Multi-armed Bandits with Compensation

Siwei Wang
IIIS, Tsinghua University
wangsw15@mails.tsinghua.edu.cn

Longbo Huang[*]
IIIS, Tsinghua University
longbohuang@tsinghua.edu.cn

Abstract

We propose and study the known-compensation multi-arm bandit (KCMAB) problem, where a system controller offers a set of arms to many short-term players for T steps. In each step, one short-term player arrives to the system. Upon arrival, the player aims to select an arm with the current best average reward and receives a stochastic reward associated with the arm. In order to incentivize players to explore other arms, the controller provides a proper payment compensation to players. The objective of the controller is to maximize the total reward collected by players while minimizing the compensation. We first provide a compensation lower bound $\Theta(\sum_i \frac{\Delta_i \log T}{KL_i})$, where Δ_i and KL_i are the expected reward gap and Kullback-Leibler (KL) divergence between distributions of arm i and the best arm, respectively. We then analyze three algorithms to solve the KCMAB problem, and obtain their regrets and compensations. We show that the algorithms all achieve $O(\log T)$ regret and $O(\log T)$ compensation that match the theoretical lower bound. Finally, we present experimental results to demonstrate the performance of the algorithms.

1 Introduction

Multi-arm bandit (MAB) is a game that lasts for an unknown time horizon T [4, 17]. In each time slot, the controller has N arms to pull, and pulling different arms will result in different feedbacks. In the stochastic MAB model [12], feedbacks from a single arm follow an associated distribution, which is unknown to the player. These feedbacks are random variables independent of any other events. After pulling the arm, the controller collects a reward that depends on the feedback. The controller aims to maximize the sum of rewards during the game by choosing a proper arm to pull in each time slot, and the decision can depend on all available information, i.e., past chosen arms and feedbacks. The common metric for evaluating the performance of a policy is the value of regret, defined as the expected difference between the player's reward and pulling an arm that generates the largest average reward.

The MAB formulation models the trade-off between exploration and exploitation, where exploration concerns exploring the potential best arms, but can result in pulling sub-optimal arms, while exploitation aims at choosing arms with the current best performance and can lose reward if that arm is in fact sub-optimal. Thus, optimizing this trade-off is very important for any controller seeking to minimize regret. However, in many real-world applications, arms are not pulled by the controller concerning long-term performance. Instead, actions are taken by short-term players interested in optimizing their instantaneous reward. In this case, an important means is to provide monetary compensation to players, so that they act as if they are pulling the arms on behalf of the controller, to jointly minimize regret, e.g., [8]. Our objective in this paper is to seek an incentivizing policy, so as to minimize regret while not giving away too much compensation.

As a concrete example, consider the scenario where a e-commerce website recommends goods to consumers. When a consumer chooses to purchase a certain good, he receives the reward of that good. The website similarly collects the same reward as a recognition of the recommendation. In this model, the website acts as a controller that decides how to provide recommendations. Yet, the actual decision is made by consumers, who are not interested in exploration and will choose to optimize their reward greedily. However, being a long-term player, the website cares more about maximizing the total reward during the game. As a result, he needs to devise a scheme to influence the choices of short-term consumers, so that both the consumers and website can maximize their benefits. One common way to achieve this goal in practice is that the website offers customized discounts for certain goods to consumers, i.e., by offering compensation to pay for part of the goods. In this case, each customer, upon arrival, will choose the good with largest expected reward plus the compensation. The goal of the e-commerce site is to find an optimal compensation policy to minimize his regret, while not spending too much additional payment.

It is important to notice the difference between regret and compensation. In particular, regret comes from pulling a sub-optimal arm, while compensation comes from pulling an arm with poor past behavior. For example, consider two arms with expected reward 0.9 and 0.1. Suppose in the first twenty observations, arm 1 has an empirical mean 0.1 but arm 2 has an empirical mean 0.9. Then, in the next time slot, pulling arm 2 will cause regret 0.8, since its expected gain is 0.8 less than arm 1. But in a short-term player's view, arm 2 behaves better than arm 1. Thus, pulling arm 2 does not require any compensation, while pulling arm 1 needs 0.8 for compensation. As a result, the two measures can behave differently and require different analysis, i.e., regret depends heavily on learning the arms well, while compensation is largely affected by how the reward dynamics behaves.

There is a natural trade-off between regret and compensation. If one does not offer any compensation, the resulting user selection policy is greedy, which will lead to a $\Theta(T)$ regret. On the other hand, if one is allowed to have arbitrary compensation, one can achieve an $O(\log T)$ regret with many existing algorithms. The key challenge in obtaining the best trade-off between regret and compensation lies in that the compensation value depends on the random history. As a consequence, different random history not only leads to different compensation value, but also results in different arm selection. Moreover, in practice, the compensation budget may be limited, e.g., a company hopes to maximize its total income which equals to reward subtracts compensation. These make it hard to analyze its behavior.

1.1 Related works

The incentivized learning model has been investigated in prior works, e.g., [8, 14, 15]. In [8], the model contains a prior distribution for each arm's mean reward at the beginning. As time goes on, observations from each arm update the posterior distributions, and subsequent decisions are made based on posterior distributions. The objective is to optimize the total discounted rewards. Following their work, [14] considered the case when the rewards are not discounted, and they show an algorithm to achieve regret upper bound of $O(\sqrt{T})$. In [15], instead of a simple game, there is a complex game in each time slot that contains more players and actions. These incentivization formulations can model many practical applications, including crowdsourcing and recommendation systems [6, 16].

In this paper, we focus on the non-Bayesian setting and consider the non-discounted reward. As has been pointed out in [14], the definition of user expectation is different in this case. Specifically, in our setting, each player selects arms based on their empirical means, whereas in the Bayesian setting, it is possible for a player to also consider posterior distributions of arms for decision making. We propose three algorithms for solving our problem, which adapt ideas from existing policies for stochastic MAB, i.e., Upper Confidence Bound (UCB) [2, 9], Thompson Sampling (TS) [18] and ε-greedy [20]. These algorithms can guarantee $O(\log T)$ regret upper bound, which matches the regret lower bound $\Theta(\log T)$ [12].

Another related bandit model is contextual bandit, where a context is contained in each time slot [3, 5, 13]. The context is given before a decision is made, and the reward depends on the context. As a result, arm selection also depends on the given context. In incentivized learning, the short-term players can view the compensation as a context, and their decisions are influenced by the context. However, different from contextual bandits, where the context is often an exogenous random variable and the controller focuses on identifying the best arm under given contexts, in our case, the context is given by the controller and itself influenced by player actions. Moreover, the controller needs to pay

for obtaining a desired context. What he needs is the best way to construct a context in every time slot, so that the total cost is minimized.

In the budgeted MAB model, e.g., [7, 19, 21], players also need to pay for pulling arms. In this model, pulling each arm costs a certain budget. The goal for budgeted MAB is to maximize the total reward subject to the budget constraint. The main difference from our work is that in budgeted MAB, the cost budget for pulling each arm is pre-determined and it does not change with the reward history. In incentivized learning, however, different reward sample paths will lead to different costs for pulling the same arm.

1.2 Our contributions

The main contributions of our paper are summarized as follows:

1. We propose and study the Known-Compensation MAB problem (KCMAB). In KCMAB, a long-term controller aims to optimize the accumulated reward but has to offer compensation to a set of short-term players for pulling arms. Short-term players, on the other hand, arrive at the system and make greedy decisions to maximize their expected reward plus compensation, based on knowledge about previous rewards and offered compensations. The objective of the long-term controller is to design a proper compensation policy, so as to minimize his regret with minimum compensation. KCMAB is a non-Bayesian and non-discounted extension of the model in [8].

2. In KCMAB, subject to the algorithm having an $o(T^\alpha)$ regret for any $\alpha > 0$, we provide a $\Theta(\log T)$ lower bound for the compensation. This compensation lower bound has the same order as the regret lower bound, which means that one cannot expect a compensation to be much less than its regret, if the regret is already small.

3. We propose algorithms to solve the KCMAB problem and present their compensation analysis. Specifically, we provide the analyses of compensation for the UCB policy, a modified ε-greedy policy and a modified-TS policy. All these algorithms have $O(\log T)$ regret upper bounds while using compensations upper bounded $O(\log T)$, which matches the lower bound (in order).

