solve. The thesis is structured in the following way, starting from the next chapter:

- Chapter 2: Background and Related Work - This chapter contains the conceptual background about multi-armed bandits and some recent work related to non-stationary bandits and incentivized exploration.

- Chapter 3: Problem Formulation - This chapter formalizes the above-mentioned problem into mathematical and algorithmic frameworks.

- Chapter 4 & 5 : Piecewise-Stationary / Continuously-Changing Environments - Both these chapters talk about the non-stationary bandit environments on which the algorithms and results are based. They contain the details of the algorithms, results, their proofs and the simulation results.

- Chapter 6 : Conclusion and Future Work - This chapter concludes and summarizes the results and mentions some future directions of research.

Chapter 2

Background and Related Work

2.1 Multi-armed Stochastic Bandits

The Multi-armed bandit (MAB) problem is a classic setting with an agent who faces a choice between competing arms (i.e actions) with unknown rewards. In a single play, the agent picks an arm and receives a reward from the environment. The agent has to figure out the rewards of the arms with multiple plays and maximize the accumulated rewards.

At any given time, the agent can pick an arm that they believe to be the best one, based on the estimates so far (i.e exploitation) or try picking some alternative arm for potential future benefit (i.e exploration). This constant tussle that the agent has to face at every timestep is known as the exploration-exploitation dilemma or trade-off.

Any general MAB algorithm would follow the following protocol.

Algorithm 1: Multi-armed bandit protocol with K arms.

```
Input Parameters: K ∈ 𝒦 arms, T rounds
for each round t ∈ [1, T] do
    Algorithm picks arm I_t ∈ 𝒦
    Algorithm observes reward X_{I_t} ∈ [0, 1]
end
```

The stationary setting is the variant of the MAB problem with K arms. The reward of each arm $i \in \mathcal{K}$, where $\mathcal{K} = \{k\}_{k=1}^{K}$ follows an unknown distribution $\mathcal{D}(i)$ with support $[0, 1]$ and mean $\mu_i = \mathbb{E}[X_i]$, which does not change with time.

At each time step $t \in [1, T]$, where T is the time horizon, a player will pull one arm $I_t \in \mathcal{K}$

and receive a reward $X_t(I_t) \sim \mathcal{D}(I_t)$, which is fed back to the MAB algorithm. In this case, $N_t(i) = \sum_{\tau=1}^{t} \mathbb{1}_{\{I_\tau = i\}}$ denote the number of times the arm i was played till time t and $\hat{\mu}_t(i) = (1/N_t(i)) \sum_{\tau=1}^{t} X_t(i) \mathbb{1}_{\{I_\tau = i\}}$ be the corresponding empirical average reward.

To quantify the measure of goodness of the algorithm, we define a notion of a regret. It is defined as the expected difference between the maximum possible total rewards and the rewards accumulated by the algorithm. The maximum total reward is the total reward we could have accumulated had we known the optimal arm from the beginning and played it at each round. We denote the expected reward for that optimal arm to be μ^*. We define the expected regret as shown below in equation 2.1.

$$\mathbb{E}\left[R_T\right] = \mu^* \cdot T - \mathbb{E}\left[\sum_{t=1}^{T} X_{I_t}\right] = \mu^* \cdot T - \sum_{t=1}^{T} \mu_t\left(I_t\right) \tag{2.1}$$

An algorithm is considered to 'solve' the above-defined MAB problem if it achieves a sub-linear regret over the time horizon T (i.e. the average regret per timestep approaches zero in the long run). We are going to look at four algorithms in this chapter that can be used to solve the stochastic multi-armed bandit problem. The lower bound on the regret of the MAB problem is the best any algorithm can perform. The instance-dependant (a common assumption where the difference between the expected rewards of the arms are large) lower bound is $\Omega\left(\log T\right)$([38]) and general case lower bound is $\Omega\left(\sqrt{T}\right)$ ([3]).

2.1.1 Explore-Then-Exploit

Due to the nature of the problem, a way to reason about this problem is to employ an exploration phase for a constant number of rounds and get estimates for the arms' rewards. Once this phase is over, we can simply use the arm which produced the highest reward. The latter phase can be called the exploitation phase.

Algorithm 2 has been taken from [38]. We choose N with the knowledge of T to minimize the overall regret. From [38], we know that a sub-linear regret of $\tilde{O}\left(T^{2/3}\right)$ is achieved.

Algorithm 2: Explore-then-Exploit Algorithm with K arms.

Explore all the K arms N times.
Pick the arm ($\hat{a}$) with highest average reward. (break ties arbitrarily)
Pick $\hat{a}$ for the rest of the time horizon.

2.1.2 ε-Greedy Algorithm

One can improve over the last algorithm, is by spreading the exploration phase more uniformly over the entire time horizon. At each time step, we can explore with a small probability, else we just keep exploiting. It is called the ε-greedy algorithm.

Another way to look at this algorithm is to think of this as an improvement on the greedy algorithm. A simple greedy algorithm does not work, as shown by [39]. Using the simple greedy, we might get stuck with a sub-optimal arm and keep on picking it, without getting to know about a better arm.

One way to overcome this problem is to have a small amount of exploration while taking the greedy choice most of the time. ε-greedy does this precisely and achieves a sub-linear regret of $\tilde{O}\left(T^{2/3}\right)$ ([38]). With a better choice of ε, (i.e decaying with time) [2] showed that a regret of $O\left(\log T\right)$ can be achieved.

Algorithm 3: ε-Greedy Algorithm

Input Parameters: $K \in \mathcal{K}$ arms, T rounds, some decreasing function $f(;)$
for each round $t \in [1, T]$ **do**
    Let $\varepsilon_t = f(t)$
    With probability $1 - \varepsilon_t$, the algorithm picks $I_t = \arg\max_{i \in \mathcal{K}} \hat{\mu}_i(t)$
    With probability ε, it picks $I_t \in \mathcal{K}$ uniformly at random.
end

2.1.3 Upper Confidence Bounds

The problem with ε-Greedy is that it has an inefficient exploration process. It keeps on exploring uniformly, even after the optimal arm is found. Another method to overcome this to use upper confidence bounds while selecting arms at each step. The philosophy used here is of

optimism under uncertainty where the arm that promises the highest reward in future is picked. The upper confidence bound of the reward estimate captures the mentioned promise.

More precisely, at each time, the algorithm picks the arm with the maximum value of $\hat{\mu}_t(i) + c_t(i)$, where $c_t(i)$ is the upper confidence bound of the arm $i \in \mathcal{K}$ at time t. Since the rewards are assumed to be generated from a distribution, and the samples are i.i.d, we can employ the Hoeffding's inequality to find that $c_t(i) = \sqrt{\frac{2 \log t}{N_t(i)}}$.

Algorithm 4: UCB-1 Algorithm

Input Parameters: $K \in \mathcal{K}$ arms, T
for each round $t \in [1, T]$ **do**
  if $t \leq K$ **then**
    The algorithm picks $I_t = K$
  else
    The algorithm picks $I_t = \arg\max_{i \in \mathcal{K}} \hat{\mu}_t(i) + \sqrt{\frac{2 \log t}{N_t(i)}}$
  end
end

The regret bound for UCB-1 (Algorithm 4), with bernoulli rewards distributions $\mathcal{D}(i)$; $\forall i \in \mathcal{K}$ with support $[0, 1]$, and with well separated expected rewards (i.e large expected reward differences between arms) of the arms, is in the order of $O(\log T)$ ([2]).

2.1.4 Thompson Sampling

Algorithm 5: Thompson Sampling Algorithm

for each round $t \in [1, T]$ **do**
  The algorithm independantly samples $\theta_t(i)$ from distribution $\mathcal{D}_t(i)$.
  It selects the arm $I_t = \arg\max_i \theta_t(i)$
end

Thompson Sampling (Algorithm 5, [40], [36]) is a Bayesian approach to the stochastic bandit problem. The algorithm starts with a prior distribution on each arm's expected reward and updates the distribution after the said arm is pulled and a reward is observed. At each timestep, the algorithm samples the expected reward (i.e. $\theta_t(i)$) for each arm according to their (posterior) distribution (i.e. $\mathcal{D}_t(i)$) then selects the arm with the highest sample reward. The choices of the

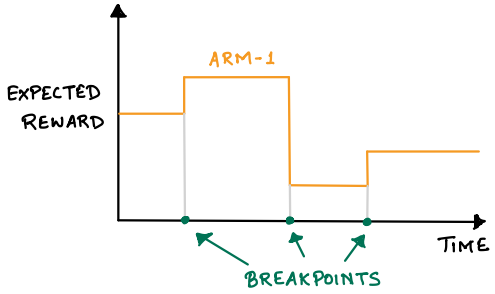

Figure 2.1: Piecewise Stationary Environment with 3 breakpoints

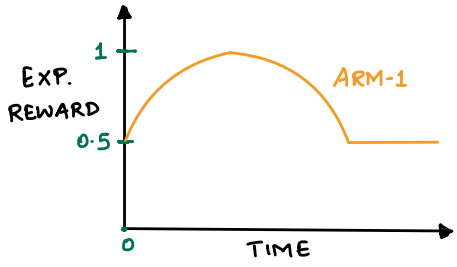

Figure 2.2: Continuously Changing Environment with variation budget of 1

prior (a popular one being the Beta distribution) or the posterior (a popular being the Gaussian distribution) distributions are problem-specific. Thompson Sampling achieves $O(\log T)$ regret ([1]).

2.2 Non-stationary Bandits

The Non-Stationary setting is the variant of the MAB problem with K arms. The reward of each arm $i \in \mathcal{K}$, where $\mathcal{K} = \{k\}_{k=1}^{K}$ follows an unknown distribution $\mathcal{D}_t(i)$ with support $[0, 1]$ and mean $\mu_t(i) = \mathbb{E}[X_i(t)]$, which may change with time.

At each time step $t \in [1, T]$, where T is the time horizon, a player will pull one arm $I_t \in \mathcal{K}$ and receive a reward $X_t(I_t) \sim \mathcal{D}_t(I_t)$, which is fed back to the algorithm scheme. In this case, $N_t(i) = \sum_{\tau=1}^{t} \mathbb{1}_{\{I_\tau = i\}}$ denote the number of times the arm i was played till time t and $\hat{\mu}_t(i) = (1/N_t(i)) \sum_{\tau=1}^{t} X_t(i) \mathbb{1}_{\{I_\tau = i\}}$ be the corresponding empirical average reward.

2.2.1 Environments

We will discuss two general ways to model Non-Stationarity in bandit settings: the Piecewise-Stationary and the Continuously-Changing environment.

The Piecewise-Stationary environment (Figure 2.1) models the scenario when changes occur in the reward distributions of the arms at abrupt time instants called breakpoints. The distributions remain stationary between breakpoints.

The Continuously-Changing environment (Figure 2.2) models the scenario when the reward

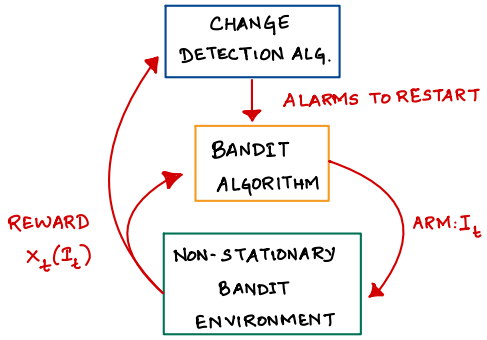

Figure 2.3: Active adaptation

distribution of the arms can change an arbitrary number of times. However, to maintain tractability, this environment has a variation budget, which limits the amount of total change throughout the time horizon.

The formulation of these environments is discussed in chapter 3. The two general approaches to solve a Non-stationary MAB problem are *active* and *passive*, discussed in the next subsection.

2.2.2 Active Adaption

The idea here is to take the changes head-on. Generally in this framework (Figure 2.3), suggested by [27] we have a change detection algorithm, which checks if there is any change in the environment and notifies the bandit algorithm (in [27]'s case, UCB-1) to restart. The mentioned method (named as CUSUM-UCB) works for the piecewise-stationary environment.

Algorithm 6 embodies the active adaptation framework (Figure 2.3), where they have UCB-1 (Algorithm 4) updates for the bandit algorithm, and they use a change detection algorithm as a submodule to check for changes in the non-stationary environment.

The authors provide a change detection algorithm (Algorithm 7) that works in the bandit setting. They use the first M samples to calculate the average, $\hat{u}_0 = (\sum_{k=1}^{M} y_k)/M$. Then they construct two random walks which have negative mean drifts before the change point and have positive mean drifts after the change. Algorithm 7 is a two-sided cumulative sum algorithm that monitors the possible positive and negative mean shifts. Let s_k^+ be the step of the upper random

walk and s_k^- be the step of the lower random walk. They provide the following definitions for the steps as

$$(s_k^+, s_k^-) = (y_k - \hat{u}_0 - \epsilon, \hat{u}_0 - y_k - \epsilon)\mathbb{1}_{\{k>M\}} \tag{2.2}$$

And, similarly, they define g_k^+ as the positive drift of the upper random walk, and g_k^- as the positive drift of the lower random walk as

$$g_k^+ = \max(0, g_{k-1}^+ + s_k^+),\ g_k^- = \max(0, g_{k-1}^- + s_k^-) \tag{2.3}$$

Algorithm 6: CD-UCB Algorithm

```
Input Parameters: T, α and and algorithm CD()
Initialize τ_i = 1, ∀i
for t ∈ [1, T] do
    Update the quantities required for UCB-1 (Algorithm 4)
    Play arm I_t and observe X_t(I_t)
    if CD(I_t, X_t(I_t)) = 1 then
        τ_{I_t} = t + 1
        reset CD(I_t,.)
    end
end
```

They proved that the regret bound for CUSUM-UCB is of the order $O\left(\sqrt{T\mathcal{B}_T \log\left(\frac{T}{\mathcal{B}_T}\right)}\right)$.

2.2.3 Passive Adaption

Unlike the active adaption, these algorithms don't track the changes in the environment but make the algorithms robust enough to have sub-linear regrets despite the changes in the background. We will look at three algorithms that embrace this approach.

The first two algorithms (8, 9) discussed are employed in a piecewise-stationary environment, and the last algorithm (11) works for continuously-changing environments.

Discounted-UCB (Algorithm 8 by [22]) finds the 'optimal' arm, while balancing exploration and exploitation. This algorithm puts more weight, according to the parameter $\gamma \in (0, 1]$ to the recent rewards in comparison to the older ones to balance 'forgetting' and 'remembering', as a breakpoint in the recent past makes the rewards before that misleading for estimation. The discount factor is chosen to minimize the overall expected reward.