4. We provide experimental results to demonstrate the performance of our algorithms. In experiments, we find that modified TS policy behaves better than UCB policy, while the modified ε-greedy policy has regret and compensation slightly larger than those under the modified-TS policy. We also compare the classic TS algorithm and our modified-TS policy. The results show that our modification is not only effective in analysis, but also impactful on actual performance. Our results also demonstrate the trade-off between regret and compensation.

2 Model and notations

In the Known-Compensation Multi-Arm Bandit (KCMAB) problem, a central controller has N arms $\{1, \cdots, N\}$. Each arm i has a reward distribution denoted by D_i with support $[0, 1]$ and mean μ_i. Without loss of generality, we assume $1 \geq \mu_1 > \mu_2 \geq \cdots \mu_N \geq 0$ and set $\Delta_i = \mu_1 - \mu_i$ for all $i \geq 2$. The game is played for T time steps. In each time slot t, a short-term player arrives at the system and needs to choose an arm $a(t)$ to pull. After the player pulls arm $a(t)$, the player and the controller each receive a reward $X(t)$ from the distribution $D_{a(t)}$, denoted by $X_{a(t)}(t) \sim D_{a(t)}$, which is an independent random variable every time arm $a(t)$ is pulled.

Different from the classic MAB model, e.g., [12], where the only control decision is arm selection, the controller can also choose to offer a compensation to a player for pulling a particular arm, so as to incentivize the player to explore an arm favored by the controller. We denote the paid compensation $c(t) = c_{a(t)}(t)$, and assume that it can depend on all the previous information, i.e., it depends on $\mathcal{F}_{t-1} = \{(a(\tau), X(\tau), c(\tau)) | 1 \leq \tau \leq t - 1\}$. Each player is assumed to choose an arm greedily to maximize his total expected income based on past observations. Here income for pulling arm i equals to $\hat{\mu}_i(t) + c_i(t)$, where $\hat{\mu}_i(t) \triangleq M_i(t)/N_i(t)$ is the empirical mean of arm i, with $N_i(t) = \sum_{\tau < t} \mathbb{I}[a(\tau) = i]$ being the total number of times for pulling arm i and

$M_i(t) = \sum_{\tau < t} \mathbb{I}[a(\tau) = i]X(t)$ being the total reward collected. Therefore, each player will pull arm $i = \text{argmax}_j\{\hat{\mu}_j(t) + c_j(t)\}$.

The long-term controller, on the other hand, concerns about the expected total reward. Following the MAB tradition, we define the following total regret:

$$Reg(T) = T \max_i \mu_i - Rew(T) = T\mu_1 - Rew(T),$$

where $Rew(T)$ denotes the expected total reward that the long-term controller can obtain until time horizon T. We then use $Com_i(T) = \mathbb{E}\left[\sum_{\tau=1}^{T} \mathbb{I}[a(\tau) = i]c(\tau)\right]$ to denote the expected compensation paid for arm i, and denote $Com(T) = \sum_i Com_i(T)$ the expected total compensation.

It is known from [12] that $Reg(T)$ has a lower bound of $\Omega(\sum_{i=2}^{N} \frac{\Delta_i \log T}{KL(D_i, D_1)})$, where $KL(P, Q)$ denotes the Kullback-Leibler (KL)-divergence between distributions P and Q, even when a single controller is pulling arms for all time. Thus, our objective is to minimize the compensation while keeping the regret upper bounded by $O(\sum_{i=2}^{N} \frac{\Delta_i \log T}{KL(D_i, D_1)})$.

Note that in the non-Bayesian model, if there are no observations for some arm i, players will have no knowledge about its mean reward and they cannot make decisions. Thus, we assume without loss of generality that in the first N time slots of the game, with some constant compensation, the long-term controller can control the short-term players to choose all the arms once. This assumption does not influence the results in this paper.

In the following, we will present our algorithms and analysis. Due to space limitation, all complete proofs in this paper are deferred to the supplementary file. We only provide proof sketches in the main text.

3 Compensation lower bound

In this section, we first derive a compensation lower bound, subject to the constraint that the algorithm guarantees an $o(T^\alpha)$ regret for any $\alpha > 0$. We will make use of the following simple fact to simplify the computation of compensation at every time slot.

Fact 1. *If the long-term controller wants the short-term player to choose arm i in time slot t, then the minimum compensation he needs to pay on pulling arm i is $c_i(t) = \max_j \hat{\mu}_j(t) - \hat{\mu}_i(t)$.*

With Fact 1, we only need to consider the case $c(t) = \max_j \hat{\mu}_j(t) - \hat{\mu}_i(t)$ for arm i.

Theorem 1. *In KCMAB, if an algorithm guarantees an $o(T^\alpha)$ regret upper bound for any fixed T and any $\alpha > 0$, then there exist examples of Bernoulli Bandits, i.e., arms having reward 0 or 1 every time, such that the algorithm must pay $\Omega\left(\sum_{i=2}^{N} \frac{\Delta_i \log T}{KL(D_i, D_1)}\right)$ for compensation in these examples.*

Proof Sketch: Suppose an algorithm achieves an $o(T^\alpha)$ regret upper bound for any $\alpha > 0$. We know that it must pull a sub-optimal arm i for $\Omega\left(\frac{\log T}{KL(D_i, D_1)}\right)$ times almost surely [12]. Now denote $t_i(k)$ be the time slot (a random variable) that we choose arm i for the k-th time. We see that one needs to pay $\mathbb{E}[\max_j \hat{\mu}_j(t_i(k)) - \hat{\mu}_i(t_i(k))] \geq \mathbb{E}[\hat{\mu}_1(t_i(k))] - \mathbb{E}[\hat{\mu}_i(t_i(k))]$ for compensation in that time slot. By definition of $t_i(k)$ and the fact that all rewards are independent with each other, we always have $\mathbb{E}[\hat{\mu}_i(t_i(k))] = \mu_i$.

It remains to bound the value $\mathbb{E}[\hat{\mu}_1(t_i(k))]$. Intuitively, when μ_1 is large, $\mathbb{E}[\hat{\mu}_1(t_i(k))]$ cannot be small, since those random variables are with mean μ_1. Indeed, when $\mu_1 > 0.9$ and D_1 is a Bernoulli distribution, one can prove that $\mathbb{E}[\hat{\mu}_1(t_i(k))] \geq \frac{\mu_1}{2} - 2\delta(T)$ with a probabilistic argument, where $\delta(T)$ converges to 0 as T goes to infinity. Thus, for large μ_1 and small μ_2 (so are μ_i for $i \geq 2$), we have that $\mathbb{E}[\hat{\mu}_1(t_i(k))] - \mu_i = \Omega(\mu_1 - \mu_i)$ holds for any i and $k \geq 2$. This means that the compensation we need to pay for pulling arm i once is about $\Theta(\mu_1 - \mu_i) = \Omega(\Delta_i)$. Thus, the total compensation $\Omega\left(\sum_{i=2}^{N} \frac{\Delta_i \log T}{KL(D_i, D_1)}\right)$. $\square$

```
1: for t = 1, 2, · · · , N do
2:     Choose arm a(t) = t.
3: end for
4: for t = N + 1, · · · do
5:     For all arm i, compute r_i(t) = √(2 log t / N_i(t)) and u_i(t) = μ̂_i(t) + r_i(t)
6:     Choose arm a(t) = argmax_i u_i(t) (with compensation max_j μ̂_j(t) − μ̂_{a(t)}(t))
7: end for
```

Algorithm 1: The UCB algorithm for KCMAB.

4 Compensation upper bound

In this section, we consider three algorithms that can be applied to solve the KCMAB problem and present their analyses. Specifically, we consider the Upper Confidence Bound (UCB) Policy [2], and propose a modified ε-Greedy Policy and a modified-Thompson Sampling Policy. Note that while the algorithms have been extensively analyzed for their regret performance, the compensation metric is significantly different from regret. Thus, the analyses are different and require new arguments.

4.1 The Upper Confidence Bound policy

We start with the UCB policy shown in Algorithm 1. In the view of the long-term player, Algorithm 1 is almost the same as the UCB policy in [2], and its regret has been proved to be $O\left(\sum_{i=2}^{N} \frac{\log T}{\Delta_i}\right)$. Thus, we focus on the compensation upper bound, which is shown in Theorem 2.