Algorithm 7: Two-sided CUSUM

Input Parameters: ϵ, M, h and $\{y_k\}_{k\geq 1}$
Initialize $g_0^+ = 0$ and $g_0^- = 0$
for each k **do**
    Calculate s_k^- and s_k^+ according to equation 2.2
    Update $g_0^+ = 0$ and $g_0^- = 0$ according to equation 2.3
    if $g_0^+ \geq h$ **and** $g_0^- \geq h$ **then**
        Return 1
    end
end

Algorithm 8: Discounted-UCB (D-UCB)

Input Parameters: $\tilde{\mu}, \gamma$
for $t \leq K$ return $I_t = t$
for $t > K$ return $I_t = \arg\max_{i\in\mathcal{K}} \tilde{\mu}_t(\gamma, i) + c_t(\gamma, i)$

Algorithm 9: Sliding-Window-UCB (SW-UCB)

Input Parameters: $\tilde{\mu}, \tau$
for $t \leq K$ return $I_t = t$
for $t > K$ return $I_t = \arg\max_{i\in\mathcal{K}} \tilde{\mu}_t(\tau, i) + c_t(\tau, i)$

For the Discounted-UCB algorithm, the following quantities are modified by UCB-1 (Algorithm 4):

$$\tilde{\mu}_t(\gamma, i) = \frac{1}{N_t(\gamma, i)} \sum_{\tau=1}^{t} \gamma^{t-\tau} \mathbb{1}_{\{I_\tau = i\}} X_\tau(i)$$

$$N_t(\gamma, i) = \sum_{\tau=1}^{t} \gamma^{t-\tau} \mathbb{1}_{\{I_\tau = i\}}$$

$$c_t(\gamma, i) = 2\sqrt{\frac{\xi \log n_t(\gamma)}{N_t(\gamma, i)}}, \; n_t(\gamma) = \sum_{i=1}^{K} N_t(\gamma, i)$$

For appropriate value of ξ. The regret bound for this algorithm is due to [15], and has a regret of $O(\sqrt{T\mathcal{B}_T} \log T)$.

We have a slightly better algorithm, which is Sliding-Window-UCB (Algorithm 9) by [15]. In this algorithm, we don't have a discount factor, but a sliding window of the history. The algorithm 'remembers' the records in the last τ (the window size) timesteps for any timestep t, and 'forgets' the ones before that. The size of the window is generally chosen to minimize the overall expected regret. The modified quantities for the Sliding-Window UCB (Algorithm 9) are shown below:

$$\tilde{\mu}_t(\tau, i) = \frac{1}{N_t(\tau, i)} \sum_{s=t-\tau+1}^{t} \mathbb{1}_{\{I_s = i\}} X_s(i)$$

$$N_t(\tau, i) = \sum_{s=t-\tau+1}^{t} \mathbb{1}_{\{I_s = i\}}$$

$$\mathcal{N}_t(\tau, i) = N_t(t, i) = \sum_{s=1}^{t} \mathbb{1}_{\{I_s = i\}}$$

$$c_t(\tau, i) = \sqrt{\frac{\xi \log(\min(t, \tau))}{N_t(\tau, i)}}$$

For appropriate value of ξ. This algorithm has a slightly better regret bound of $O\left(\sqrt{T\mathcal{B}_T \log T}\right)$.

The next algorithm is Rexp3 by [5] which works for the continuously-changing environment. In this scheme, the algorithm works in batches, and it restarts a bandit algorithm (which is used as a submodule) once a batch is over. The batch size is chosen to minimize the overall expected reward. A variant of this algorithm (11) is discussed in chapter 3 in proper detail.

For this work, we have only considered the passive approach for countering non-stationarity. We will define the problem precisely in chapter 3.

2.2.4 Lower Bounds

For the continuously-changing environment, [5] proved that the lower bound is $\Omega\left(T^{2/3}\right)$. For the piecewise-stationary environment, the lower bound is shown to be $\Omega\left(\sqrt{T}\right)$ by [15]. In this section, we will see an alternate (and simpler in some sense) proof which also shows the same.

Let us define the worst-case regret of a policy π on a set of stochastic non-stationary bandit environments $\mathcal{E}$ and time horizon T, to be

$$\mathcal{R}_T(\pi, \mathcal{E}) = \sup_{e \in \mathcal{E}} \mathcal{R}_T(\pi, e) \tag{2.4}$$

Let Π be the set of all policies. Then the *minimax regret* is defined as

$$\mathcal{R}_T^*(\mathcal{E}) = \inf_{\pi \in \Pi} \mathcal{R}_T(\pi, \mathcal{E}) = \inf_{\pi \in \Pi} \sup_{e \in \mathcal{E}} \mathcal{R}_T(\pi, e) \tag{2.5}$$

The main idea behind this proof has been inspired by [25], which says that we need to reduce the bandit problem to hypothesis testing. We must select two bandit problem instances in such a way that the instances are as close to each other as possible, and must be competing against each other such that we can choose an action or a sequence of actions that is good for one bandit and is not for the other. The lower bound will follow by optimizing this trade-off.

Assumption 2.2.1. *We assume that any bandit arm i in the bandit instances of $\mathcal{E}$, will have $\mathtt{KL}(\mathcal{D}_{\mu_1}, \mathcal{D}_{\mu_2}) = c(\mu_1 - \mu_2)^2$ for some constant c and where $\mathcal{D}_z$ is any distribution with support $[0, 1]$ and with mean z.*

In this proof, we are going to use the Bretagnolle-Huber inequality, whose formal statement is given in the following lemma.

Lemma 2.2.2. *Let P and Q be the probability measures on the same measurable space $(\Omega, \mathcal{F})$, and let $A \in \mathcal{F}$ be an arbitrary event. Then,*

$$P(A) + Q(A^c) \geq \frac{1}{2} \exp\left(-\mathtt{KL}(P, Q)\right) \tag{2.6}$$

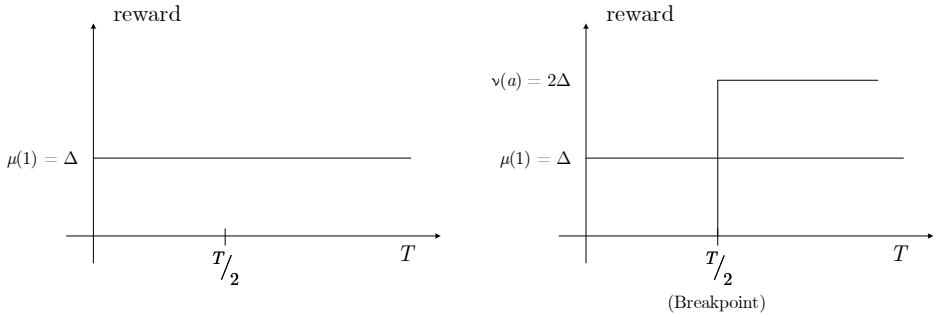

Figure 2.4: Bandit instance $\mathcal{E}$.

Theorem 2.2.3 (Main Regret Lower Bound)**.** *Let $K > 1$ and $T \geq K - 1$ and assumption 2.2.4 be true. Then, for a policy class Π and a non-stationary environment $\mathcal{E}$, the minimax regret is*

$$\mathcal{R}_T^* (\mathcal{E}) \geq \frac{1}{32} \sqrt{\frac{(K-1)\,T}{c}} \tag{2.7}$$

Proof. Let us fix a policy $\pi \in \Pi$. Let $\Delta \in [0, 1/2]$ be some constant to be chosen later. In $\mathcal{E}$, we will have only one breakpoint at $t = T/2$ (Figure 2.4). Let us start with a bandit instance $e_\mu = (\mathcal{D}_t^\mu(i))_{i=1}^K$ with reward vector $\mu \in [0,1]^{K \times T}$ such that $\mu_t(i) = \mathbb{E}\left[\mathcal{D}_t^\mu(i)\right]$, is defined as

$$\forall t; \mu_t(i) = \begin{cases} \Delta & \text{if } i = 1, \\ 0 & \textit{otherwise} \end{cases}$$

When the policy π will interact with the bandit instance e_μ for a horizon T, it will give rise to the composite probability distribution $\mathbb{P}_{e_\mu}^\pi$, and the expectation under this distribution will be denoted by $\mathbb{E}_{e_\mu}^\pi$. For the second bandit instance $e_\nu = (\mathcal{D}_t^\nu(i))_{i=1}^K$, we will have a reward vector $\nu \in [0,1]^{K \times T}$, such that $\nu_t(i) = \mathbb{E}\left[\mathcal{D}_t^\nu(i)\right]$, is defined as

$$\nu_t(i) = \begin{cases} 2\Delta & \text{if } i = a \text{ and } t \geq T/2, \\ \mu_t(i) & \textit{otherwise} \end{cases}$$

where $a = \arg\min_{j \neq 1} \mathbb{E}_{e_\mu}^\pi \left[N_T(j)\right]$.

Since we want to make the bandit instances as close to each other as possible and at the same time want π to perform badly on ν, we chose the arm $a \in \mathcal{K}$, which is explored/played the least number of times.

We know that $\sum_{i=1}^{K} \mathbb{E}_{e_\mu}^{\pi}[N_T(i)] \geq (K-1)\mathbb{E}_{e_\mu}[N_T(a)]$, it holds that $\mathbb{E}_{e_\mu}^{\pi}[N_T(a)] \leq T/(K-1)$. Similar to the first bandit instance, we have $\mathbb{P}_{e_\nu}^{\pi}$ as the composite probability distribution and $\mathbb{E}_{e_\nu}^{\pi}$ as the corresponding expectation under it.

$$\mathbb{P}_{e_\nu}^{\pi}[X_{1:T}|I_{1:T}] = \prod_{t=1}^{T} \mathcal{D}_t^{\nu}(I_t)(X_t) \tag{2.8}$$

where $X = (X_1, X_2, .., X_T)$ is the vector of realized rewards under policy π when $I = (I_1, I_2, .., I_T)$ arms where played.

The worst case regrets for each of the bandit instances with respect to the policy π are as shown below:

$$\mathcal{R}_T(\pi, e_\mu) > \mathbb{P}_{e_\mu}^{\pi}\left[N_{T/2}(1) < T/4\right] \cdot \frac{T\Delta}{4} \tag{2.9}$$

$$\mathcal{R}_T(\pi, e_\nu) > \mathbb{P}_{e_\nu}^{\pi}\left[N_{T/2}(1) \geq T/4\right] \cdot \frac{T\Delta}{4} \tag{2.10}$$

Let $\mathcal{D}_0 = \mathcal{D}_t^{\mu}(a)$ and $\mathcal{D}_{2\Delta} = \mathcal{D}_t^{\nu}(a)$ for $t \geq T/2$. Now, we can use the Bretagnolle-Huber inequality to find the lower bound of the sum of worst-case regrets.

$$\begin{aligned}
\mathcal{R}_T(\pi, e_\mu) + \mathcal{R}_T(\pi, e_\nu) &> \frac{T\Delta}{4} \cdot \left(\mathbb{P}_{e_\mu}^{\pi}\left[N_{T/2}(1) < T/4\right] + \mathbb{P}_{e_\nu}^{\pi}\left[N_{T/2}(1) \geq T/4\right]\right) \\
&\geq \frac{T\Delta}{8} \cdot \exp\left(-\mathtt{KL}\left(\mathbb{P}_{e_\mu}^{\pi}, \mathbb{P}_{e_\nu}^{\pi}\right)\right) \\
&= \frac{T\Delta}{8} \cdot \exp\left(-\sum_{t=T/2}^{T} \mathbb{P}_{e_\mu}^{\pi}\left[I_t = a\right] \mathtt{KL}\left(\mathcal{D}_0, \mathcal{D}_{2\Delta}\right)\right) \\
&= \frac{T\Delta}{8} \cdot \exp\left(-4c\Delta^2 \mathbb{E}_{e_\mu}^{\pi}\left[N_{T/2:T}(a)\right]\right) \\
&= \frac{T\Delta}{8} \cdot \exp\left(\frac{-4Tc\Delta^2}{K-1}\right)
\end{aligned} \tag{2.11}$$

The result is completed by choosing $\Delta = \sqrt{(K-1)/4Tc} \leq 1/2$. The final steps are lower bounding $\exp(-1)$ and using $2\max(a, b) \geq a + b$.

□

Remark 1. *From theorem 2.2.4, we conclude that the lower bound for the minimax regret for the non-stationary environment $\mathcal{E}$ is $\Omega(\sqrt{T})$. In the above method, there is an extra assumption(2.2.4) which does not feature in [15]'s proof. [25] (Bandit Algorithms) have shown in section 15.3 that it is almost always the case, and is not an unreasonable assumption to make.*

2.3 Incentivized Exploration of Stochastic Bandits

Early work on incentivized exploration and learning includes [13, 23, 9] that introduced a Bayesian Incentivized model with discounted regret and compensation, and [31] that considered the non-discounted case and showed that their algorithm achieved $O(\sqrt{T})$ regret. [41] analyzed the non-Bayesian and non-discount reward case and showed $O(\log T)$ regret and compensation. [29] considered the biased user feedback under the influence of incentives and showed that despite the reward drift, the proposed algorithms achieve $O(\log T)$ regret and compensation.

Related work includes those on the robustness of MAB to adversarial attack, e.g., [30] that proposed a multi-layer active arm elimination race algorithm for stochastic bandits with adversarial corruptions, [12] that studied the strategic behaviour of rational arms and showed that UCB, ϵ-Greedy and Thompson Sampling achieve $O(\max(B, \log T))$ regret bound under any strategy of the arms, with B being the total budget.

Related work also includes those on the Bayesian Incentive Compatible (BIC) bandit exploration, see, e.g., [32, 31, 10, 37], where the principal wishes to persuade the agent to take some action which benefits the principal, known as Bayesian Persuasion [21]. See also the text by [38] that provides a review of the broad area of incentivized exploration from various aspects.

The setting which mostly inspires this work is from [29]. The authors present a scenario (introduced in Chapter 1) where the decision-maker(principal) and the arm-puller(player) are different entities and the principal provides compensation to the player to encourage exploration. The authors assumed a stationary bandit environment and showed that their framework achieves sub-linear regret and compensation of $O(\log T)$. This setting forms the basis of the framework we

use (Algorithm 10) for solving the piecewise-stationary environment (section 3.2).

Chapter 3

Problem Formulation

3.1 Basic Setting

Consider a variant of the multi-armed bandit problem where a principal has a set $\mathcal{K}$ of K arms. The rewards $\{X_t(i)\}_{t=1}^T$ of each arm $i \in \mathcal{K}$ is modelled as a sequence of independent random variables with possibly different distributions that are unknown to the agents/principal and *may vary across time* (Figure 1.3). Denote by $\mu_t(i) = \mathbb{E}[X_t(i)]$ the mean reward of arm $i \in \mathcal{K}$ at time step $t \leq T$. At each time step t, a player pulls one arm $I_t \in \mathcal{K}$ and receives a reward r_t, which is then fed back to the principal and other players.

We consider a real-life scenario where the principal and the players may have different interests (Figure 1.4). The principal would like the players to select the arms in such a way to adequately explore different arms in order to maximize the accumulated rewards. However, a player may be heavily influenced by the feedback of others and behaves myopically in face of uncertainty, i.e., pulls the arm with the currently highest empirical rewards (exploitation only). Under such situations, in order to incentivize the players to explore, the principal may provide some compensation χ_t to the player such that she will pull the arm suggested by a certain bandit algorithm that achieves a good exploration-exploitation tradeoff and maximizes the accumulated rewards. However, since the player receives some compensation, her feedback from the pull can be biased and may include some drift δ_t on top of the "true" reward $X_t(I_t)$, captured by some unknown, non-decreasing function $\delta_t = f(\chi_t)$ of the compensation χ_t. Notice that the biased feedback $X_t(I_t) + \delta_t$ is collected, and the principal and payers cannot distinguish either $X_t(I_t)$ or δ_t from it.

Assumption 3.1.1. *The "true" reward has a normalized support of* $[0, 1]$*, and we assume that the drifted reward at any time will be projected onto* $[0, 1]$*.*

The two metrics used to judge the algorithms are regret and compensation. The former is the accumulated reward difference between the best arm (in hindsight) and the arm played at every time-step, and the latter is the accumulated difference between the rewards of the arm chosen greedily by the player and the arm chosen by the principal using some bandit algorithm.

We aim to understand and characterize the efficacy of the proposed compensation scheme in incentivizing exploration, in particular, if the algorithm is robust to the drifted reward so that the decisions based on biased feedback achieve a sublinear regret and if the proposed incentive mechanism is cost-efficient (i.e. sublinear compensation) to the principal. While existing work such as [29] has studied this important question in the setting of a stationary bandit, in this thesis, we investigate the more challenging setting of a non-stationary bandit as will be described next.

3.2 Piecewise-Stationary Environment

Algorithm 10: Incentivized MAB under Reward Drift

for $t \in [1, T]$ **do**
  $I_t = \text{PRINCIPALALG}$(required parameters)
  $G_t = \arg\max_{i \in \mathcal{K}} \tilde{\mu}_i(t)$
  if $G_t \neq I_t$ **then**
    Principal gives compensation of $\chi_t = \tilde{\mu}_{G_t} - \tilde{\mu}_{I_t}$
    Reward received after playing I_t is $r_t = X_t(I_t) + \delta_t$ where reward drift $\delta_t = f(\chi_t)$
  else
    Reward received is $r_t = X_t(I_t)$ with no compensation.
  end
end

In the *abruptly changing environment*, the reward distributions change at unknown time instants called *breakpoints* and remain fixed otherwise (hence piecewise-stationary environment) (Figure 2.1). We denote by $\mathcal{B}_T$ the total number of breakpoints that occur before time T.

Algorithm 10, adopted from [29], describes a framework of incentivized exploration for the piecewise-stationary environment. The principal chooses an arm I_t according to a certain non-

stationary bandit algorithm PrincipalAlg (that will be discussed in Chapter 4), and the player will choose the greedy arm G_t if without any compensation. The principal then provides a compensation χ_t that is the difference in the empirical average rewards of the arms to the player. The bias δ_t of the player after receiving the compensation is added to the "true" reward $X_t(I_t)$. The function f is assumed to be Lipschitz continuous in accordance with Assumption 1 in [29].

3.3 Continuously-Changing Environment

Algorithm 11: Restarting technique with a Stochastic Bandit algorithm

1 **Input Parameters**: $\tau, \mathcal{K}, T$
2 **Initialize**: $j = 1$
3 **while** $j \leq \lceil T/\tau \rceil$ **do**
4 set $\alpha = (j-1)\tau$
5 **for** $t = 1, .., \min\{T, \alpha + \tau\}$ **do**
6 $I_t = \text{BanditAlg}(t)$
7 $G_t = \arg\max_{k \in \mathcal{K}} \tilde{\mu}_t(k)$
8 steps 4-9 from Algorithm 10
9 **end**
10 Increment $j = j + 1$, and return at step 3.
11 **end**

In this environment (Figure 2.2), the mean rewards of the arms can change an arbitrary number of times but have a variation budget, which limits the total change throughout the horizon [5]. Algorithm 11 describes a framework of incentivized exploration for the continuously-changing environment. Here τ is the batch size, j is the batch number, and lines 4-9 describe the sequence of operations that are carried out on a single batch. G_t is the greedy arm chosen by the player if without any compensation, and I_t is the arm chosen by the principal according to certain stochastic bandit algorithm such as UCB1, ε-greedy or Thompson sampling as mentioned in Chapter 5.

We denote the best possible expected reward at any epoch t by $\mu_t^* = \max_{i \in \mathcal{K}} \mu_t(i)$. We assume that the expected reward of each arm $\mu_t(i)$ may change at any (possibly every) decision epoch. Let V_t be a non-decreasing sequence $\forall t \in [1, T]$ of positive real numbers such that $V_1 = 0$ and $KV_t \leq t; \forall t$ and for normalizing purposes set $V_2 = 2 \cdot K^{-1}$. We refer to V_T as the variation

budget over the time horizon T.

The *temporal uncertainty set*, is the set of reward vector sequences that are subject to the variation budget V_T over the set of decision epochs $\{1, .., T\}$:

$$\mathcal{V} = \left\{ \mu \in [0,1]^{K \times T} : \sum_{t=1}^{T} \sup_{i \in \mathcal{K}} |\mu_t(i) - \mu_{t+1}(i)| \leq V_T \right\}.$$

Almost all the stochastic bandit algorithms assume that the difference between any pair of arms' mean rewards is large (i.e., the arms' mean rewards are well separated) in order to maintain mathematical tractability (e.g., avoiding zero or near-zero value in the denominator). However, in the continuously-changing environment, the nature of reward change might make this typical assumption too strong. Therefore, we need to give the rewards more freedom to vary but impose certain assumptions to make them mathematically tractable.

Definition 3.3.1. *The minimum difference between the average mean rewards of the best overall arm and any other arm i within a single batch $\Delta(i)$, is defined as:*

$$\Delta(i) = \min_{j \in [1,m]} \frac{1}{\tau} \sum_{t \in \mathcal{T}_j} (\mu_t^* - \mu_t(i))$$

where m is the total number of batches and $\mathcal{T}_j$ is the set of timestamps within the batch j.

Assumption 3.3.2. *There exists a constant $M \in (0,1)$ such that $\Delta(i) \geq M$ for any $i \in \mathcal{K}$.*

Definition 3.3.3. *The event $\mathcal{E}$ is defined as $\mu_t(i) - \mu_t(j) \leq \varepsilon$ for given $\varepsilon \in (0,1)$ and $i, j \in \mathcal{K}$.*

Assumption 3.3.4. *For any given time epoch α, there exists $\beta \in (0,1)$ such that $\sum_{\alpha \in [1,T]} \sum_{i \in \mathcal{K}} \mathbb{1}_{\{\mathcal{E}\}} \leq \alpha^{\beta}$.*

Chapter 4

Piecewise Stationary Environment

In this section, we investigate the piecewise-stationary environment, and employ the Discounted-UCB algorithm (Algorithm 12), proposed in [22], and the Sliding-Window-UCB algorithm (Algorithm 13), proposed in [15], for PrincipalAlg in Algorithm 10.

Algorithm 12: Discounted-UCB (D-UCB)

Input Parameters: $\tilde{\mu}, \gamma$
for $t \leq K$ return $I_t = t$
for $t > K$ return $I_t = \arg\max_{i \in \mathcal{K}} \tilde{\mu}_t(\gamma, i) + c_t(\gamma, i)$

In D-UCB, γ is the discounting factor. We denote $\tilde{\mu}_t(\gamma, i) = (1/N_t(\gamma, i)) \sum_{\alpha=1}^{t} \gamma^{t-\alpha} \mathbb{1}_{\{I_\alpha = i\}} X_\alpha(i)$ is the estimated drifted discounted average of the expected rewards of the arm i, and $\hat{\mu}_t(\gamma, i)$ denote the pure discounted average. The weighted frequency of i till time t is denoted by $N_t(\gamma, i) = \sum_{\alpha=1}^{t} \gamma^{t-\alpha} \mathbb{1}_{\{I_\alpha = i\}}$, and the unweighted frequency by $\mathcal{N}_t(i) = N_t(1, i)$. The padding function is $c_t(\gamma, i) = 2\sqrt{\xi \log n_t(\gamma)/N_t(\gamma, i)}$ For appropriate value of ξ. We also denote the sum of the weighted frequencies for all arms till time t to be $n_t(\gamma) = \sum_{i=1}^{K} N_t(\gamma, i)$.

Algorithm 13: Sliding-Window-UCB (SW-UCB)

Input Parameters: $\tilde{\mu}, \tau$
for $t \leq K$ return $I_t = t$
for $t > K$ return $I_t = \arg\max_{i \in \mathcal{K}} \tilde{\mu}_t(\tau, i) + c_t(\tau, i)$

In SW-UCB, τ is the size of the sliding window. We denote $\tilde{\mu}_t(\tau, i)$ to be the estimated drifted average of the expected rewards of the arm i which is $(1/N_t(\tau, i)) \sum_{s=t-\tau+1}^{t} \mathbb{1}_{\{I_s = i\}} X_s(i)$, and $\hat{\mu}_t(\tau, i)$ denote the pure average. The weighted frequency of i till time t is denoted by $N_t(\tau, i) =$

$\sum_{s=t-\tau+1}^{t} \mathbb{1}_{\{I_s=i\}}$, and the unweighted frequency by $\mathcal{N}_t(\tau, i) = N_t(t, i)$. The padding function is $c_t(\tau, i) = \sqrt{\xi \log(\min(t, \tau))/N_t(\tau, i)}$ For appropriate value of ξ.

4.1 Theoretical Results

4.1.1 Regret

We look at the regret bounds for D-UCB and SW-UCB in this section. Let $\mathcal{B}_T$ be the total number of breakpoints before time T. Since the rewards have normalized support of $[0, 1]$, the regret contribution for an arm i will be upper bounded by $\mathbb{E}[\widehat{N}_T(i)]$, where $\widehat{N}_T(i)$ is the total number of times i was played when it was not the best arm during the first T rounds, i.e.,

$$\widehat{N}_T(i) = \sum_{t=1}^{T} \mathbb{1}_{\{I_t=i\neq i^*\}}. \tag{4.1}$$

4.1.1.1 Discounted UCB with Algorithm 10

Let $\tilde{\mu}_t(\gamma, i)$ denote the estimated drifted discounted average of the expected rewards of the arm i, and $\hat{\mu}_t(\gamma, i)$ the pure discounted average.

We denote by $\Delta_{\mu_T}(i)$ the minimum of the difference of the expected reward $\mu_t(i^*)$ of the best arm and the expected reward $\mu_t(i)$ of the arm i over $t \in [1, T]$ when i is not the optimal arm, i.e.,

$$\Delta_{\mu_T}(i) = \min_{t\in[1,T];\ i\neq i^*} \{\mu_t(i^*) - \mu_t(i)\}. \tag{4.2}$$

Assumption 4.1.1. *[Assumption 1 from [29]] The reward drift function $f_t(x)$ is non-decreasing with $f_t(0) = 0$, and is Lipschitz continuous, i.e., there exists a constant l_t such that $|f_t(x) - f_t(y)| \leq l_t|x - y|$ for any x and y. Moreover, without loss of generality, the rewards $X_i(t)$ are assumed to be in $[0, 1]$ for all $i \in \mathcal{K}$ and $t \in [1, T]$.*

Lemma 4.1.2. *The sum of the weighted play frequencies of all the arms $i \in \mathcal{K}$ till time t, denoted by $n_t(\gamma)$, is upper bounded by* $\min(t, 1/(1-\gamma))$.

Proof. We have

$$\begin{aligned} n_t(\gamma) &= \sum_{i=1}^{K} N_t(\gamma, i) \\ &= \sum_{i=1}^{K} \sum_{\tau=1}^{t} \mathbb{1}_{\{I_\tau = i\}} \gamma^{t-\tau} \\ &= \sum_{\tau=1}^{t} \gamma^{t-\tau} \leq \sum_{\tau=1}^{\infty} \gamma^{\tau} = \frac{1}{1-\gamma}. \end{aligned} \tag{4.3}$$

Notice that $n_t(\gamma)$ cannot be more than t, we get the final upper bound. □

Lemma 4.1.3. *The total discounted reward drift of arm i till time t, denoted by $D_t(\gamma, i)$, is upper bounded by $2l\mathcal{N}_t(i)\sqrt{\xi \log(n_t(\gamma))/(1-\gamma)}$, where $l = \max_t l_t$.*

Proof. The principal has to provide compensation when both of the following inequalities hold:

$$\tilde{\mu}_t(\gamma, G_t) \geq \tilde{\mu}_t(\gamma, I_t), \tag{4.4}$$

$$\tilde{\mu}_t(\gamma, G_t) + c_t(\gamma, G_t) \geq \tilde{\mu}_t(\gamma, I_t) + c_t(\gamma, I_t). \tag{4.5}$$

The above two inequalities imply $\tilde{\mu}_t(\gamma, G_t) - \tilde{\mu}_t(\gamma, I_t) \leq c_t(\gamma, I_t) - c_t(\gamma, G_t)$. By Assumption 4.1.1,

$$\begin{aligned} \tau &\leq l_t \left(c_t(\gamma, I_t) - c_t(\gamma, G_t)\right) \\ &\leq l_t \left(2\sqrt{\frac{\xi \log(n_t(\gamma))}{N_t(\gamma, I_t)}} - 2\sqrt{\frac{\xi \log(n_t(\gamma))}{N_t(\gamma, G_t)}}\right) \\ &\leq 2l_t \sqrt{\frac{\xi \log(n_t(\gamma))}{N_t(\gamma, I_t)}}. \end{aligned} \tag{4.6}$$

Thus, the total drift

$$
\begin{aligned}
D_t(\gamma, i) &= \sum_{\tau=1}^{t} \mathbb{1}_{\{I_\tau = i\}} \delta_\tau \\
&= \sum_{\tau=1}^{t} \mathbb{1}_{\{I_\tau = i\}} 2l_\tau \sqrt{\frac{\xi \log(n_t(\gamma))}{N_\tau(\gamma, i)}} \\
&\leq 2l\sqrt{\xi \log(n_t(\gamma))} \sum_{\tau=1}^{t} \mathbb{1}_{\{I_\tau = i\}} \frac{1}{\sqrt{N_\tau(\gamma, i)}} \\
&= 2l\sqrt{\xi(1-\gamma)\log(n_t(\gamma))} \sum_{\tau=1}^{\mathcal{N}_t(i)} \frac{1}{\sqrt{1-\gamma^\tau}} \\
&\leq 2l\mathcal{N}_t(i)\sqrt{\frac{\xi \log(n_t(\gamma))}{1-\gamma}}.
\end{aligned} \tag{4.7}
$$

□

The following theorem is for the regret when Discounted-UCB is used as the principal's algorithm within Algorithm 10.

Theorem 4.1.4. *Let $\xi > 1/2$, $T > 1$ and $\gamma \in (0, 1)$. For any arm $i \in \mathcal{K}$, we have*

$$
\mathbb{E}\left[\widehat{N}_T(i)\right] \leq \left(B(\gamma)T(1-\gamma) + C(\gamma)\frac{\mathcal{B}_T}{1-\gamma}\right)\log\left(\frac{1}{1-\gamma}\right) \tag{4.8}
$$

where

$$
B(\gamma) = \frac{16(1-\gamma)\xi}{F(\gamma)}\frac{\lceil T(1-\gamma)\rceil}{T(1-\gamma)} + \frac{2\left\lceil -\log(1-\gamma)/\log(1+4\sqrt{1-1/2\xi})\right\rceil}{-\log(1-\gamma)(1-\gamma^{1/(1-\gamma)})} \tag{4.9}
$$

and

$$
F(\gamma) = \gamma^{1/(1-\gamma)}\left(\Delta_{\mu_T}(i)\sqrt{1-\gamma} - 4l\sqrt{-\xi\log(1-\gamma)}\right)^2 \tag{4.10}
$$

and

$$
C(\gamma) = \frac{(\gamma-1)\log((1-\gamma)\xi\log(n_K(\gamma))}{\log(1-\gamma)\log(\gamma)} \tag{4.11}
$$

Proof. This proof has been adapted from [15]'s analysis of discounted-UCB and [29]'s analysis of incentivized exploration with UCB-1.