Theorem 2. *In Algorithm 1, we have that*

$$Com(T) \leq \sum_{i=2}^{N} \frac{16 \log T}{\Delta_i} + \frac{2N\pi^2}{3}$$

Proof Sketch: First of all, it can be shown that the each sub-optimal arm is pulled for at most $\frac{8}{\Delta_i^2} \log T$ times in Algorithm 1 with high probability. Since in every time slot t the long-term controller choose the arm $a(t) = \operatorname{argmax}_j \hat{\mu}_j(t) + r_j(t)$, we must have $\hat{\mu}_{a(t)}(t) + r_{a(t)}(t) = \max_j(\hat{\mu}_j(t) + r_j(t)) \geq \max_j \hat{\mu}_j(t)$. This implies that the compensation is at most $r_{a(t)}(t)$. Moreover, if arm $a(t)$ has been pulled the maximum number of times, i.e., $N_{a(t)}(t) = \max_j N_j(t)$, then $r_{a(t)}(t) = \min_j r_j(t)$ (by definition). Thus, $\hat{\mu}_{a(t)}(t) = \max_j \hat{\mu}_j(t)$, which means that the controller does not need to pay any compensation.

Next, for any sub-optimal arm i, with high probability the compensation that the long-term controller pays for it can be upper bounded by:

$$Com_i(T) \leq \mathbb{E}\left[\sum_{\tau=1}^{N_i(T)} \sqrt{\frac{2 \log T}{\tau}}\right] \leq \mathbb{E}\left[\sqrt{8 N_i(T) \log T}\right] \overset{(a)}{\leq} \sqrt{8 \mathbb{E}[N_i(T)] \log T} \leq \frac{8 \log T}{\Delta_i}$$

Here the inequality (a) holds because $\sqrt{x}$ is concave. As for the optimal arm, when $N_1(t) \geq \sum_{i=2}^{N} \frac{8 \log T}{\Delta_i^2}$, with high probability $N_1(t) = \max_j N_j(t)$. Thus, the controller does not need to pay compensation in time slots with $a(t) = 1$ and $N_1(t) \geq \sum_{i=2}^{N} \frac{8 \log T}{\Delta_i^2}$. Using the same argument, the compensation for arm 1 is upper bounded by $Com_1(T) \leq \sum_{i=2}^{N} \frac{8 \log T}{\Delta_i}$ with high probability. Therefore, the overall compensation upper bound is given by $Com(T) \leq \sum_{i=2}^{N} \frac{16 \log T}{\Delta_i}$ with high probability. □

4.2 The modified ε-greedy policy

The second algorithm we propose is a modified ε-greedy policy, whose details are presented in Algorithm 2. The modified ε-greedy algorithm, though appears to be similar to the classic ε-greedy algorithm, has a critical difference. In particular, instead of randomly choosing an arm to explore,

```
1: Input: ε,
2: for t = 1, 2, · · · , N do
3:    Choose arm a(t) = t.
4: end for
5: a_e ← 1
6: for t = N + 1, · · · do
7:    With probability min{1, ε/t}, choose arm a(t) = a_e and set a_e ← (a_e mod N) + 1 (with
       compensation max_j μ̂_j(t) − μ̂_{a(t)}(t)).
8:    Else, choose the arm a(t) = argmax_i μ̂_i(t).
9: end for
```

Algorithm 2: The modified ε-greedy algorithm for KCMAB.

we use the round robin method to explore the arms. This guarantees that, given the number of total explorations, each arm will be explored a deterministic number of times. This facilitates the analysis for compensation upper bound.

In the regret analysis of the ε-greedy algorithm, the random exploration ensures that at time slot t, the expectation of explorations on each arm is about $\frac{\epsilon}{N} \log t$. Thus, the probability that its empirical mean has a large error is small. In our algorithm, the number of explorations of each single arm is almost the same as classic ε-greedy algorithm in expectation (with only a small constant difference). Hence, adapting the analysis from ε-greedy algorithm gives the same regret upper bound, i.e. $O(\sum_{i=2}^{N} \frac{\Delta_i \log T}{\Delta_2^2})$ when $\epsilon = \frac{cN}{\Delta_2^2}$.

Next, we provide a compensation upper bound for our modified ε-greedy algorithm.

Theorem 3. *In Algorithm 2, if we have $\epsilon = \frac{cN}{\Delta_2^2}$, then*

$$Com(T) \leq \sum_{i=2}^{N} \frac{c\Delta_i \log T}{\Delta_2^2} + \frac{N^2}{2\Delta_2} \sqrt{c \log T}. \tag{1}$$

Proof Sketch: Firstly, our modified ε-greedy algorithm chooses the arm with the largest empirical mean in non-exploration steps. Thus, we only need to consider the exploration steps, i.e., steps during which we choose to explore arms according to round-robin. Let $t_i^\varepsilon(k)$ be the time slot that we explore arm i for the k-th time. Then the compensation the controller has to pay in this time slot is $\mathbb{E}[\max_j \mû_j(t_i^\varepsilon(k)) − μ̂_i(t_i^\varepsilon(k))]$.

Since the rewards are independent of whether we choose to explore, one sees that $\mathbb{E}[μ̂_i(t_i^\varepsilon(k))] = μ_i$. Thus, we can decompose $\mathbb{E}[\max_j μ̂_j(t_i^\varepsilon(k)) − μ̂_i(t_i^\varepsilon(k))]$ as follows:

$$
\begin{aligned}
\mathbb{E}[\max_j μ̂_j(t_i^\varepsilon(k)) − μ̂_i(t_i^\varepsilon(k))] &= \mathbb{E}[\max_j(μ̂_j(t_i^\varepsilon(k)) − μ_i)] \\
&\leq \mathbb{E}[\max_j(μ̂_j(t_i^\varepsilon(k)) − μ_j)] + \mathbb{E}[\max_j(μ_j − μ_i)]. \tag{2}
\end{aligned}
$$

The second term in (2) is bounded by $\Delta_i = μ_1 − μ_i$. Summing over all these steps and all arms, we obtain the first term $\sum_{i=2}^{N} \frac{c\Delta_i \log T}{\Delta_2^2}$ in our bound (1).

We turn to the first term in (2), i.e., $\mathbb{E}[\max_j(μ̂_j(t_i^\varepsilon(k)) − μ_j)]$. We see that it is upper bounded by

$$\mathbb{E}[\max_j(μ̂_j(t_i^\varepsilon(k)) − μ_j)] \leq \mathbb{E}[\max_j(μ̂_j(t_i^\varepsilon(k)) − μ_j)^+] \leq \sum_j \mathbb{E}[(μ̂_j(t_i^\varepsilon(k)) − μ_j)^+]$$

where $(*)^+ = \max\{*, 0\}$. When arm i has been explored k times (line 7 in Algorithm 2), we know that all other arms have at least k observations (in the first N time slots, there is one observation for each arm). Hence, $\mathbb{E}[(μ̂_j(t_i^\varepsilon(k)) − μ_j)^+] = \frac{1}{2}\mathbb{E}[|μ̂_j(t_i^\varepsilon(k)) − μ_j|] \leq \frac{1}{4\sqrt{k}}$ (the equality is due to the fact that $\mathbb{E}[|x|] = 2\mathbb{E}[x^+]$ if $\mathbb{E}[x] = 0$).

Suppose arm i is been explored in time set $T_i = \{t_i^\varepsilon(1), \cdots\}$. Then,

$$\sum_{k \leq |T_i|} \mathbb{E}[\max_j(μ̂_j(t_i^\varepsilon(k)) − μ_j)^+] \leq \sum_{k \leq |T_i|} \frac{N}{4\sqrt{k}} \leq \frac{N\sqrt{|T_i|}}{2}$$

6

```
1: Init: $\alpha_i = 1, \beta_i = 1$ for each arm $i$.
2: for $t = 1, 2, \cdots, N$ do
3:     Choose arm $a(t) = t$ and receive the observation $X(t)$.
4:     Update($\alpha_{a(t)}, \beta_{a(t)}, X(t)$)
5: end for
6: for $t = N + 1, N + 3, \cdots$ do
7:     For all $i$ sample values $\theta_i(t)$ from Beta distribution $\mathbf{B}(\alpha_i, \beta_i)$;
8:     Play action $a_1(t) = \mathrm{argmax}_i \hat{\mu}_i(t)$, get the observation $X(t)$. Update($\alpha_{a_1(t)}, \beta_{a_1(t)}, X(t)$)
9:     Play action $a_2(t+1) = \mathrm{argmax}_i \theta_i(t)$ (with compensation $\max_j \hat{\mu}_j(t+1) - \hat{\mu}_{a_2(t+1)}(t+1)$),
        receive the observation $X(t+1)$. Update($\alpha_{a_2(t+1)}, \beta_{a_2(t+1)}, X(t+1)$)
10: end for
```