We upper bound the number of times the suboptimal arm i is played as follows.

$$\widehat{N}_T(i) = 1 + \sum_{t=K+1}^{T} \mathbb{1}_{\{I_t = i \neq i_t^*\}} \tag{4.12}$$

which can be rewritten as

$$\widehat{N}_T(i) = 1 + \sum_{t=K+1}^{T} \mathbb{1}_{\{I_t = i \neq i_t^*; N_t(\gamma,i) < A(\gamma)\}} + \sum_{t=K+1}^{T} \mathbb{1}_{\{I_t = i \neq i_t^*; N_t(\gamma,i) \geq A(\gamma)\}} \tag{4.13}$$

where

$$A(\gamma) = \frac{16(1-\gamma)\xi \log(n_t(\gamma))}{\left(\Delta\sqrt{1-\gamma} - 4l\sqrt{(1-\gamma)\xi \log(n_t(\gamma))}\right)^2}. \tag{4.14}$$

The next few steps directly follow from [15]'s analysis for the same definitions of $D(\gamma)$ and $\mathcal{T}(\gamma)$. We can bound $\widehat{N}_T(i)$ by:

$$\widehat{N}_T(i) \leq 1 + \lceil T(1-\gamma) \rceil A(\gamma)\gamma^{-1/(1-\gamma)} + \mathcal{B}_T D(\gamma) + \sum_{t \in \mathcal{T}(\gamma)} \mathbb{1}_{\{I_t = i \neq i_t^*; N_t(\gamma,i) \geq A(\gamma)\}}. \tag{4.15}$$

Now, for $t \in \mathcal{T}$ the event $E : \{I_t = i \neq i_t^*; N_t(\gamma, i) \geq A(\gamma)\}$ will occur when the following inequality holds

$$\mathcal{Z} : \tilde{\mu}_t(\gamma, i) + c_t(\gamma, i) > \tilde{\mu}_t(\gamma, i^*) + c_t(\gamma, i^*). \tag{4.16}$$

Expanding the inequality using the definitions of $\hat{\mu}_t(\gamma, i)$, $\tilde{\mu}_t(\gamma, i)$ and lemma 4.1.2, we get the following:

$$\mathcal{Z} : \hat{\mu}_t(\gamma, i) + \frac{D_t(\gamma, i)}{\mathcal{N}_t(\gamma, i)} + c_t(\gamma, i) > \hat{\mu}_t(\gamma, i^*) + \frac{D_t(\gamma, i^*)}{\mathcal{N}_t(\gamma, i^*)} + c_t(\gamma, i^*). \tag{4.17}$$

Therefore, the upper bound for the difference between the expected reward of the optimal arm i^* and the current arm i is

$$\begin{aligned} \hat{\mu}_t(\gamma, i^*) - \hat{\mu}_t(\gamma, i) &< \frac{D_t(\gamma, i)}{\mathcal{N}_t(\gamma, i)} + c_t(\gamma, i) \\ &= 2\sqrt{\xi \log(n_t(\gamma))} \left(\frac{l}{\sqrt{1-\gamma}} + \frac{1}{\sqrt{N_t(\gamma, i)}} \right). \end{aligned} \tag{4.18}$$

Now, we can decompose E as the following, as for $\mathcal{Z}$ to happen, at least one of the events E_t^i has to occur:

$$\therefore \{I_t = i \neq i_t^*; N_t(\gamma, i) \geq A(\gamma)\} \subseteq E_t^1 \cup E_t^2 \cup E_t^3, \tag{4.19}$$

where

$$E_t^1 = \{\tilde{\mu}_t(\gamma, i) > \mu_t(\gamma, i) + c_t(\gamma, i)\}, \tag{4.20}$$

$$E_t^2 = \{\tilde{\mu}_t(\gamma, i^*) < \mu_t(\gamma, i^*) - c_t(\gamma, i^*)\}, \tag{4.21}$$

$$E_t^3 = \left\{\hat{\mu}_t(\gamma, i^*) - \hat{\mu}_t(\gamma, i) < 2\sqrt{\xi \log(n_t(\gamma))}\left(\frac{l}{\sqrt{1-\gamma}} + \frac{1}{\sqrt{N_t(\gamma, i)}}\right)\right\}. \tag{4.22}$$

E_t^1 is when the algorithm overestimates the average reward of arm i, E_t^2 when the algorithm underestimates the average reward of the best arm i^*, and E_t^3 is when the expected rewards for both the arms i and i^* are too close.

By union bound we have $\mathbb{P}[E] \leq \sum_i \mathbb{P}[E_t^i]$. However, for the choice of $A(\gamma)$, E_t^3 never occurs, as

$$\begin{aligned}
\hat{\mu}_t(\gamma, i^*) - \hat{\mu}_t(\gamma, i) &< 2\sqrt{\xi \log(n_t(\gamma))}\left(\frac{l}{\sqrt{1-\gamma}} + \frac{1}{\sqrt{N_t(\gamma, i)}}\right) \\
&\leq 2\sqrt{\xi \log(n_t(\gamma))}\left(\frac{l}{\sqrt{1-\gamma}} + \frac{1}{\sqrt{A(\gamma)}}\right) \\
&= \frac{\Delta_{\mu_T}(i)}{2}
\end{aligned}$$

Since the changing of expected reward values would make the estimates of them *biased*, we can't use the Hoeffding type bounds for E_t^1 and E_t^2. Therefore, we can use the results from [15]'s

analysis for E_t^1 and E_t^2, by using their novel tool to handle the same.

We have $\mathbb{P}\left[E_t^3\right] = 0$ and from [15] we get

$$\mathbb{P}\left[E_t^1\right] = \mathbb{P}\left[E_t^2\right] \leq \left\lceil \frac{\log(n_t(\gamma))}{\log(1+\eta)} \right\rceil n_t(\gamma)^{-2\xi\left(1-\frac{\eta^2}{16}\right)} \tag{4.23}$$

We finally have the bound, by taking $\xi > 1/2$ and $\eta = 4\sqrt{1-(1/2\xi)}$, so as to make $2\xi(1 = \eta^2/16) = 1$:

$$\mathbb{E}\left[\widehat{N}_T(i)\right] \leq 1 + \lceil T(1-\gamma)\rceil A(\gamma)\gamma^{-1/(1-\gamma)} + \mathcal{B}_T D(\gamma) + Y$$

where

$$Y = \frac{1}{1-\gamma} + \left\lceil \frac{\log\left(\frac{1}{1-\gamma}\right)}{\log(1+4\sqrt{1-(1/2\xi)})} \right\rceil \frac{T(1-\gamma)}{1-\gamma^{(1/(1-\gamma))}} \tag{4.24}$$

We obtain the statement of the theorem by substituting the values of $A(\gamma)$, $D(\gamma)$ and $n_t(\gamma)$.

□

Corollary 4.1.4.1 (Algorithm 10 + D-UCB Regret Bound)**.** *If the horizon T and the number of breakpoints $\mathcal{B}_T$ are known in advance, the discount factor γ can be approximately chosen to minimize the RHS from Theorem 4.1.4. Taking $\gamma = 1 - \eta \cdot \sqrt{\mathcal{B}_T/T}$, for some $\eta > 0$ we get the regret, some $\tilde{\eta} > 0$, as*

$$\mathbb{E}\left[\widehat{N}_T(i)\right] \leq \tilde{\eta} \cdot \sqrt{T\mathcal{B}_T}\log(T) \tag{4.25}$$

4.1.1.2 Sliding-Window UCB with Algorithm 10

Let the total number of breakpoints before time T be $\mathcal{B}_T$ and let $\widehat{N}_T(i)$ denote the number of times arm i was played when it was not the best arm during the first T rounds. Then,

$$\widehat{N}_T(i) = \sum_{t=1}^{T} \mathbb{1}_{\{I_t = i \neq i^*\}} \tag{4.26}$$

Let $\tilde{\mu}_i(\tau, t)$ denote the estimated drifted average of the expected rewards of the arm i, and $\hat{\mu}_i(\tau, t)$ denote the pure estimated average of the expected reward $\mu_t(i) = \mathbb{E}[X_i(t)]$

We denote $\Delta_{\mu_T}(i)$ as the minimum of the difference between the expected reward of the best arm $\mu_t(i^*)$ and the expected reward of the ith arm $\mu_t(i)$ for $\forall t \in [1, T]$ when i is not the optimal arm.

$$\Delta_{\mu_T}(i) = \min_{t\in[1,T];\ i\neq i^*} \{\mu_t(i^*) - \mu_t(i)\} \tag{4.27}$$

Lemma 4.1.5. *The total discounted reward drift of an arm i till time t, denoted by $D_i(\tau, t)$ is upper bounded by $\mathcal{N}_t(i) l \sqrt{\xi \log(\min(t, \tau))}$, where $l = \max_t l_t$.*

Proof. The principal has to provide compensation when both of the mentioned inequalities hold:

$$\tilde{\mu}_{G_t}(\tau, t) \geq \tilde{\mu}_{I_t}(\tau, t) \tag{4.28}$$

$$\tilde{\mu}_{G_t}(\tau, t) + c_t(\tau, G_t) \geq \tilde{\mu}_{I_t}(\tau, t) + c_t(\tau, I_t) \tag{4.29}$$

The two inequalities imply $\tilde{\mu}_{G_t}(\tau, t) - \tilde{\mu}_{I_t}(\tau, t) \leq c_t(\tau, I_t) - c_t(\tau, G_t)$. Using assumption 4.1.1, we have the following result.

$$\begin{aligned}
\tau &\leq l_t \left(c_t(\tau, I_t) - c_t(\tau, G_t)\right) \\
&\leq l_t \left(\sqrt{\frac{\xi \log(\min(t, \tau))}{N_t(\tau, I_t)}} - \sqrt{\frac{\xi \log(\min(t, \tau))}{N_t(\tau, G_t)}} \right) \\
&\leq l_t \sqrt{\frac{\xi \log(\min(t, \tau))}{N_t(\tau, I_t)}}
\end{aligned} \tag{4.30}$$

Now, using the value of τ, the total drift is found in the following way:

$$
\begin{aligned}
D_i(\tau,t) &= \sum_{s=1}^{t} \mathbb{1}_{\{I_s=i\}}\delta_s \\
&= \sum_{s=1}^{t} \mathbb{1}_{\{I_s=i\}} l_s \sqrt{\frac{\xi \log(\min(t,\tau))}{N_s(\tau,i)}} \\
&\leq l\sqrt{\xi \log(\min(t,\tau))} \sum_{s'=1}^{\mathcal{N}_t(i)} \frac{1}{\sqrt{N_{s'}(\tau,i)}} \\
&\leq \mathcal{N}_t(i) l \sqrt{\xi \log(\min(t,\tau))} \quad (\text{Since } N_{s'}(\tau,i) \geq 1)
\end{aligned}
\tag{4.31}
$$

□

The following theorem is for the regret when Sliding-Window-UCB is used as the principal's algorithm within Algorithm 10.

Theorem 4.1.6. *Let $\xi > 1/2$. For any integer $\tau > 0$ and any arm $i \in \mathcal{K}$,*

$$\mathbb{E}\left[\widehat{N}_T(i)\right] \leq C(\tau)\frac{T\log(\tau)}{\tau} + \tau\mathcal{B}_T + \log^2(\tau)$$

where

$$C(\tau) = \frac{\left(l\sqrt{\tau}+1\right)^2 \xi}{\left(\Delta_{\mu_T}(i)\right)^2}\frac{\lceil T/\tau \rceil}{T/\tau} + \frac{2}{\log(\tau)}\left\lceil \frac{\log(\tau)}{\log(1+4\sqrt{1-(2\xi)^{-1}})} \right\rceil$$

Proof. This proof has been adapted from [15]'s analysis of sliding-window-UCB and [29]'s of incentivized settings for UCB-1 respectively.

We upper bound the number of times the suboptimal arm i is played as follows:

$$\widehat{N}_T(i) = 1 + \sum_{t=K+1}^{T} \mathbb{1}_{\{I_t=i\neq i_t^*\}} \tag{4.32}$$

Which can be expanded to

$$
\begin{aligned}
\widehat{N}_T(i) = 1 &+ \sum_{t=K+1}^{T} \mathbb{1}_{\{I_t=i\neq i_t^*; N_t(\tau,i)<A(\tau)\}} \\
&+ \sum_{t=1}^{T} \mathbb{1}_{\{I_t=i\neq i_t^*; N_t(\tau,i)\geq A(\tau)\}}
\end{aligned}
\tag{4.33}
$$

where

$$A(\tau) = \frac{\left(l\sqrt{\tau}+1\right)^2 \xi \log(\tau)}{\left(\Delta_{\mu_T(i)}\right)^2} \tag{4.34}$$

The next few steps directly follow from [15]'s analysis, for the same definitions of $\mathcal{T}(\tau)$, and we can bound $\widehat{N}_T(i)$ by:

$$\widehat{N}_T(i) \leq 1 + \lceil T/\tau \rceil A(\tau) + \tau \mathcal{B}_T + \sum_{t \in \mathcal{T}(\tau)} \mathbb{1}_{\{I_t = i \neq i_t^*; N_t(\tau,i) \geq A(\tau)\}} \tag{4.35}$$

Now, for $t \in \mathcal{T}(\tau)$ the event $E : \{I_t = i \neq i_t^*; N_t(\tau, i) \geq A(\tau)\}$ will occur when the following inequality holds,

$$\mathcal{Z} : \tilde{\mu}_i(\tau, t) + c_t(\tau, i) > \tilde{\mu}_{i^*}(\tau, t) + c_t(\tau, i^*) \tag{4.36}$$

Expanding the inequality using the definitions of $\hat{\mu}_t(\tau, i)$, $\tilde{\mu}_t(\tau, i)$ and lemma 4.1.5 we get the following:

$$\mathcal{Z} : \hat{\mu}_i(\tau, t) + \frac{D_i(\tau, t)}{\mathcal{N}_t(\tau, i)} + c_t(\tau, i) > \hat{\mu}_{i^*}(\tau, t) + \frac{D_{i^*}(\tau, t)}{\mathcal{N}_t(\tau, i^*)} + c_t(\tau, i^*) \tag{4.37}$$

Therefore, the upper bound for the difference between the expected reward of the optimal arm i^* and the current arm i is

$$\begin{aligned}
\hat{\mu}_{i^*}(\tau, t) - \hat{\mu}_i(\tau, t) &< \frac{D_i(\tau, t)}{\mathcal{N}_t(\tau, i)} + c_t(\tau, i) \\
&= l\sqrt{\xi \log(\min(t, \tau))} + \sqrt{\frac{\xi \log(\min(t, \tau))}{N_t(\tau, i)}} \\
&= \left(l\sqrt{N_t(\tau, i)} + 1\right) \sqrt{\frac{\xi \log(\min(t, \tau))}{N_t(\tau, i)}} \\
&\leq \left(l\sqrt{\tau} + 1\right) \sqrt{\frac{\xi \log(\min(t, \tau))}{N_t(\tau, i)}}
\end{aligned} \tag{4.38}$$

Now, we can decompose E as the following, for $\mathcal{Z}$ to happen, at least one of the events E_t^i has to occur.

$$\therefore \{I_t = i \neq i_t^*; N_t(\tau, i) \geq A(\tau)\} \subseteq E_t^1 \cup E_t^2 \cup E_t^3 \tag{4.39}$$

Where

$$E_t^1 = \{\tilde{\mu}_t(\tau, i) > \mu_t(\tau, i) + c_t(\tau, i)\} \tag{4.40}$$

$$E_t^2 = \{\tilde{\mu}_t(\tau, i^*) < \mu_t(\tau, i^*) - c_t(\tau, i^*)\} \tag{4.41}$$

$$E_t^3 = \left\{\hat{\mu}_{\tau,i^*}(t) - \hat{\mu}_i(\tau, t) < \left(l\sqrt{\tau} + 1\right)\sqrt{\frac{\xi \log(\min(t, \tau))}{N_t(\tau, i)}}\right\} \tag{4.42}$$

E_t^1 is when the algorithm is overestimating the average reward of arm i, E_t^2 when the algorithm is underestimating the average reward of the best arm i^*, and the E_t^3 is when the expected rewards for both the arms i and i^* are too close.

By union bound we have $\mathbb{P}[E] \leq \sum_i \mathbb{P}[E_t^i]$. However, for the choice of $A(\tau)$, E_t^3 never occurs as

$$\left(l\sqrt{\tau} + 1\right)\sqrt{\frac{\xi \log(\min(t, \tau))}{N_t(\tau, i)}} \leq \left(l\sqrt{\tau} + 1\right)\sqrt{\frac{\xi \log(\tau))}{A(\tau)}} = \frac{\Delta_{\mu_T}(i)}{2} \tag{4.43}$$

For $t \in \mathcal{T}(\tau)$, the bias will vanish because the rewards won't change. We have $\mathbb{P}\left[E_t^3\right] = 0$ and from [15]

$$\mathbb{P}\left[E_t^1\right] = \mathbb{P}\left[E_t^2\right] \leq \left\lceil \frac{\log(\min(t, \tau))}{\log(1 + \eta)} \right\rceil \min(t, \tau)^{-2\xi\left(1 - \frac{\eta^2}{16}\right)} \tag{4.44}$$

We finally have the bound, by taking $\xi > 1/2$ and $\eta = 4\sqrt{1 - (1/2\xi)}$, so as to make $2\xi(1 = \eta^2/16) = 1$:

$$\mathbb{E}\left[\widehat{N}_T(i)\right] \leq \underbrace{1 + \lceil T/\tau \rceil A(\tau) + \tau \mathcal{B}_T}_{M} + \underbrace{2 \sum_{t=1}^{T} \frac{\left\lceil \frac{\log(\min(t,\tau))}{\log(1+\eta)} \right\rceil}{\min(t,\tau)}}_{N} \tag{4.45}$$

Substituting $A(\tau)$ in M and expanding N we upper bound $\mathbb{E}\left[\widehat{N}_T(i)\right]$ by

$$1 + \lceil T/\tau \rceil \frac{(l\sqrt{\tau}+1)^2 \, \xi \log(\tau)}{(\Delta_{\mu_T(i)})^2} + \frac{2T}{\tau} \left\lceil \frac{\log(\tau)}{\log(1+\eta)} \right\rceil + \tau \mathcal{B}_T + \log^2(\tau) \tag{4.46}$$

□

Corollary 4.1.6.1 (Algorithm 10 + SW-UCB Regret Bound)**.** *If the horizon time T is known in advance, the window size τ can be chosen to minimize the RHS in the equation from Theorem 4.1.6. Taking $\tau = \eta \cdot \sqrt{T \log(T)/\mathcal{B}_T}$, we get the regret, for some $\tilde{\eta} > 0$, as*

$$\mathbb{E}\left[\widehat{N}_T(i)\right] \leq \tilde{\eta} \cdot l^2 \sqrt{\mathcal{B}_T T \log(T)} \tag{4.47}$$

Remark 2. *The lower bound of the regret for any algorithm scheme for the piecewise stationary environment is $\Omega(\sqrt{T})$ as shown by [15]. There is an alternate proof presented in the section 2.2.4. Therefore, our algorithm scheme (Algorithm 10) with both D-UCB and SW-UCB is optimal up to some powers of $\log T$.*

Remark 3. *For Algorithm 10 with D-UCB, the regret is proportional to $O(1/(k-l)^2)$ for some $k > 0$, whereas, SW-UCB is proportional to $O(l^2)$.*

4.1.2 Compensation

4.1.2.1 Discounted UCB with Algorithm 10

For finding the compensation, we can add up the compensation of all the arms over the entire horizon. We have to consider the conditions for providing compensation and the amount provided.

Theorem 4.1.7. *Let $\xi > 1/2$. For any $\gamma \in (0, 1]$ and for arm $i \in \mathcal{K}$, The overall compensation over the horizon time T is*

$$\mathbb{E}\left[C_T(i)\right] \leq (\mathcal{B}_T + 1)\sqrt{\xi \log(1/1-\gamma)}\mathbb{E}\left[\widehat{N}_T(i)\right]$$

Proof. Compensation has to be given when,

$$\tilde{\mu}_t(\gamma, i) > \tilde{\mu}_t(\gamma, i^*) \tag{4.48}$$

and

$$\tilde{\mu}_t(\gamma, i) + \sqrt{\frac{\xi \log(n_t(\gamma))}{N_t(\gamma, i)}} < \tilde{\mu}_t(\gamma, i^*) + \sqrt{\frac{\xi \log(n_t(\gamma))}{N_t(\gamma, i^*)}} \tag{4.49}$$

Therefore, the compensation is given to the agent even when the agent pulls the optimal arm and $N_t(\gamma, i^*) < N_t(\gamma, i)$. Which means that the average number of times a player is compensated for pulling the best arm is upper bounded by $M = \max_{i \neq i^*} \mathbb{E}[\widehat{N}_T(i)]$.

Now, since the rewards can change with a breakpoint, so can the best arm. We need to consider each interval between breakpoints and find the maximum equivalent of M.

Let t_b be the timestamp at which the bth breakpoint occurs, $\forall b \in [1, \mathcal{B}_T]$. Let V and S (see Figure 4.1) be the set of intervals between breakpoints and the sizes of each intervals respectively. For convenience, consider $t_0 = 0$

$$V = \{(t_{b-1}, t_b) \ : \ \forall b \in [1, \mathcal{B}_T]\} \tag{4.50}$$

$$S = \{s_v \ : \ s_v = |t_{b-1} - t_b| \ \forall v \in V; \ \forall b \in [1, \mathcal{B}_T]\} \tag{4.51}$$

Let $v^* \in V$ be the interval in which the principal pays the compensation maximum number of times M_V, when the player plays the interval's best arm i_v^*. With a slight abuse of notation, let $\widehat{N}_{s_v}(i)$ be the number of times the suboptimal arm i was played in the interval v. Therefore,

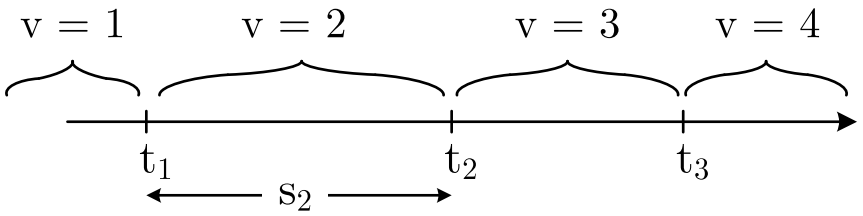

Figure 4.1: Time intervals representing sets V and S for compensation analysis.

$$v^* = \arg\max_{v \in V} \left[\max_{i \in \mathcal{K}: i \neq i_v^*} \mathbb{E}\left[\widehat{N}_{s_v}(i)\right] \right] \tag{4.52}$$

Taking i_* to be the best arm in the interval v^*, We get,

$$\begin{aligned} M_V &\leq \mathcal{B}_T \max_{i \neq i_*} \mathbb{E}\left[\widehat{N}_{s_v}(i)\right] \\ &\leq \mathcal{B}_T \max_{i \neq i_*} \mathbb{E}\left[\widehat{N}_T(i)\right] \end{aligned} \tag{4.53}$$

We get the total expected compensation as

$$\begin{aligned} \mathbb{E}\left[C_T\right] &\leq \sum_{i=1}^{K} \sum_{j=1}^{\mathbb{E}\left[\widehat{N}_T(i)\right]} \sqrt{\frac{\xi \log(n_t(\gamma))}{N_j(\gamma, i)}} \\ &\leq \sum_{j=1}^{M_V} \sqrt{\frac{\xi \log(n_t(\gamma))}{N_j(\gamma, i))}} + \sum_{i=1}^{K-1} \sum_{j=1}^{\mathbb{E}\left[\widehat{N}_T(i)\right]} \sqrt{\frac{\xi \log(_t(\gamma))}{N_j(\gamma, i))}} \\ &\leq \sqrt{\xi \log(n_t(\gamma))} \left(\sum_{j=1}^{\mathcal{B}_T \max_{i \neq i^*} \mathbb{E}\left[\widehat{N}_T(i)\right]} 1 + \sum_{i=1}^{K-1} \sum_{j=1}^{\mathbb{E}\left[\widehat{N}_T(i)\right]} 1 \right) \\ &\leq \sqrt{\xi \log(n_t(\gamma))} \left(\sum_{i=1}^{K} \sum_{j=1}^{\mathcal{B}_T \mathbb{E}\left[\widehat{N}_T(i)\right]} 1 + \sum_{i=1}^{K-1} \sum_{j=1}^{\mathbb{E}\left[\widehat{N}_T(i)\right]} 1 \right) \\ &\leq \sqrt{\xi \log(n_t(\gamma))} \left(\sum_{i=1}^{K} \sum_{j=1}^{(\mathcal{B}_T+1)\mathbb{E}\left[\widehat{N}_T(i)\right]} 1 \right) \\ &\leq \sqrt{\xi \log(n_t(\gamma))} \left(\sum_{i=1}^{K} (\mathcal{B}_T + 1)\, \mathbb{E}\left[\widehat{N}_T(i)\right] \right) \end{aligned} \tag{4.54}$$

□

Corollary 4.1.7.1 (Algorithm 10 + D-UCB Compensation Bound)**.** *For the same values of ξ and γ from Theorem 4.1.7, we get the compensation contribution of arm i, for some $\eta > 0$, upper bounded by*

$$\mathbb{E}\left[C_T(i)\right] \leq \eta \cdot \mathcal{B}_T^{3/2}\sqrt{T}(\log(T))^{3/2} \tag{4.55}$$

4.1.2.2 Sliding-Window with Algorithm 10

For SW-UCB, a compensation is given when $\tilde{\mu}_i(\tau,t) > \tilde{\mu}_{i^*}(\tau,t)$ and $\tilde{\mu}_i(\tau,t) + c_t(\tau,i) < \tilde{\mu}_{i^*}(\tau,t) + c_t(\tau,i^*)$, which means that the compensation is given to the agent even when the agent pulls the optimal arm and $N_t(\tau,i^*) < N_t(\tau,i)$. Which means that the average number of times a player is compensated for pulling the best arm is upper bounded by $\max_{i\neq i^*}\mathbb{E}[\widehat{N}_T(i)]$.

Theorem 4.1.8. *Let $\xi > 1/2$. For any integer τ and for arm $i \in \mathcal{K}$, The overall compensation over the horizon time T is*

$$\mathbb{E}\left[C_T(i)\right] \leq \sqrt{\xi\tau\log(\tau)} + \sqrt{\frac{\xi\log(\tau)}{\tau}}\left(\mathcal{B}_T + 1\right)\mathbb{E}\left[\widehat{N}_T(i)\right]$$

Proof. Compensation has to be given when,

$$\tilde{\mu}_i(\tau,t) > \tilde{\mu}_{i^*}(\tau,t) \tag{4.56}$$

and

$$\tilde{\mu}_i(\tau,t) + \sqrt{\frac{\xi\log(\min(t,\tau))}{N_t(\tau,i)}} < \tilde{\mu}_{i^*}(\tau,t) + \sqrt{\frac{\xi\log(\min(t,\tau))}{N_t(\tau,i^*)}} \tag{4.57}$$

Therefore, the compensation is given to the agent even when the agent pulls the optimal arm and $N_t(\tau,i^*) < N_t(\tau,i)$. Which means that the average number of times a player is compensated for pulling the best arm is upper bounded by $M = \max_{i\neq i^*}\mathbb{E}[\widehat{N}_T(i)]$.

Now, since the rewards can change with a breakpoint, so can the best arm. We need to consider each interval between breakpoints and find the maximum equivalent of M.