Algorithm 3: The Modified Thompson Sampling Algorithm for KCMAB.

```
1: Input: $\alpha_i, \beta_i, X(t)$
2: Output: updated $\alpha_i, \beta_i$
3: $Y(t) \leftarrow 1$ with probability $X(t)$, 0 with probability $1 - X(t)$
4: $\alpha_i \leftarrow \alpha_i + Y(t); \beta_i \leftarrow \beta_i + 1 - Y(t)$
```

Algorithm 4: Procedure Update

Since $\mathbb{E}[|T_i|] = \frac{c}{\Delta_2^2} \log T$, we can bound the first term in (2) as $\frac{N^2 \sqrt{c \log T}}{2\Delta_2}$. Summing this with $\sum_{i=2}^{N} \frac{c\Delta_i \log T}{\Delta_2^2}$ above for the second term, we obtain the compensation upper bound in (1). □

4.3 The Modified Thompson Sampling policy

The third algorithm we propose is a Thompson Sampling (TS) based policy. Due to the complexity of the analysis for the traditional TS algorithm, we propose a modified TS policy and derive its compensation bound. Our modification is motivated by the idea of the LUCB algorithm [10]. Specifically, we divide time into rounds containing two time steps each, and pull not only the arm with largest sample value, but also the arm with largest empirical mean in each round. The modified TS policy is presented in Algorithm 3, and we have the following theorem about its regret and compensation.

Theorem 4. *In Algorithm 3, we have*

$$Reg(T) \leq \sum_i \frac{2\Delta_i}{(\Delta_i - \varepsilon)^2} \log T + O\left(\frac{N}{\varepsilon^4}\right) + F_1(\boldsymbol{\mu})$$

for some small $\varepsilon < \Delta_2$ and $F_1(\boldsymbol{\mu})$ does not depend on (T, ε). As for compensation, we have:

$$Com(T) \leq \sum_i \frac{8}{\Delta_i - \varepsilon} \log T + N \log T + O\left(\frac{N}{\varepsilon^4}\right) + F_2(\boldsymbol{\mu})$$

where $F_2(\boldsymbol{\mu})$ does not depend on (T, ε) as well.

Proof Sketch: In round $(t, t + 1)$, we assume that we first run the arm with largest empirical mean on time slot t and call t an empirical step. Then we run the arm with largest sample on time slot $t + 1$ and call $t + 1$ a sample step.

We can bound the number of sample steps during which we pull a sub-optimal arm, using existing results in [1], since all sample steps form an approximation of the classic TS algorithm. Moreover, [11] shows that in sample steps, the optimal arm is pulled for many times (at least t^b at time t with a constant $b \in (0, 1)$). Thus, after several steps, the empirical mean of the optimal arm will be accurate enough. Then, if we choose to pull sub-optimal arm i during empirical steps, arm i must have an inaccurate empirical mean. Since the pulling will update its empirical mean, it is harder and harder for the arm's empirical mean to remain inaccurate. As a result, it cannot be pulled for a lot of times during the empirical steps as well.

Next, we discuss how to bound its compensation. It can be shown that with high probability, we always have $|\theta_i(t) - \hat{\mu}_i(t)| \leq r_i(t)$, where $r_i(t) = \sqrt{\frac{2 \log t}{N_i(t)}}$ is defined in Algorithm 1. Thus, we can
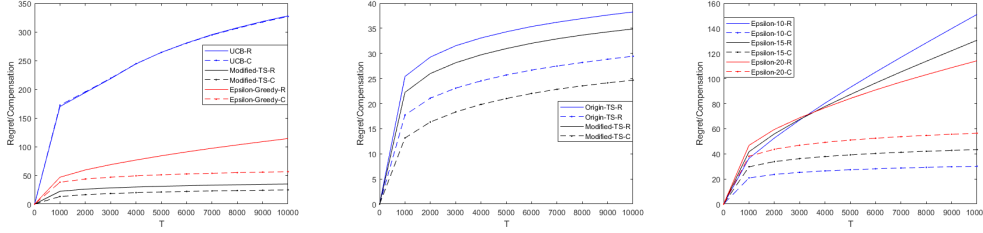
7

Figure 1: Regret and Compensation of Three policies.   Figure 2: Regret and Compensation of TS and modified-TS.   Figure 3: Regret and Compensation of modified ε-greedy.

focus on the case that $|\theta_i(t) - \hat{\mu}_i(t)| \leq r_i(t)$ happens for any i and t. Note that we do not need to pay compensation in empirical steps. In sample steps, suppose we pull arm i and the largest empirical mean is in arm $j \neq i$ at the beginning of this round. Then, we need to pay $\max_k \hat{\mu}_k(t+1) - \hat{\mu}_i(t+1)$, which is upper bounded by $\hat{\mu}_j(t) - \hat{\mu}_i(t) + (\hat{\mu}_j(t+1) - \hat{\mu}_j(t))^+ \leq \hat{\mu}_j(t) - \hat{\mu}_i(t) + \frac{1}{N_j(t)}$ (here $\hat{\mu}_i(t+1) = \hat{\mu}_i(t)$). As $\theta_i(t) \geq \theta_j(t)$, we must have $\hat{\mu}_i(t) + r_i(t) \geq \theta_i(t) \geq \theta_j(t) \geq \hat{\mu}_j(t) - r_j(t)$, which implies $\hat{\mu}_j(t) - \hat{\mu}_i(t) \leq r_i(t) + r_j(t)$. Thus, what we need to pay is at most $r_i(t) + r_j(t) + \frac{1}{N_j(t)}$ if $i \neq j$, in which case we can safely assume that we pay $r_j(t) + \frac{1}{N_j(t)}$ during empirical steps, and $r_i(t)$ during sample steps.

For an sub-optimal arm i, we have $Com_i(T) \leq \sum_i \frac{4}{\Delta_i - \varepsilon} \log T + O(\frac{1}{\varepsilon^4}) + F_1(\boldsymbol{\mu}) + \log T$ (summing over $r_i(t)$ gives the same result as in the UCB case, and summing over $\frac{1}{N_i(t)}$ is upper bounded by $\log T$). As for arm 1, when $a_1(t) = a_2(t+1) = 1$, we do not need to pay $r_1(t)$ twice. In fact, we only need to pay at most $\frac{1}{N_1(t)}$. Then, the number of time steps that $a_1(t) = a_2(t+1) = 1$ does not happen is upper bounded by $\sum_{i=2}^N \left(\frac{2}{(\Delta_i - \varepsilon)^2} \log T \right) + O\left(\frac{N}{\varepsilon^4} \right) + F_1(\boldsymbol{\mu})$, which is given by regret analysis. Thus, the compensation we need to pay on arm 1 is upper bounded by $\sum_i \frac{4}{\Delta_i - \varepsilon} \log T + O(\frac{1}{\varepsilon^4}) + F_1(\boldsymbol{\mu}) + \log T$. Combining the above, we have the compensation bound $Com(T) \leq \sum_i \frac{8}{\Delta_i - \varepsilon} \log T + N \log T + O(\frac{1}{\varepsilon^4}) + F_2(\boldsymbol{\mu})$. $\square$

5   Experiments

In this section, we present experimental results for the three algorithms, i.e., the UCB policy, the modified ε-greedy policy and the modified TS policy. We also compare our modified TS policy with origin TS policy to evaluate their difference. In our experiments, there are a total of nine arms with expected reward vector $\boldsymbol{\mu} = [0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1]$. We run the game for $T = 10000$ time steps. The experiment runs for 1000 times and we take the average over these results. The "-R" represents the regret of that policy, and "-C" represents the compensation.

The comparison of the three policies in this paper is shown in Figure 1. We can see that modified-TS performs best in both regret and compensation, compared to other algorithms. As for the modified ε-greedy policy, when the parameter ϵ is chosen properly, it can also achieve a good performance. In our experiment, we choose $\epsilon = 20$.

In Figure 2, we see that modified-TS performs better than TS in both compensation and regret, which means that our modification is effective. Figure 3 shows the different performance of the modified ε-greedy policies with different ϵ values. Here we choose ϵ to be 10,15 and 20. From the experiments, we see the trade-off between regret and compensation: low compensation leads to high regret, and high compensation leads to low regret.