Let t_b be the timestamp at which the bth breakpoint occurs, $\forall b \in [1, \mathcal{B}_T]$. Let V and S be the set of intervals between breakpoints and the sizes of each interval respectively.

$$V = \{(t_{b-1}, t_b) \ : \ \forall b \in [1, \mathcal{B}_T]\} \tag{4.58}$$

$$S = \{s_v \ : \ s_v = |t_{b-1} - t_b| \ \forall v \in V; \ \forall b \in [1, \mathcal{B}_T]\} \tag{4.59}$$

Let $v^* \in V$ be the interval in which the principal pays the compensation maximum number of times M_V, when the player plays the interval's best arm i_v^*. Therefore,

$$v^* = \arg\max_{v \in V} \left[\max_{i \in \mathcal{K}: i \neq i_v^*} \mathbb{E}\left[\tilde{N}_{s_v}(i)\right] \right] \tag{4.60}$$

Taking i_* to be the best arm in the interval v^*, We get,

$$\begin{aligned} M_V &\leq \mathcal{B}_T \max_{i \neq i_*} \mathbb{E}\left[\tilde{N}_{s_v}(i)\right] \\ &\leq \mathcal{B}_T \max_{i \neq i_*} \mathbb{E}\left[\widehat{N}_T(i)\right] \end{aligned} \tag{4.61}$$

Now, noting that $\sum_{i=1}^{n} 1/\sqrt{\min(\tau, i)} \leq (\sqrt{\tau} + (n/\sqrt{\tau}))$, we get the total expected compensation as

$$
\begin{aligned}
\mathbb{E}\left[C_T\right] &\leq \sum_{i=1}^{K} \sum_{j=1}^{\mathbb{E}\left[\widehat{N}_T(i)\right]} \sqrt{\frac{\xi \log(\tau)}{N_j(\tau,i))}} \\
&\leq \underbrace{\sum_{j=1}^{M_V} \sqrt{\frac{\xi \log(\tau)}{N_j(\tau,i))}}}_{X} + \underbrace{\sum_{i=1}^{K-1} \sum_{j=1}^{\mathbb{E}\left[\widehat{N}_T(i)\right]} \sqrt{\frac{\xi \log(\tau)}{N_j(\tau,i))}}}_{Y} \\
&\leq \sqrt{\xi \log(\tau)} \left[\sum_{i=1}^{K} \sum_{j=1}^{(\mathcal{B}_T+1)\mathbb{E}\left[\widehat{N}_T(i)\right]} \frac{1}{\sqrt{\min(\tau, j)}} \right] \\
&\leq \sqrt{\xi \log(\tau)} \left[\sum_{i=1}^{K} \left(\sqrt{\tau} + \frac{(\mathcal{B}_T + 1)\, \mathbb{E}\left[\widehat{N}_T(i)\right]}{\sqrt{\tau}} \right) \right] \\
&= \sum_{i=1}^{K} \left(\sqrt{\xi \tau \log(\tau)} + \sqrt{\frac{\xi \log(\tau)}{\tau}} \left(\mathcal{B}_T + 1\right) \mathbb{E}\left[\widehat{N}_T(i)\right] \right)
\end{aligned}
\tag{4.62}
$$

Where X expands as the following

$$
\begin{aligned}
X &\leq \sqrt{\xi \log(\tau)} \sum_{j=1}^{\mathcal{B}_T \max_{i \neq i_*} \mathbb{E}\left[\widehat{N}_T(i)\right]} \frac{1}{\sqrt{\min(\tau, j)}} \\
&\leq \sqrt{\xi \log(\tau)} \sum_{i=1}^{K} \sum_{j=1}^{\mathcal{B}_T \mathbb{E}\left[\widehat{N}_T(i)\right]} \frac{1}{\sqrt{\min(\tau, j)}}
\end{aligned}
\tag{4.63}
$$

and Y as

$$
Y \leq \sqrt{\xi \log(\tau)} \sum_{i=1}^{K-1} \sum_{j=1}^{\mathbb{E}\left[\widehat{N}_T(i)\right]} \frac{1}{\sqrt{\min(\tau, j)}} \tag{4.64}
$$

□

Corollary 4.1.8.1 (Algorithm 10 + SW-UCB Compensation Bound)**.** *For the same values of ξ and τ from theorem 4.1.8, we get the compensation contribution of arm i, for some $\eta > 0$, upper bounded by*

$$
\mathbb{E}\left[C_T(i)\right] \leq \eta \cdot l^2 (\mathcal{B}_T)^{7/4} T^{1/4} (\log(T))^{3/4} \tag{4.65}
$$

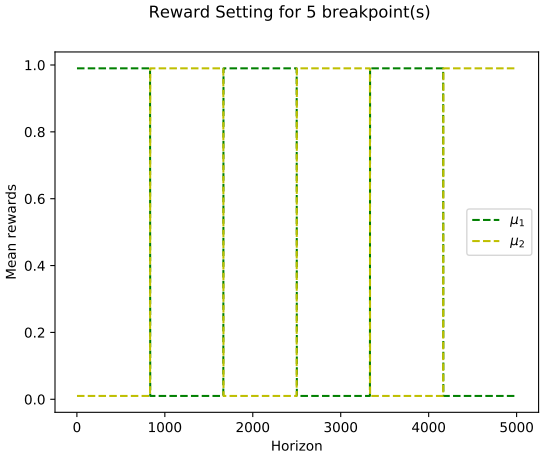

Figure 4.2: Mean rewards for piecewise-stationary setting with $\mathcal{B}_T = 5$ and 2 arms.

4.2 Simulation Results

For this environment, we have two arms $\mathcal{K} = \{1, 2\}$. The initial rewards are $\mu_1 = 0.99$ and $\mu_2 = 0.01$, which just flips(i.e swaps values) at every breakpoint. The breakpoints are the points $kT/p;\ \forall k \in [1, p-1]$ which divide the entire horizon into p equal parts, for some $p > 0$. So, if we want to divide the horizon T into 3 equal parts, we have the breakpoints at $\lfloor T/3 \rfloor$ and $\lfloor 2T/3 \rfloor$. Check figure 4.2 for $\mathcal{B}_T = 5$ with six equal (almost) parts of the horizon of $T = 5000$.

For experimenting with D-UCB, we have used $\gamma = 1 - (1/\gamma_C)\sqrt{\mathcal{B}_T/T}$, where we tune γ_C to minimize the regret. For SW-UCB, we have used $\tau = \lfloor \tau_C \sqrt{T \log(T)/\mathcal{B}_T} \rfloor$ and tuned τ_C for regret minimization. At each time step, the received reward is compared to the reward for the best arm at that time step and accumulated till the end. We average out the regret and compensation values at each step by running multiple repetitions. In this case, we have 100 repetitions.

Figures 4.3 and 4.4 has the performance of the algorithms discounted UCB and the sliding window UCB with Algorithm 1 with $T = 5000$. Both the algorithms clearly outperform the vanilla UCB1 with Algorithm 1 for the same horizon. The frequent changes force the UCB1 to make mistakes at the start of each breakpoint, as it considers the entire history, but D-UCB considers the decaying history and SW-UCB considers just a window adjusts quickly to change in the reward distribution of the arms, and causes lower regret.

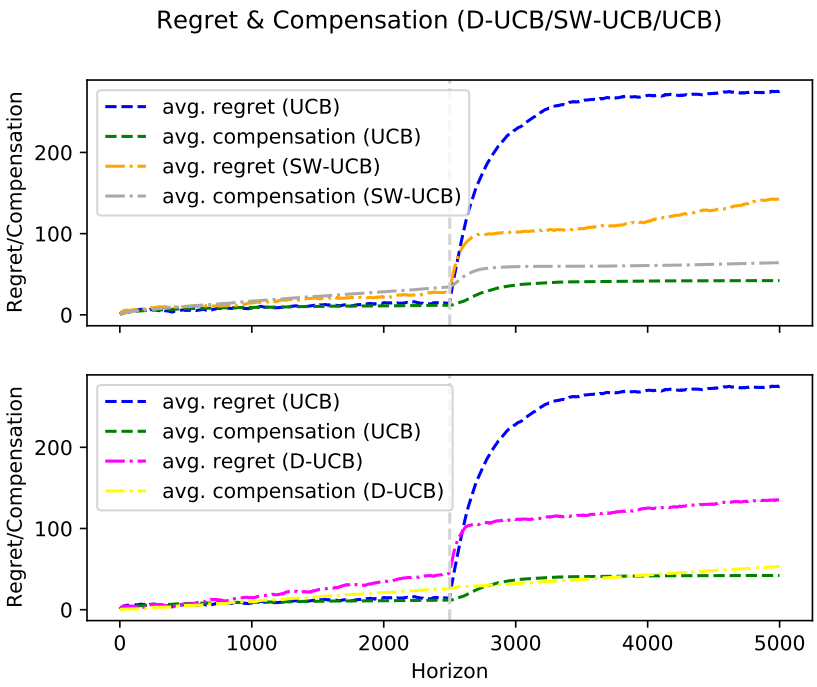

Figure 4.3: (Upper) Regret and Compensation performance of D-UCB with Algorithm 1 with $\gamma_C = 10$ (Below) Regret and Compensation performance of SW-UCB with Algorithm 1 with $\tau_C = 0.9$, both with $T = 5000$ and $\mathcal{B}_T = 1$

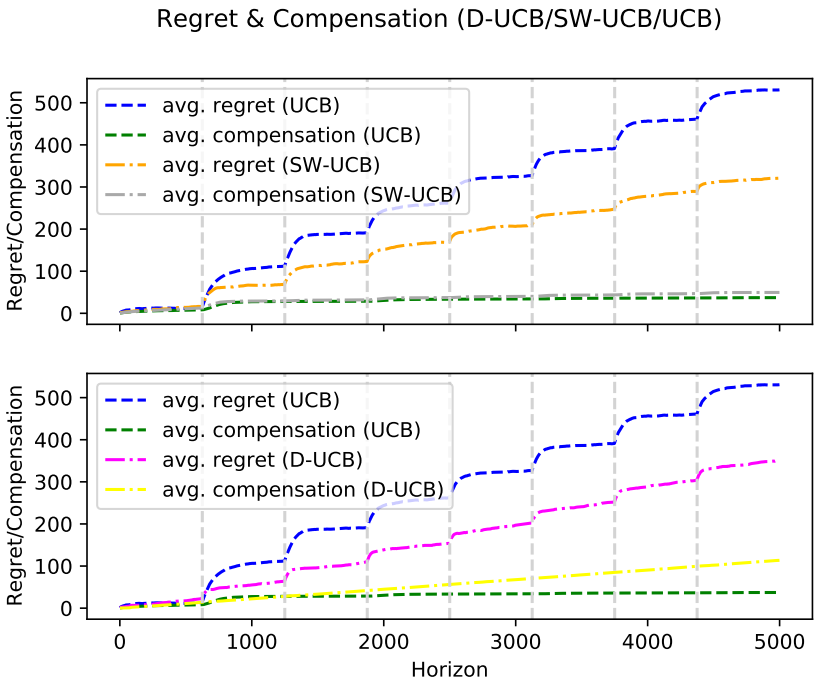

Figure 4.4: (Upper) Regret and Compensation performance of D-UCB with Algorithm 1 with $\gamma_C = 40$ (Below) Regret and Compensation performance of SW-UCB with Algorithm 1 with $\tau_C = 1$, both with $T = 5000$ and $\mathcal{B}_T = 1$

$\mathcal{B}_T$	γ_C	τ_C	R_U	R_S	R_D	C_U	C_S	C_D
1	15	1	275.2	135.1	142.7	42.1	53.2	64.2
2	10	1	364.2	203.5	205.7	42.5	70.7	92.3
3	15	1	430.4	239.5	247.1	41.6	81.2	82.8
4	15	0.95	394.8	264.1	259.7	41.4	95.1	89.2
5	10	1	423.7	288.9	302.4	39.8	100.8	112.3
6	25	1	481.8	330.1	279.1	38.5	107.9	67.6
7	30	0.95	484.2	339.0	299.7	38.6	117.1	59.2

Table 4.1: This is the data of the performance of SW-UCB + Algorithm 1 with varying number of breakpoints $\mathcal{B}_T$ and $l = 0.05$. The subscripts U, D, S stands for UCB1, D-UCB and SW-UCB respectively with R as the regret and C as the compensation values.

Table 4.1 shows the performance of SW-UCB and D-UCB with Algorithm 1 respectively with a varying number of breakpoints. The corresponding parameters τ_C and γ_C are also presented, which minimized the regrets. All the regret and compensation values are within the theoretical bounds. Besides, the regret is consistently lower than the UCB1 counterpart for both D-UCB and SW-UCB, as all the parameters $\gamma, \gamma_C, \tau, \tau_C$ are tuned to minimize regret. The values considered are $\tau_C = [10, 20, 30, 40]$ and $\gamma_C = [0.9, 0.95, 1, 2]$ through experimentation.

One can notice that in both cases the regret is growing in the order of $O(\sqrt{\mathcal{B}_T})$ with a varying number of breakpoints, as the theoretical analysis suggests. For compensation, we can notice that SW-UCB increases more rapidly than D-UCB, which explains the higher sensitivity of SW-UCB for the number of breakpoints which is in the order $O(\mathcal{B}_T^{7/4})$ compared to $O(\mathcal{B}_T^{3/2})$.

Chapter 5

Continuously-Changing Environment

We apply the restarting mechanism introduced in [5] in this section, to the continuously-changing environment. Since this is a passive approach to non-stationarity, the mechanism won't look for the changing points but will restart the bandit algorithm (which is used as a submodule BanditAlg) every $\tau \in [1, T]$ time epochs (or timesteps). We select the τ to minimize the upper bound on overall regret. The process is described in Algorithm 11 in Section 3.3

We will use the UCB1, ε-greedy and Thompson sampling for BanditAlg algorithm under reward drift.

For this environmental model, the regret upper bound for all the policies with Algorithm 11 and the compensation for UCB1, under drifted reward achieve $\tilde{O}(T^{2/3})$. The compensation for ε-Greedy and Thompson Sampling are $\tilde{O}(T^{1/3})$.

5.1 Theoretical Results

In this section, we show the regrets and compensations for Algorithm 11 with UCB1, ε-greedy and Thompson sampling as the bandit algorithms under reward drift.

5.1.1 Regret

We define the regret $\mathcal{R}^{\pi}(\mathcal{V}, T)$ for any policy $\pi \in \mathcal{P}$, where $\mathcal{P}$ is the policy class in accordance with [5] Section 2, compared to a *dynamic oracle* as the worst case difference between the expected performance of pulling the *best arm* at each time epoch t and expected performance under the

policy π.

$$\mathcal{R}^{\pi}(\mathcal{V},T)=\sup_{\mu\in\mathcal{V}}\left\{\sum_{t=1}^{T}\mu_t^*-\mathbb{E}^{\pi}\left[\sum_{t=1}^{T}\mu_t^{\pi}\right]\right\} \tag{5.1}$$

where the expectation $\mathbb{E}^{\pi}[.]$ is taken with respect to the noisy rewards, as well as to the policy's actions.

Theorem 5.1.1. *Let π be any* BanditAlg *policy under reward drift with regret of $\lambda\sqrt{TK\log(T)}$ for some constant $\lambda>0$, and the batch size $\tau=(\lambda T/V_T)^{2/3}(K\log(T))^{1/3}$, for Algorithm 2. Then, for $T\geq 2, K\geq 2$ and $V_T\in[1/K,T/K]$, the total regret is*

$$\mathcal{R}^{\pi}(\mathcal{V},T)\leq 2\lambda^{1/3}\cdot V_T^{1/3}(K\log(T))^{1/3}T^{2/3}$$

Proof. We follow the proof structure from [5] First, we break the horizon into sequences of batches of size τ each and then analyse the performance gap between the single best action and the dynamic oracle in each batch. Then, we plug in the known performance of BanditAlg relative to the single best action, under reward drift. We sum them over the batches to establish the regret bound for Algorithm 11.

Let us fix $T\geq 1, K\geq 2$ and $V_T\in[1/K,T/K]$. Let π be BanditAlg policy under reward drift. Let $\tau\in\{1,...,T\}$ be the batch size, which we will choose later. We break the horizon into sequence of batches $\mathcal{T}_j;\forall j\in[1,m]$, where $m=\lceil T/\tau\rceil$, of size τ, except possibly the last one. We decompose the regret in the batch j, with $\gamma=\mathbb{E}\left[\max_{k\in\mathcal{K}}\left\{\sum_{t\in\mathcal{T}_j}X_t(k)\right\}\right]$ as:

$$\mathbb{E}\left[\sum_{t\in\mathcal{T}_j}(\mu_t^*-\mu_t^{\pi})\right]=\underbrace{\sum_{t\in\mathcal{T}_j}\mu_t^*-\gamma}_{J_1}+\underbrace{\gamma-\mathbb{E}\left[\sum_{t\in\mathcal{T}_j}\mu_t^{\pi}\right]}_{J_2} \tag{5.2}$$

The first component J_1 is the expected loss associated with using a single action over batch j. The second component J_2 is the expected regret relative to the best static action in batch j. From (6) in [5], we know that $J_1\leq 2V_j\tau$.