6   Conclusion

We propose and study the KCMAB problem where a controller offers compensation to incentivize players for arm exploration. We first provide the analysis of a compensation lower bound achieved by regret-minimizing algorithms. Then, we consider three algorithms, namely, UCB, modified ε-greedy and modified TS. We show that all three algorithms achieve good regret bounds, while keeping

order-optimal compensation. We also conduct experiments and the results validate our theoretical findings.

References

[1] S. Agrawal and N. Goyal. Further optimal regret bounds for thompson sampling. In *AISTATS*, pages 99–107, 2013.

[2] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[3] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The non-stochastic multi-armed bandit problem. *Siam Journal on Computing*, 32(1):48–77, 2002.

[4] D. A. Berry and B. Fristedt. *Bandit problems: sequential allocation of experiments (Monographs on statistics and applied probability)*. Springer, 1985.

[5] A. Beygelzimer, J. Langford, L. Li, L. Reyzin, and R. Schapire. Contextual bandit algorithms with supervised learning guarantees. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 19–26, 2011.

[6] Y.-K. Che and J. Hörner. Recommender systems as mechanisms for social learning. *The Quarterly Journal of Economics*, 133(2):871–925, 2017.

[7] R. Combes, C. Jiang, and R. Srikant. Bandits with budgets: Regret lower bounds and optimal algorithms. *ACM SIGMETRICS Performance Evaluation Review*, 43(1):245–257, 2015.

[8] P. Frazier, D. Kempe, J. Kleinberg, and R. Kleinberg. Incentivizing exploration. In *Fifteenth ACM Conference on Economics and Computation*, pages 5–22, 2014.

[9] J. Gittins. Multi-armed bandit allocation indices. wiley-interscience series in systems and optimization. 1989.

[10] S. Kalyanakrishnan, A. Tewari, P. Auer, and P. Stone. Pac subset selection in stochastic multi-armed bandits. In *International Conference on Machine Learning*, 2012.

[11] E. Kaufmann, N. Korda, and R. Munos. Thompson sampling: an asymptotically optimal finite-time analysis. In *Proceedings of the 23rd international conference on Algorithmic Learning Theory*, pages 199–213, 2012.

[12] T. L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

[13] O.-A. Maillard and R. Munos. Adaptive bandits: Towards the best history-dependent strategy. In *International Conference on Artificial Intelligence and Statistics*, pages 570–578, 2011.

[14] Y. Mansour, A. Slivkins, and V. Syrgkanis. Bayesian incentive-compatible bandit exploration. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, pages 565–582. ACM, 2015.

[15] Y. Mansour, A. Slivkins, V. Syrgkanis, and Z. S. Wu. Bayesian exploration: Incentivizing exploration in bayesian games. *arXiv preprint arXiv:1602.07570*, 2016.

[16] Y. Papanastasiou, K. Bimpikis, and N. Savva. Crowdsourcing exploration. *Management Science*, 2017.

[17] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[18] W. 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, 1933.

[19] L. Tran-Thanh, A. C. Chapman, A. Rogers, and N. R. Jennings. Knapsack based optimal policies for budget-limited multi-armed bandits. In *AAAI*, 2012.

[20] C. J. C. H. Watkins. Learning form delayed rewards. *Ph.d.thesis Kings College University of Cambridge*, 1989.

[21] Y. Xia, H. Li, T. Qin, N. Yu, and T.-Y. Liu. Thompson sampling for budgeted multi-armed bandits. In *IJCAI*, pages 3960–3966, 2015.

Supplementary Material

The proofs of all lemmas are shown in the end of the sections they belong to.

A Proof of Theorem 1

Fact 2. *(Theorem 2 in [12]) If an algorithm guarantees $o(T^\alpha)$ regret upper bound for any $\alpha > 0$, then for any $\varepsilon > 0$, we have:*

$$\lim_{T \to \infty} \Pr[N_i(t) \geq (1 - \varepsilon)\frac{\log T}{KL(D_i, D_1)}] = 1$$

Fact 2 means that we will pull arm i for at least $(1 - \varepsilon)\frac{\log T}{KL(D_i,D_1)}$ times almost surely, i.e. there exists a function $\delta(T) \to 0$ as $T \to \infty$ such that $\Pr[N_i(T) \geq (1 - \varepsilon)\frac{\log T}{KL(D_i,D_1)}] \geq 1 - \delta(T)$.

Consider the time step when we pull arm i for the k-th time ($2 \leq k \leq (1 - \varepsilon)\frac{\log T}{KL(D_i,D_1)}$), and we use $t_i(k)$ to denote the random variable of this time slot. Since we may not pull arm i for k times until time T, we suppose that the game lasts for infinite number of times, and $t_i(k)$ can be larger than T. By the definition of $t_i(k)$, we must have $\mathbb{E}[\hat\mu_i(t_i(k))] = \mu_i$.

Then the compensation we need to pay for pulling arm i for the k-th time can be bounded as following:

$$\Pr[t_i(k) \leq T]\mathbb{E}[c(t_i(k))|t_i(k) \leq T] = \Pr[t_i(k) \leq T]\mathbb{E}[\max_j \hat\mu_j(t_i(k)) - \hat\mu_i(t_i(k))|t_i(k) \leq T]$$

$$\geq \mathbb{E}[\max_j \hat\mu_j(t_i(k)) - \hat\mu_i(t_i(k))] - \Pr[t_i(k) > T] \quad (3)$$

$$\geq \mathbb{E}[\max_j \hat\mu_j(t_i(k))] - \mathbb{E}[\hat\mu_i(t_i(k))] - \delta(T) \quad (4)$$

$$\geq \mathbb{E}[\hat\mu_1(t_i(k))] - \mathbb{E}[\hat\mu_i(t_i(k))] - \delta(T)$$

$$= \mathbb{E}[\hat\mu_1(t_i(k))] - \mu_i - \delta(T) \quad (5)$$

Eq. (3) is because that $\max_j \hat\mu_j(t_i(k)) - \hat\mu_i(t_i(k)) \leq 1$, and Eq. (4) is because that $\Pr[t_i(k) > T] \leq \delta(T)$, which is given by Fact 2.

In Theorem 1, we suppose that D_1 is a Bernoulli distribution, i.e. $\Pr[X_1(t) = 1] = \mu_1$ and $\Pr[X_1(t) = 0] = 1 - \mu_1$. Then we can use a 0-1 string to represent the history of arm 1. In this case $\hat\mu_1(t_i(k)) = \frac{\#(s)}{|s|}$, where $\#(s)$ is the number of 1s in string s. To simplify the notations, we use $z - s$ to denote the string that removes prefix s from z, and $s + z$ to denote the string given by adding prefix s to z.

We can thus rewrite $\mathbb{E}[\hat\mu_1(t_i(k))]$ as:

$$\mathbb{E}[\hat\mu_1(t_i(k))] = \sum_s p(s)\frac{\#(s)}{|s|},$$

where $p(s)$ is the probability that at time slot $t_i(k)$, the history of arm 1 forms string s. Note that this expectation is hard to evaluate, since all the arms are coupled due to the algorithms, which makes $\hat\mu_1(t_i(k))$ dependent on $t_i(k)$.

We define two events as following: $\mathcal{A}_L(s)$ is the event that the first $|s|$ feedbacks of arm 1 form string s, and $\mathcal{B}_L(s)$ is the event that at time slot $t_i(k)$, the feedbacks of arm 1 forms string s. Then $\Pr[\mathcal{A}_L(s)] = P(s, \mu_1)$, where $P(s, \mu_1) = \mu_1^{\#(s)}(1 - \mu_1)^{|s|-\#(s)}$, and $\Pr[\mathcal{B}_L(s)] = p(s)$.

In our model, we suppose that we pull each arm once in the first N time slots, then $\sum_{|s|=0} p(s) = 0$. By these two equations, we have $\sum_{|s|>0} p(s) = 1$, thus p is a probability distribution on all 0-1 strings. Since p is a probability distribution, when the first $|s|$ feedbacks of arm 1 form string s, there must be a string z such that $\mathcal{B}_L(z)$ happens and either z is prefix of s or s is prefix of z. Moreover, if z is prefix of s, we still need the next $|s| - |z|$ feedbacks from arm 1 form the string $s - z$. This

means the following equation holds:

$$\mathcal{A}_L(s) = \cup_{z:pre(z,|s|)=s}\mathcal{B}_L(z)\cup_{y\in sub(s)}(\mathcal{B}_L(y)\cap\{\text{The next } |s|-|y| \text{ feedbacks form string } s-y\}), \tag{6}$$

where $pre(z,n)$ is the prefix of z with length n, and $sub(s) = \{pre(s,j)|1 \le j \le |s|-1\}$ is the set that contains all prefixes of s but does not contain s itself.