Considering J_2 being after all the regret of a policy with respect to the single best action,

within a batch, we can plug in the performance of BanditAlg under reward drift.

$$J_2 = \mathbb{E}\left[\max_{k\in\mathcal{K}}\left\{\sum_{t\in\mathcal{T}_j} X_t(k)\right\} - \mathbb{E}\left[\sum_{t\in\mathcal{T}_j}\mu_t^{\pi}\right]\right] \leq \lambda\sqrt{\tau K\log(\tau)} \tag{5.3}$$

The next step is to sum them over the entire horizon. There are $m = \lceil T/\tau \rceil$ batches. Therefore, the overall regret is:

$$\begin{aligned}
\mathcal{R}^{\pi}(\mathcal{V},T) &= \sup_{\mu\in\mathcal{V}}\left\{\sum_{t=1}^{T}\mu_t^{*} - \mathbb{E}^{\pi}\left[\sum_{t=1}^{T}\mu_t^{\pi}\right]\right\} \\
&\leq \sum_{j=1}^{m}\left(\lambda\sqrt{\tau K\log(\tau)} + 2V_j\tau\right) \\
&\leq \left(\frac{2T}{\tau}\right)\cdot\lambda\sqrt{\tau K\log(\tau)} + 2V_T\tau \text{ (since } m\geq 1) \\
&= 2T\cdot\lambda\sqrt{\frac{K\log(\tau)}{\tau}} + 2V_T\tau
\end{aligned} \tag{5.4}$$

Selecting $\tau = (\lambda T/V_T)^{2/3}\,(K\log(T))^{1/3}$, we get

$$\mathcal{R}^{\pi}(\mathcal{V},T) \leq 2\lambda^{1/3}\cdot V_T^{1/3}\,(K\log(T))^{1/3}\,T^{2/3} \tag{5.5}$$

This concludes the proof. □

Corollary 5.1.1.1. *If UCB1, ε-Greedy or Thompson Sampling is used as the* BanditAlg *policy under reward drift for Algorithm 11, then, from theorem 5.1.1 there exist some constant $\tilde{\lambda}$, such that*

$$\mathcal{R}^{UCB1}(\mathcal{V},T) \leq \tilde{\lambda}\cdot(l+1)^{1/3}\,V_T^{1/3}\,(K\log(T))^{1/3}\,T^{2/3} \tag{5.6}$$

$$\mathcal{R}^{\varepsilon G}(\mathcal{V},T) \leq \tilde{\lambda}\cdot l^{1/6}M^{-1/3}V_T^{1/3}\,(K\log(T))^{1/3}\,T^{2/3} \tag{5.7}$$

$$\mathcal{R}^{TS}(\mathcal{V},T) \leq \tilde{\lambda}\cdot l^{1/9}\underline{\Delta}^{-1/3}V_T^{1/3}\,(K\log(T))^{1/3}\,T^{2/3} \tag{5.8}$$

Proof. The regret for *UCB1* under reward drift, for a batch j, for some constant C, is

$$\mathcal{R}^{UCB1} \leq \sum_{i \in \mathcal{K}; \Delta_j(i) > 0} \frac{C\,(l+1)^2}{\Delta_j(i)} \log(\tau) \tag{5.9}$$

courtesy of [29]. A more general regret bound can be derived which is independent of the $\Delta_j(i); \forall i$. Adapting this from [38], we can fix some $\varepsilon \in (0, 1)$.

- The regret contributed by all the arms with $\Delta_j(i) > \varepsilon$ is at most $\frac{CK(l+1)^2 \log(\tau)}{\varepsilon}$.
- The regret contributed by all the arms with $\Delta_j(i) \leq \varepsilon$ is at most $\varepsilon \cdot \tau^{\beta}$.

Therefore, the total regret for this batch j is at most:

$$\mathcal{R}^{UCB1} \leq \varepsilon \cdot \tau^{\beta} + \frac{CK\,(l+1)^2 \log(\tau)}{\varepsilon} \tag{5.10}$$

For $\varepsilon = \sqrt{\frac{CK(l+1)^2 \log(\tau)}{\tau}}$ and $\beta = 1$, we have $\mathcal{R}^{UCB1} \leq \sqrt{C}\,(l+1)\sqrt{\tau K \log(\tau)}$. This result is almost optimal as it almost matches the minimax lower bound of any stochastic bandit algorithm: $O(\sqrt{\tau})$. Using Theorem 5.1, and substituting $\lambda = \sqrt{C}\,(l+1)$, we get the value of required $\mathcal{R}^{UCB1}$.

The regret for ε-Greedy under reward drift, for a batch j, for some constant C, is

$$\mathcal{R}^{\varepsilon G} \leq \frac{l \cdot C}{M^2} \sum_{i \in \mathcal{K}; \Delta_j(i) > 0} \frac{\log(\tau)}{(\Delta_j(i))^2} \tag{5.11}$$

courtesy of [29]. A more general regret bound can be derived which is independent of the $\Delta_j(i); \forall i$. Adapting this from [38], we can fix some $\varepsilon \in (0, 1)$.

- The regret contributed by all the arms with $\Delta_j(i) > \varepsilon$ is at most $\frac{KlC \log(\tau)}{M^2 \varepsilon^2}$.
- The regret contributed by all the arms with $\Delta_j(i) \leq \varepsilon$ is at most $\varepsilon \cdot \tau^{\beta}$.

Therefore, the total regret for this batch j is at most:

$$\mathcal{R}^{\varepsilon G} \leq \varepsilon \cdot \tau^{\beta} + \frac{KlC \log(\tau)}{M^2 \varepsilon^2} \tag{5.12}$$

For $\varepsilon = \left(\frac{KlC\log(\tau)}{M^2\tau}\right)^{1/3}$ and $\beta = 3/4$, we have

$$\mathcal{R}^{\varepsilon G} \leq \left(Cl/M^2\right)^{1/3} \left(K\log(\tau)\right)^{1/3} \sqrt{\tau} \leq \left(Cl/M^2\right)^{1/3} \sqrt{3\tau K\log(\tau)}$$

for $\tau \geq 1$. This result is almost optimal as it almost matches the minimax lower bound of any stochastic bandit algorithm: $O(\sqrt{\tau})$. Using theorem 5.1.1, and substituting $\lambda = \sqrt{3}\left(Cl/M^2\right)^{1/3}$, we get the value of required $\mathcal{R}^{\varepsilon G}$.

The analysis is the same as above for Thompson Sampling, by replacing M with $\underline{\Delta}$ (See Definition 2 in [29]), we conclude the proof.

□

Remark 4. *Theorem 5.1.1 is a general result, which considers all the Stochastic bandit policies which achieve $O(\log T)$ regret. As mentioned in section 3.3, the assumptions behind the above result are based on the separation of the mean rewards of the arms. The regret of $O(\lambda\sqrt{TK\log T})$ for UCB1 is found by relaxing the above assumption, by considering that the arms' mean rewards can be arbitrarily close. For ε-Greedy and Thompson Sampling achieves the same regret form with some additional assumptions mentioned in section 3.3.*

Remark 5. *The constant M in Corollary 5.1.1 is from Assumption 3.3.2, and $\underline{\Delta}$ is from [29] Definition 2.*

5.1.2 Compensation

We use the results of the compensation for each bandit algorithm under reward drift from [29] for each batch, and then multiply it by the number of batches.

Corollary 5.1.1.2. *If UCB1, ε-Greedy or Thompson Sampling is used as the* BanditAlg *policy under reward drift for Algorithm 11, then there exist some constant $\tilde{\lambda}$, such that, the total average compensation is*

$$\mathcal{C}^{UCB1}(\mathcal{V},T) \leq \tilde{\lambda}\cdot(l+1)^{-1/3}\,V_T^{1/3}\cdot(K\log(T))^{1/3}\,T^{2/3} \tag{5.13}$$

$$\mathcal{C}^{\varepsilon G}(\mathcal{V},T) \leq \tilde{\lambda}\cdot l^{7/9}M^{-5/9}\,(KV_T\log(T))^{2/3}\,T^{1/3} \tag{5.14}$$

$$\mathcal{C}^{TS}(\mathcal{V},T) \leq \tilde{\lambda}\cdot \underline{\Delta}^{-14/9}\cdot l^{5/9}\,(KV_T\log(T))^{2/3}\,T^{1/3} \tag{5.15}$$

Proof. From Theorem 2 in [29], we know that expected compensation for a batch j, would be $\mathbb{E}\left[\mathcal{C}_j^{\varepsilon G}\right] \leq \tilde{C}Kl\log(\tau)/M$ for some $\tilde{C}>0$. Summing the result over the whole horizon:

$$\begin{aligned}\mathcal{C}^{\varepsilon G} &\leq \left\lceil\frac{T}{\tau}\right\rceil\mathbb{E}\left[\mathcal{C}_j^{\varepsilon G}\right]\\ &\leq \frac{2T}{\tau}\cdot\frac{\tilde{C}Kl\log(T)}{M}\ (\text{since }\tau\geq 1)\end{aligned}$$

Substituting the value of τ from corollary 5.1.1, we get the result.

From Theorem 3 [29], we know that the expected compensation for a batch j, would be $\mathbb{E}\left[\mathcal{C}_j^{TS}\right] \leq \tilde{C}\cdot 2K(l+1)\log(\tau)/\underline{\Delta}^2$ for some $\tilde{C}>0$. The analysis is the same beyond this point as shown above with appropriate values.

From Theorem 1 [29], we know that the expected compensation for a batch j, would be $\mathbb{E}\left[\mathcal{C}_j^{UCB1}\right] \leq \tilde{C}\,(l+1)\cdot\sum_{i\neq i^*}\log(\tau)/\Delta_j(i)$ for some $\tilde{C}>0$.

From [38] method, used above that can be generalized to $\bar{C}\cdot\sqrt{K\tau\log(\tau)}$ for some constant $\bar{C}>0$. Summing the result over the entire horizon we get:

$$\begin{aligned}\mathcal{C}^{UCB1} &\leq \left\lceil\frac{T}{\tau}\right\rceil\mathbb{E}\left[\mathcal{C}_j^{UCB1}\right]\\ &\leq \bar{C}\cdot\frac{2T}{\tau}\cdot\sqrt{K\tau\log(\tau)}\end{aligned}$$

Substituting τ and λ subsequently, we get the desired expression. □

5.2 Simulation Results

For the numerical results, we consider instances where two arms are available: $\mathcal{K} = \{1, 2\}$. The reward associated with arm i at time t is $X_t(i)$, and has a bernoulli distribution with a changing expectation $\mu_t(i)$:

$$X_t(i) = \begin{cases} 1 & \text{with prob. } \mu_t(i) \\ 0 & \text{with prob. } 1 - \mu_t(i) \end{cases}$$

$\forall t \in [1, T]$ and for any pulled arm $i \in \mathcal{K}$. We have two evolution patterns for $\mu_t(i)$. Both follow the sinusoidal definitions from [5], check figure 5.1. The first setting has the following functions for $\mu_t(i)$.

$$\mu_t(1) = \frac{1}{2} + \frac{1}{2}\sin\left(\frac{V_T \pi t}{T}\right), \quad \mu_t(2) = \frac{1}{2} + \frac{1}{2}\sin\left(\frac{V_T \pi t}{T} + \pi\right)$$

$\forall t \in [1, T]$. The second setting has the following functions:

$$\mu_t(1) = \begin{cases} \frac{1}{2} + \frac{1}{2}\sin\left(\frac{3V_T \pi t}{T} + \pi\right) & \text{if } t < \frac{T}{3} \\ 0 & \text{otherwise} \end{cases}$$

$$\mu_t(2) = \begin{cases} \frac{1}{2} + \frac{1}{2}\sin\left(\frac{3V_T \pi t}{T} - \pi\right) & \text{if } t < \frac{T}{3} \\ 1 & \text{otherwise} \end{cases}$$

Both the settings are sinusoidal and have a variation budget $V_T = 3$. They describe different changing environments under the same variation budget. While in the first instance the variation budget is spent throughout the whole horizon, in the second one the same variation budget is spent only over the first third of the whole horizon.

At each epoch $t \in [1, T]$ the policy selects an arm $i \in \mathcal{K}$. Then the binary rewards are generated, and $X_t(i)$ is observed. The pointwise regret at time t is $X_t(k^*) - X_t(k)$ where k is the arm played by the policy and $k^* = \arg\max_{k \in \mathcal{K}} \mu_t(k)$. We run the experiment with multiple repetitions and averaging out the regret and compensation at each epoch.

Figure 5.3 shows the performance of Algorithm 11 with UCB1, ϵ-Greedy and Thompson Sampling, and figure 5.2 shows the reward performance of the same, with brief explanations. Table

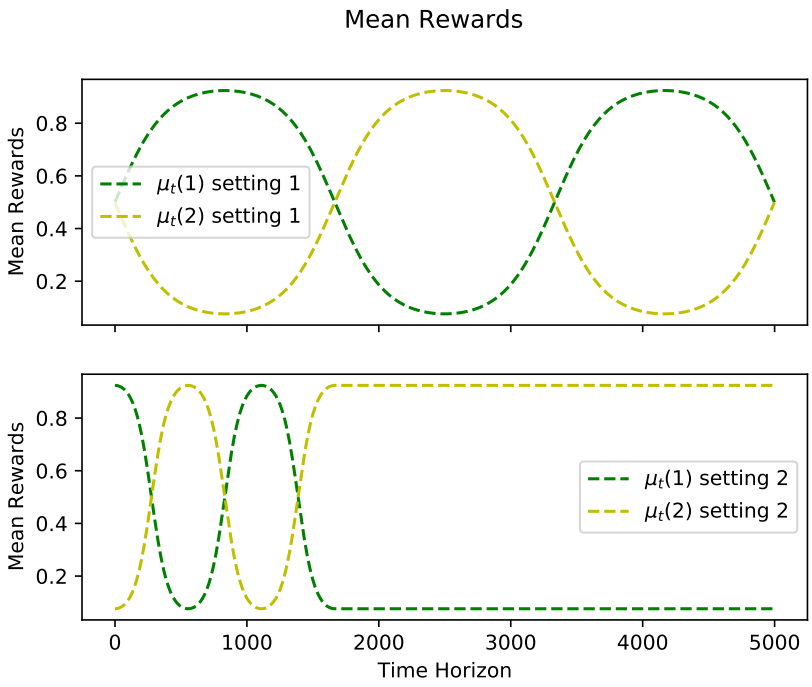

Figure 5.1: (Upper) Mean rewards for setting 1. (Below) Mean rewards for setting 2.