For any $s \ne z$, we mush have $\mathcal{B}_L(s) \cap \mathcal{B}_L(z) = \emptyset$, thus

$$
\begin{aligned}
\Pr[\mathcal{A}_L(s)] &= \sum_{z:pre(z,|s|)=s}\Pr[\mathcal{B}_L(z)] + \sum_{y\in sub(s)}\Pr[\mathcal{B}_L(y)\cap\{\text{The next } |s|-|y| \text{ feedbacks forms string } s-y\}]\\
&= \sum_{z:pre(z,|s|)=s}p(z) + \sum_{y\in sub(s)}p(y)P(s-y,\mu_1)
\end{aligned}
\tag{7}
$$

Eq. (7) is because that the next $|s|-|y|$ feedbacks are independent with event $\mathcal{B}_L(y)$. From this equation, we have $P(s,\mu_1) = \sum_{z:pre(z,|s|)=s}p(z) + \sum_{y\in sub(s)}p(y)P(s-y,\mu_1)$.

To simplify the analysis, we construct p^T from p by adding the probability of $p(s)$ with $s > T$ to $p(pre(s,T))$:

$$
p^T(s) = \begin{cases}
p(s) & 1 \le |s| < T\\
p(s) + \sum_{z:pre(z,T)=s}p(z) & |s| = T\\
0 & |s| > T
\end{cases}
$$

This does not influence the Eq. (7) for all s with $|s| \le T$ (the probability mass for strings longer than T is included in its T-sized prefix). Thus $P(s,\mu_1) = \sum_{z:pre(z,|s|)=s}p^T(z)+\sum_{y\in sub(s)}p^T(y)P(s-y,\mu_1)$ still holds.

Now we can use p^T to bound $\mathbb{E}[\hat{\mu}_1(t_i(k))]$ as:

$$\mathbb{E}[\hat{\mu}_1(t_i(k))] = \sum_s p(s)\frac{\#(s)}{|s|} \ge \sum_s p^T(s)\frac{\#(s)}{|s|} - \delta(T) \tag{8}$$

Here $\sum_{|s|>T}p(s) \le \delta(T)$ holds by Fact 2 ($\cup_{|s|>T}\mathcal{B}_L(s)$ implies $N_i(T) \le k \le (1-\varepsilon)\frac{\log T}{KL(D_i,D_1)}$).

From p^T, we can then build another q^T as follows.

$$
q^T(s) = \begin{cases}
\frac{p^T(s)}{P(s,\mu_1)-\sum_{y\in sub(s)}p^T(y)P(s-y,\mu_1)} & P(s,\mu_1) - \sum_{y\in sub(s)}p^T(y)P(s-y,\mu_1) > 0\\
1 & P(s,\mu_1) - \sum_{y\in sub(s)}p^T(y)P(s-y,\mu_1) = 0
\end{cases}
$$

One can check that $q^T(s) = 1$ for any $|s| = T$ and $0 \le q^T(s) \le 1$ for any $1 \le |s| < T$.

Lemma 1. *We can get the value of p^T from q^T by the following equation,*

$$p^T(s) = P(s,\mu_1)\prod_{j=1}^{|s|-1}(1 - q^T(pre(s,j)))q^T(s) \tag{9}$$

Lemma 1 means that every possible p^T have a unique q^T match and vice versa. Then we can consider a set of $q^T(s)$ such that the corresponding $p^T(s)$ minimizes $\sum_{1\le|s|\le T}p^T(s)\frac{\#(s)}{|s|}$. We write $emp(q^T,\mu_1,T)$ to denote the corresponding value $\sum_{1\le|s|\le T}p^T(s)\frac{\#(s)}{|s|}$. Then for given q^T, we can use Algorithm 5 to compute $emp(q^T,\mu_1,T)$.

Proposition 1. *Algorithm 5 returns the value $emp(q^T,\mu_1,T)$ correctly.*

12

```
1: Input: $q^T, \mu_1, T$.
2: for $s$ with $|s| = T$ do
3:     $g(s) \leftarrow \frac{\#(s)}{|s|}$.
4: end for
5: for $t = T - 1, \cdots, 1$ do
6:     for $s$ with $|s| = t$ do
7:         $g(s) \leftarrow q^T(s)\frac{\#(s)}{|s|} + (1 - q^T(s))(\mu g(S + "1") + (1 - \mu)g(S + "0"))$
8:     end for
9: end for
10: $emp(q^T, \mu_1, T) = \mu g("1") + (1 - \mu)g("0")$
11: Output: $emp(q^T, \mu_1, T)$.
```

Algorithm 5: Calculate $emp(q^T, \mu_1, T)$

```
1: Input: $\mu_1, T$.
2: for $a = 0, 1, \cdots, T$ do
3:     $f(a, T - a) \leftarrow \frac{a}{T}$.
4: end for
5: for $t = T - 1, \cdots, 1$ do
6:     for $a = 0, 1, \cdots, t$ do
7:         $f(a, t - a) \leftarrow \min\{\frac{a}{t}, \mu f(a + 1, t - a) + (1 - \mu)f(a, t - a + 1)\}$
8:     end for
9: end for
10: $f(0, 0) = \mu f(1, 0) + (1 - \mu)f(0, 1)$
11: Output: $DP(\mu_1, T) = f(0, 0)$
```

Algorithm 6: DP stopping policy

To find q^T that minimizes $emp(q^T, \mu_1, T)$, we introduce a Dynamic Programming policy as Algorithm 6. It starts by setting $f(s) = \frac{\#(s)}{|s|}$ for all strings with $|s| = T$. After that, if all strings s with $|s| = k$ have their $f(s)$, we start to consider s with $|s| = k - 1$. For any $|s| = k - 1$, the DP policy will check whether it is good or not to stop at s, i.e. only if $\frac{\#(s)}{|s|}$ is smaller than $\mu_1 f(s + "1") + (1 - \mu_1)f(s + "0")$, we choose to set $q^T(s) = 1$ and $f(s) = \frac{\#(s)}{|s|}$, otherwise we set $q^T(s) = 0$ and $f(s) = \mu_1 f(s + "1") + (1 - \mu_1)f(s + "0")$.

Intuitively, the DP policy is the best one can do, which is shown in the following lemma.

Lemma 2. *For any given q^T, $emp(q^T, \mu_1, T) \geq DP(\mu_1, T)$, where $DP(\mu_1, T)$ is the output value of Algorithm 6.*

Now from Eq. (8) and Lemma 2, we have:

$$\mathbb{E}[\hat{\mu}_1(t_i(k))] \geq DP(\mu_1, T) - \delta(T) \tag{10}$$

Then we need a lower bound on $DP(\mu_1, T)$, which is given in the following lemma.

Lemma 3. *For any $\mu_1 \geq 0.9$ and $T \geq 1$, we have $DP(\mu_1, T) \geq \frac{\mu_1}{2}$*

From Lemma 3, Eq. (10) and Eq. (5), we know that when $\mu_1 \geq 0.9, 0.2 \geq \mu_2 \geq \mu_3 \geq \cdots \geq \mu_N$, $\mathbb{E}[c(t_i(k))] \geq DP(\mu_1, T) - \mu_i - 2\delta(T) \geq \frac{\Delta_i}{4} - 2\delta(T)$ for any i and $2 \leq k \leq (1 - \varepsilon)\frac{\log T}{KL(D_i, D_1)}$. Thus the compensation we need to pay is $\Omega(\sum_{i=2}^{N}(\Delta_i - 2\delta(T))\frac{\log T}{KL(D_i, D_1)})$. When $T \rightarrow \infty$, we have $\delta(T) \rightarrow 0$. Therefore, the compensation is lower bounded by $\Omega(\sum_{i=2}^{N}\frac{\Delta_i \log T}{KL(D_i, D_1)})$.

A.1 Proof of Lemma 1

We prove this lemma by induction.

13

For the strings s with $|s| = 1$, since there is no string in $sub(s)$, we have $q^T(s) = \frac{p^T(s)}{P(s,\mu_1)}$ by definition of q^T. Thus Eq. (9) holds.

If for all strings s with $|s| \leq k$, Eq. (9) holds. Then consider a string s with $|s| = k+1$, we choose z as the longest string in $sub(s)$ such that $q^T(z) > 0$.

If such z does not exist. Then by definition of $q^T(z)$, we know $p^T(z) = 0$ for all $z \in sub(s)$. Thus we have $P(s,\mu_1)\prod_{j=1}^{|s|-1}(1 - q^T(pre(s,j)))q^T(s) = P(s,\mu_1)q^T(s) = p^T(s)$ holds.

When such z exists, let $x = s - z$, and then we have

$$P(s,\mu_1)\prod_{j=1}^{|s|-1}(1 - q^T(pre(s,j)))q^T(s)$$

$$= \left(P(z,\mu_1)\prod_{j=1}^{|z|-1}(1 - q^T(pre(z,j)))\right)(1 - q^T(z))q^T(s)P(x,\mu_1)$$

$$= \frac{p^T(z)}{q^T(z)}(1 - q^T(z))q^T(s)P(x,\mu_1) \tag{11}$$

Eq. (11) is because that by induction, Eq. (9) holds for any $|z| \leq k$.

If $P(z,\mu_1) - \sum_{y\in sub(z)}p^T(y)P(z-y,\mu_1) = 0$, then we must have $p^T(z) = 0$, thus $P(s,\mu_1) - \sum_{y\in sub(s)}p^T(y)P(s-y,\mu_1) \leq P(x,\mu_1)\left(P(z,\mu_1) - \sum_{y\in sub(z)}p^T(y)P(z-y,\mu_1)\right) = 0$, which means $p^T(s) = 0$. On the other hand, $P(z,\mu_1) - \sum_{y\in sub(z)}p^T(y)P(z-y,\mu_1) = 0$ means $q^T(z) = 1$, thus Eq. (9) holds.

If $P(z,\mu_1) - \sum_{y\in sub(z)}p^T(y)P(z-y,\mu_1) > 0$, then

$$\frac{p^T(z)}{q^T(z)}(1 - q^T(z))q^T(s)P(x,\mu_1)$$

$$= p^T(z)\frac{1 - q^T(z)}{q^T(z)}q^T(s)P(x,\mu_1)$$

$$= p^T(z)\left(\frac{P(z,\mu_1) - \sum_{y\in sub(z)}p^T(y)P(z-y,\mu_1)}{p^T(z)} - 1\right)q^T(s)P(x,\mu_1)$$

$$= \left(P(z,\mu_1) - \sum_{y\in sub(z)}p^T(y)P(z-y,\mu_1) - p^T(z)\right)q^T(s)P(x,\mu_1)$$

$$= \left(P(s,\mu_1) - \sum_{y\in sub(s)}p^T(y)P(s-y,\mu_1)\right)q^T(s) \tag{12}$$

$$= p^T(s)$$

Eq. (12) is because that for all $s' \in sub(s)$ and $|s'| > |z|$, $q^T(s') = 0$ implies $p^T(s') = 0$.

By induction, we finish the proof of Lemma 1.

A.2 Proof of Lemma 2

We use induction to prove that $emp(q^T, \mu_1, T) \geq DP(\mu_1, T)$.

For all s with size $|s| = T$, we can see that $f(\#(s), |s| - \#(s)) = \frac{\#(s)}{|s|} \leq g(s)$, where $f(\#(s), |s| - \#(s))$ is the value in Algorithm 6 with input μ_1, T.

If for all s with size $|s| = k$, we have $f(\#(s), |s| - \#(s)) \leq g(s)$, then consider any s' with size $|s'| = k - 1$.

$$f(\#(s'), |s'| - \#(s')) = \min\{\frac{\mathbf{1}(s')}{|s'|}, \mu f(\#(s) + 1, |s| - \#(s)) + (1 - \mu)f(\#(s), |s| - \#(s) + 1)\}$$
$$\leq \min\{\frac{\mathbf{1}(s')}{|s'|}, \mu g(s' + "1") + (1 - \mu)g(s' + "0")\}$$
$$\leq g(s')$$

Thus by induction, we have $f(0, 1) \leq g("0")$ and $f(1, 0) \leq g("1")$, which means $emp(q^T, \mu_1, T) \geq DP(\mu_1, T)$.

A.3 Proof of Lemma 3

We calculate this by summing over the difference between $DP(\mu_1, T)$ and $DP(\mu_1, T + 1)$. Let $f^{\mu_1,T}(a, b) - f^{\mu_1,T+1}(a, b) = \delta_{a,b}^{\mu_1,T}$, where $f^{\mu,T}(a, b)$ is the value of $f(a, b)$ when inputting μ, T into Algorithm 6.

First consider the case that $a + b = T$, $f^{\mu_1,T}(a, b) = \frac{a}{T}$, while $f^{\mu_1,T+1}(a, b) = \min\{\frac{a}{T}, \mu_1 \frac{a+1}{T+1} + (1 - \mu_1)\frac{a}{T+1}\}$.

Since $\mu_1 \frac{a+1}{T+1} + (1 - \mu_1)\frac{a}{T+1} = \frac{a+\mu_1}{T+1} = \frac{a}{T} + \frac{\mu_1 - a/T}{T+1} = \frac{a}{T} + \frac{\mu_1 T - a}{T(T+1)}$. When $a > \mu_1 T$, we have $\delta_{a,b}^{\mu_1,T} = \frac{a - \mu_1 T}{T(T+1)}$; otherwise $\delta_{a,b}^{\mu_1,T} = 0$. Thus $\delta_{a,b}^{\mu_1,T} = \frac{1}{T(T+1)}(a - \mu_1 T)^+$.

Then we consider the case that $a + b = t < T$. By definition, $f^{\mu_1,T}(a, b) = \min\{\frac{a}{t}, \mu_1 f^{\mu_1,T}(a + 1, b) + (1 - \mu_1)f^{\mu_1,T}(a, b + 1)\}$. Thus

$$
\begin{aligned}
\delta_{a,b}^{\mu_1,T} &= f^{\mu_1,T}(a, b) - f^{\mu_1,T+1}(a, b) \\
&= \min\{\frac{a}{t}, \mu_1 f^{\mu_1,T}(a + 1, b) + (1 - \mu_1)f^{\mu_1,T}(a, b + 1)\} \\
&\quad - \min\{\frac{a}{t}, \mu_1 f^{\mu_1,T+1}(a + 1, b) + (1 - \mu_1)f^{\mu_1,T+1}(a, b + 1)\} \\
&\leq (\mu_1 f^{\mu_1,T}(a + 1, b) + (1 - \mu_1)f^{\mu_1,T}(a, b + 1)) \\
&\quad - (\mu_1 f^{\mu_1,T+1}(a + 1, b) + (1 - \mu_1)f^{\mu_1,T+1}(a, b + 1)) \quad\quad (13) \\
&= \mu_1 \delta_{a+1,b}^{\mu_1,T} + (1 - \mu_1)\delta_{a,b+1}^{\mu_1,T}
\end{aligned}
$$

Eq. (13) is because of the fact that $f^{\mu_1,T}(a, b) \geq f^{\mu_1,T+1}(a, b)$ for any given (a, b).

This implies :

$$
\begin{aligned}
\delta_{0,0}^{\mu_1,T} &\leq \mu_1 \delta_{1,0}^{\mu_1,T} + (1 - \mu_1)\delta_{0,1}^{\mu_1,T} \\
&\leq \mu_1^2 \delta_{2,0}^{\mu_1,T} + 2\mu_1(1 - \mu_1)\delta_{1,1}^{\mu_1,T} + (1 - \mu_1)^2 \delta_{0,2}^{\mu_1,T} \\
&\leq \sum_{a+b=3} \binom{3}{a} \mu_1^a (1 - \mu_1)^b \delta_{a,b}^{\mu_1,T} \\
&\leq \cdots \\
&\leq \sum_{a+b=T} \binom{T}{a} \mu_1^a (1 - \mu_1)^b \delta_{a,b}^{\mu_1,T} \\
&= \mathbb{E}[\frac{1}{T(T+1)}(a - \mu_1 T)^+] \\
&= \frac{1}{T(T+1)}\mathbb{E}[(a - \mu_1 T)^+]
\end{aligned}
$$

The expectation is taken over a binomial distribution $a \sim Binomial(T, \mu_1)$.