V_T	3	6	9	12	15	18	24
$\mathcal{R}^U$	156.1	175.9	185.7	191.4	198.3	207.5	210.3
$\mathcal{C}^U$	88.9	107.4	119.8	127.2	135.0	145.6	149.8
$\mathcal{R}^{\epsilon G}$	143.1	164.1	180.2	192.0	202.3	215.0	229.0
$\mathcal{C}^{\epsilon G}$	37.3	52.4	64.1	73.6	80.7	88.7	99.5
$\mathcal{R}^{TS}$	125.1	147.2	163.8	177.8	185.9	197.8	211.6
$\mathcal{C}^{TS}$	48.4	69.0	84.9	97.5	107.5	118.1	132.2

Table 5.1: This is the data of the performance of Algorithm 11 with all the policies: UCB1, ϵ-Greedy and Thompson Sampling for varying number of variation budget V_T. The superscripts $U, \epsilon G, TS$ stands for UCB1, ϵ-Greedy and Thompson Sampling respectively with $\mathcal{R}$ as the regret and $\mathcal{C}$ as the compensation values.

5.1 shows the performance of the same with varying degree of variation budget V_T. The regret values for all the policies increase by a maximum of $V_T^{1/3}$ as suggested by the theory. For compensation, we can see that the values increase a little more quickly for ϵ-Greedy and Thompson Sampling compared to UCB1, as UCB1 is less sensitive to V_T comparatively. However, ϵ-Greedy does not drastically change its compensation values in comparison to Thompson Sampling, which might suggest that the upper bound can be tightened.

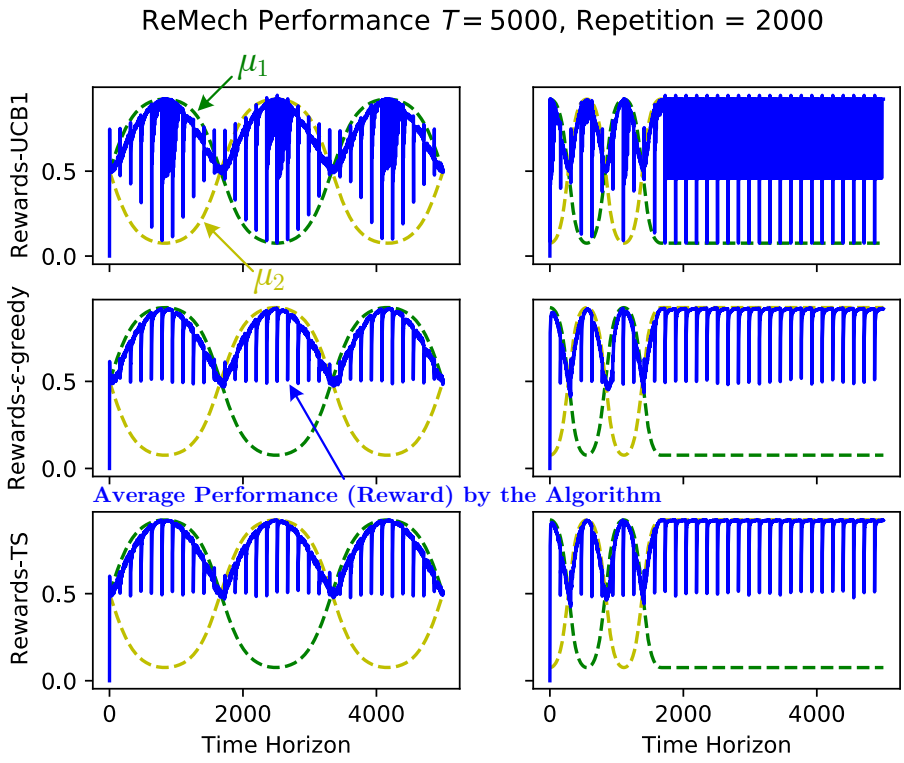

Figure 5.2: Algorithm 11 (written as ReMech in the diagram) performance with $T = 100$ with 2000 repetitions. The blue line is the average performance of Algorithm 11 at various epochs with UCB1, ε-Greedy and Thompson Sampling as the BanditAlg policy. The dotted yellow is the mean reward for arm 1 and the green is for arm 2. The lines beyond the green or yellow lines are due to the added drift in rewards. The lines which touch the bottom (only applicable to Algorithm 11 + UCB1 are due to the exploration phase of the UCB1 algorithm. The first column is the performance of all the policies for setting 1 and the second column is for setting 2.

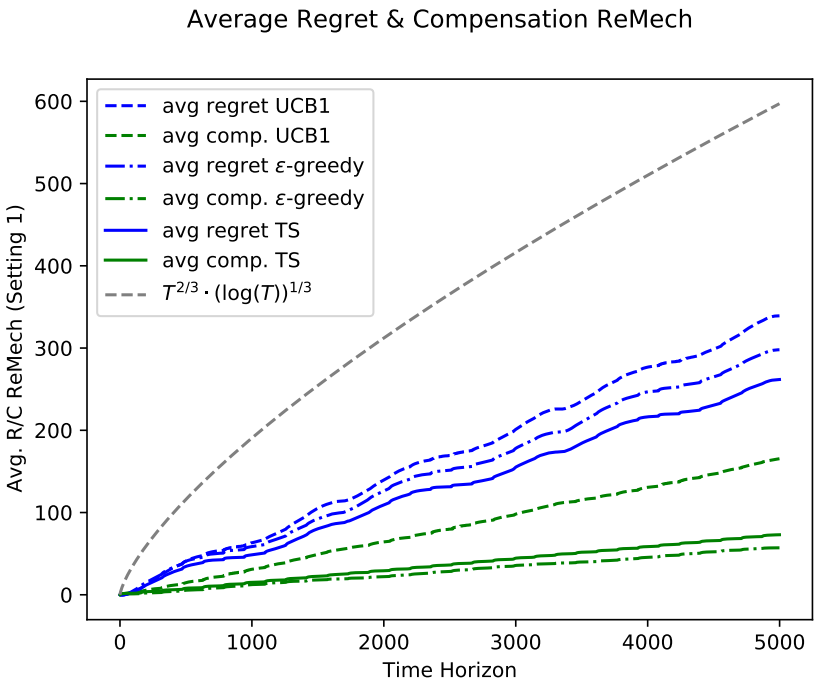

Figure 5.3: Algorithm 11 performance for a large horizon with $T = 5000$ with 2000 repetitions. This is the performance of all 3 policies for setting 1.

Chapter 6

Conclusion and Future Work

We studied the incentivized exploration problem under reward drift in a multi-armed bandit setting where the reward distributions of the arms may change over time, where the agent provides biased feedback under the influence of an incentive. We considered two different environments which capture the non-stationarity of rewards: piecewise-constant and continuously-changing. For a piecewise-constant environment, we propose a scheme with Algorithm 10 [29] with discounted UCB (algorithm 8) [22] and sliding window UCB(algorithm 9) [15] as it's subroutine. We achieve $(\tilde{O}(\sqrt{T}), \tilde{O}(\sqrt{T}))$ for D-UCB and $(\tilde{O}(\sqrt{T}), \tilde{O}(T^{1/4}))$ for SW-UCB with Algorithm 10 as the upper bounds for (regret, compensation) pair. Next we considered the continuously-changing environment as the model of non-stationarity, where we considered the restarting mechanism (algorithm 11) [5] to counter the changing rewards and the three algorithms of MAB literature: UCB1, ε-Greedy and Thompson Sampling along with it. We showed that the regret for all three algorithms and the compensation for UCB1, under drifted reward achieve $\tilde{O}(T^{2/3})$. The compensation for ε-Greedy and Thompson Sampling are $\tilde{O}(T^{1/3})$. Since all the schemes we propose are sub-linear in the size of horizon, we conclude that they are effective in incentivized exploration under reward drift with non-stationary rewards.

For future directions, one way is to look at the trust factor between the agent and the principal. Currently, we assume that the agent always listens to the principal and chooses the arm recommended to them, however, we can relax this assumption and see what effect it has on the regret and compensation of the overall model.

We can check the extent of information shared with the agents. In this work, the entire history was indirectly available to every agent in the form of the empirical average of the rewards. But in this case, there might be a temptation of using their own algorithms or heuristic policies to select arms at any instant. It is our assumption of trust which keeps things tractable. We can look at the effect of relaxing this assumption to the regret and the compensation.

For the related problem, but with a bayesian approach, introduced by [24] was discussed in [38]. The author showed that simply sending the recommendation to the agent should suffice.

The area of non-stationarity can also be explored to solve the incentivized exploration problem for the 'active' approach (section 2.2.2) to non-stationarity, or general non-stationary environments, which still provides sub-linear regret and compensation. Along with that, we can incorporate more information or context (usage of linear or contextual bandits) about the agents while providing recommendations.

Bibliography

[1] Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. CoRR, abs/1111.1797, 2011.

[2] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. Machine Learning, 47:235–256, 2004.

[3] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. SIAM J. Comput., 32:48–77, 2002.

[4] Donald A. Berry and Bert Fristedt. Bandit problems : sequential allocation of experiments / Donald A. Berry, Bert Fristedt. Chapman and Hall London ; New York, 1985.

[5] Omar Besbes, Yonatan Gur, and Assaf Zeevi. Stochastic multi-armed-bandit problem with non-stationary rewards. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems, volume 27. Curran Associates, Inc., 2014.

[6] Djallel Bouneffouf, A. Bouzeghoub, and Alda Lopes Gançarski. A contextual-bandit algorithm for mobile context-aware recommender system. In ICONIP, 2012.

[7] Eric Brochu, Matthew W. Hoffman, and Nando de Freitas. Portfolio Allocation for Bayesian Optimization. arXiv e-prints, page arXiv:1009.5419, September 2010.

[8] Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems, 2012.

[9] Yeon-Koo Che and Johannes Hörner. Recommender systems as mechanisms for social learning*. Quarterly Journal of Economics, 133:871–925, 05 2018.

[10] Lee Cohen and Yishay Mansour. Optimal algorithm for bayesian incentive-compatible exploration. In Proceedings of the 2019 ACM Conference on Economics and Computation, EC '19, page 135–151, New York, NY, USA, 2019. Association for Computing Machinery.

[11] Zahra Ehsani and M. Ehsani. Effect of quality and price on customer satisfaction and commitment in iran auto industry. 2015.

[12] Zhe Feng, David C. Parkes, and Haifeng Xu. The intrinsic robustness of stochastic bandits to strategic manipulation, 2020.

[13] Peter Frazier, David Kempe, Jon Kleinberg, and Robert Kleinberg. Incentivizing exploration. In Proceedings of the Fifteenth ACM Conference on Economics and Computation, EC '14, page 5–22, New York, NY, USA, 2014. Association for Computing Machinery.

[14] Yi Gai, Bhaskar Krishnamachari, and Ramesh Jain. Learning multiuser channel allocations in cognitive radio networks: A combinatorial multi-armed bandit formulation. pages 1 – 9, 05 2010.

[15] Aurélien Garivier and Eric Moulines. On upper-confidence bound policies for non-stationary bandit problems, 2008.

[16] J. C. Gittins. Bandit processes and dynamic allocation indices. Journal of the Royal Statistical Society. Series B (Methodological), 41(2):148–177, 1979.

[17] Lee Gwo-Guang and Lin Hsiu-Fen. Consumer perceptions of e-service quality in online shopping. International Journal of Retail Distribution Management, 33:161–176, 02 2005.

[18] Li Han, David Kempe, and Ruixin Qiang. Incentivizing exploration with heterogeneous value of money. Lecture Notes in Computer Science, page 370–383, 2015.

[19] Christoph Hirnschall, Adish Singla, Sebastian Tschiatschek, and Andreas Krause. Learning user preferences to incentivize exploration in the sharing economy. Proceedings of the AAAI Conference on Artificial Intelligence, 32(1), Apr. 2018.

[20] Nicole Immorlica, Jieming Mao, Alex Slivkins, and Steven Wu. Bayesian exploration with heterogeneous agents. In The Web Conference 2019, May 2019.

[21] Emir Kamenica and Matthew Gentzkow. Bayesian persuasion. American Economic Review, 101(6):2590–2615, October 2011.

[22] L Kocsis and C Szepesvari. Discounted ucb pascal challenges workshop. Venice, Italy (April 2006), 2006.

[23] Ilan Kremer, Yishay Mansour, and Motty Perry. Implementing the wisdom of the crowd. volume 122, pages 605–606, 06 2013.

[24] Ilan Kremer, Yishay Mansour, and Motty Perry. Implementing the wisdom of the crowd. volume 122, pages 605–606, 06 2013.

[25] Tor Lattimore and Csaba Szepesvári. Bandit Algorithms. Cambridge University Press, 2020.

[26] Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. Proceedings of the 19th international conference on World wide web - WWW '10, 2010.

[27] Fang Liu, Joohyun Lee, and Ness Shroff. A change-detection based framework for piecewise-stationary multi-armed bandit problem, 2017.

[28] Yang Liu and Chien-Ju Ho. Incentivizing high quality user contributions: New arm generation in bandit learning, 2018.

[29] Zhiyuan Liu, Huazheng Wang, Fan Shen, Kai Liu, and Lijun Chen. Incentivized exploration for multi-armed bandits under reward drift. Proceedings of the AAAI Conference on Artificial Intelligence, 34(04):4981–4988, Apr. 2020.

[30] Thodoris Lykouris, Vahab Mirrokni, and Renato Paes Leme. Stochastic bandits robust to adversarial corruptions, 2018.

[31] Yishay Mansour, Aleksandrs Slivkins, and Vasilis Syrgkanis. Bayesian incentive-compatible bandit exploration. In Proceedings of the Sixteenth ACM Conference on Economics and Computation, EC '15, page 565–582, New York, NY, USA, 2015. Association for Computing Machinery.

[32] Yishay Mansour, Aleksandrs Slivkins, and Vasilis Syrgkanis. Bayesian incentive-compatible bandit exploration, 2019.

[33] Anne Martensen, Lars Gronholdt, and Kai Kristensen. The drivers of customer satisfaction and loyalty: Cross-industry findings from denmark. Total Quality Management, 11(4-6):544–553, 2000.

[34] William H. Press. From the Cover: Bandit solutions provide unified ethical models for randomized clinical trials and comparative effectiveness research. Proceedings of the National Academy of Science, 106(52):22387–22392, December 2009.

[35] Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. Learning diverse rankings with multi-armed bandits. In Proceedings of the 25th International Conference on Machine Learning, ICML '08, page 784–791, New York, NY, USA, 2008. Association for Computing Machinery.

[36] Daniel Russo, Benjamin Van Roy, Abbas Kazerouni, and Ian Osband. A tutorial on thompson sampling. CoRR, abs/1707.02038, 2017.

[37] Mark Sellke and Aleksandrs Slivkins. The price of incentivizing exploration: A characterization via thompson sampling and sample complexity, 2021.

[38] Aleksandrs Slivkins. Introduction to multi-armed bandits, 2021.

[39] Richard S. Sutton and Andrew G. Barto. Reinforcement Learning: An Introduction. A Bradford Book, Cambridge, MA, USA, 2018.

[40] WILLIAM R THOMPSON. ON THE LIKELIHOOD THAT ONE UNKNOWN PROBABILITY EXCEEDS ANOTHER IN VIEW OF THE EVIDENCE OF TWO SAMPLES. Biometrika, 25(3-4):285–294, 12 1933.

[41] Siwei Wang and Longbo Huang. Multi-armed bandits with compensation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 31. Curran Associates, Inc., 2018.