Since $\mathbb{E}[a] = \mu_1 T$, we have that:

$$\mathbb{E}[(a - \mu_1 T)^+] = \frac{1}{2}\mathbb{E}[|a - \mu_1 T|] \leq \frac{1}{2}\sqrt{\mathbb{E}[(a - \mu_1 T)^2]} = \frac{1}{2}\sqrt{T\mu_1(1 - \mu_1)},$$

which leads to the upper bound $\delta_{0,0}^{\mu_1,T} \leq \frac{\sqrt{\mu_1(1-\mu_1)}}{2(T+1)\sqrt{T}}$.

Thus,

$$
\begin{aligned}
DP(\mu_1, T) &\geq DP(\mu_1, 1) - \sum_{T=1}^{\infty} \frac{\sqrt{\mu_1(1-\mu_1)}}{2(T+1)\sqrt{T}} \\
&\geq \mu_1 - \frac{\sqrt{\mu_1(1-\mu_1)}}{2} \sum_{T=1}^{\infty} \frac{1}{T^{3/2}} \\
&\geq \mu_1 - \frac{\sqrt{\mu_1(1-\mu_1)}}{2}\left(1 + \int_1^{\infty} \frac{1}{T^{3/2}} dT\right) \\
&= \mu_1 - \frac{\sqrt{\mu_1(1-\mu_1)}}{2}(1 + 2) \\
&= \mu_1 - \frac{3\sqrt{\mu_1(1-\mu_1)}}{2}
\end{aligned}
$$

When $\mu_1 \geq 0.9$, $\frac{3\sqrt{\mu_1(1-\mu_1)}}{2} \leq 0.45$, thus $DP(\mu_1, T) \geq \mu_1 - 0.45 \geq \frac{\mu_1}{2}$.

B   Proof of Theorem 1 when T is unknown

This proof is suggested by an anonymous reviewer of our paper during the review process. We thank the reviewer for the idea.

Proposition 2. *In KCMAB, if an algorithm guarantees an $o(T^\alpha)$ regret upper bound for any T and any $\alpha > 0$, then the algorithm must pay $\Omega\left(\sum_{i=2}^{N} \frac{\Delta_i \log T}{KL(D_i, D_1)}\right)$ for compensation.*

Choose $N_1^*(\varepsilon) = \frac{9}{2\Delta_2^2} \log \frac{9}{\varepsilon\Delta_2^2}$, and $N_1^{**}(\varepsilon)$ be the time step such that with probability $1 - \frac{\varepsilon}{2}$, $N_1(t) > \frac{t}{2}$ for any $t \geq N_1^{**}(\varepsilon)$. Notice that $N_1^{**}(\varepsilon)$ must exists since it does not depend on T and the algorithm has $o(T)$ regret in expectation.

Now choose $T_1^*(\varepsilon) = \max\{2N_1^*(\varepsilon), N_1^{**}(\varepsilon)\}$. Note that $T_1^*(\varepsilon)$ does not depend on T as well. The probability that $\exists t > T_1^*(\varepsilon)$, with $\hat{\mu}_1(t) < \mu_1 - \frac{\Delta_2}{3}$ can be upper bounded by:

$$
\begin{aligned}
&\Pr[\exists t > T_1^*(\varepsilon), \hat{\mu}_1(t) < \mu_1 - \frac{\Delta_2}{3}] \\
\leq\ & \Pr[N_1(T_1^*(\varepsilon)) \leq N_1^*(\varepsilon)] + \Pr[\{N_1(T_1^*(\varepsilon)) > N_1^*(\varepsilon)\} \wedge \{\exists t > T_1^*(\varepsilon), \hat{\mu}_1(t) < \mu_1 - \frac{\Delta_2}{3}\}] \\
\leq\ & \Pr[N_1(T_1^*(\varepsilon)) \leq N_1^*(\varepsilon)] + \Pr[\exists n > N_1^*(\varepsilon) \text{ s.t. } N_1(t) = n, \hat{\mu}_1(t) < \mu_1 - \frac{\Delta_2}{3}] \\
\leq\ & \Pr[N_1(T_1^*(\varepsilon)) \leq N_1^*(\varepsilon)] + \sum_{n > N_1^*(\varepsilon)} \left(\Pr[N_1(t) = n, \hat{\mu}_1(t) < \mu_1 - \frac{\Delta_2}{3}]\right) \qquad (14)
\end{aligned}
$$

The first term in (14) has upper bound $\frac{\varepsilon}{2}$ by definition of $N_1^{**}(\varepsilon)$ and $T_1^*(\epsilon)$.

As for the second term in (14), notice that $\{N_1(t) = n, \hat{\mu}_1(t) < \mu_1 - \frac{\Delta_2}{3}\}$ implies the first n feedbacks of arm 1 have an empirical mean less than $\mu_1 - \frac{\Delta_2}{3}$. By Chernoff Bound, $\Pr[N_1(t) = n, \hat{\mu}_1(t) < \mu_1 - \frac{\Delta_2}{3}] \leq \exp(-2n\Delta_2^2/9)$. Therefore $\sum_{n > N_1^*(\varepsilon)} \Pr[N_1(t) = n, \hat{\mu}_1(t) < \mu_1 - \frac{\Delta_2}{3}] \leq \frac{9}{2\Delta_2^2} \exp(-2N_1^*(\varepsilon)\Delta_2^2/9) \leq \frac{\varepsilon}{2}$.

This means that with probability at least $1 - \varepsilon$, for any $t > T_1^*(\varepsilon)$, $\hat{\mu}_1(t) \geq \mu_1 - \frac{\Delta_2}{3}$.

Similarly, for any sub-optimal arm i, we can find $T_i^*(\varepsilon)$ such that with probability $1 - \varepsilon$, $\hat{\mu}_i(t) \leq \mu_i + \frac{\Delta_2}{3}$ for any $t > T_1^*(\varepsilon)$. The only difference in this argument is that we need to use Fact 2 instead of the fact that the algorithm has $o(T)$ regret in expectation.

Let $T^*(\varepsilon) = \max_i T_i^*(\varepsilon)$. We know that after $T^*(\varepsilon)$, with probability at least $1 - N\varepsilon$, pulling arm i once needs at least $\hat{\mu}_1(t) - \hat{\mu}_i(t) \geq \frac{\Delta_i}{3}$ for compensation.

Before time $T^*(\varepsilon)$, every arm can be pulled for at most $T^*(\varepsilon)$ time steps. As T goes to infinity, by Fact 2, every sub-optimal arm i needs to be pulled for at least $(1 - \varepsilon)\frac{\log T}{KL(D_i, D_1)}$ times. Thus the player needs to pay at least $\left((1 - \varepsilon)\frac{\log T}{KL(D_i, D_1)} - T^*(\varepsilon)\right) \times \frac{\Delta_i}{3}$ for compensation on arm i until time T.

Taking T going to infinity and setting $\varepsilon = \frac{1}{2N}$, since $T^*(\varepsilon)$ does not depend on T, the total compensation is

$$\Omega\left(\sum_{i=2}^{N} \frac{\Delta_i \log T}{KL(D_i, D_1)}\right)$$

C Proof of Theorem 2

After the first N time steps, every arm i has $\hat{\mu}_i(t) = M_i(t)/N_i(t)$.

Notice that we always choose the arm i with maximum value $\hat{\mu}_i(t) + r_i(t)$. Thus the arm $a(t)$ satisfies the following inequality:

$$\hat{\mu}_{a(t)}(t) + r_{a(t)}(t) \geq \max_j \hat{\mu}_j(t) + r_j(t) \geq \max_j \hat{\mu}_j(t)$$

This means that we need to pay at most $r_{a(t)}(t)$ for compensation.

For each sub-optimal arm $i \neq 1$, if it is chosen at time t, then we must have $\hat{\mu}_i(t) + r_i(t) \geq \hat{\mu}_1(t) + r_1(t)$. Since $\mu_1 = \mu_i + \Delta_i$, we have:

$$\hat{\mu}_i(t) + \mu_1 + 2r_i(t) \geq \mu_i + r_i(t) + \hat{\mu}_1(t) + r_1(t) + \Delta_i$$

This implies that one of the following three events must happen:

$$\mathcal{A}_i^{UCB}(t) = \{\hat{\mu}_i(t) \geq \mu_i + r_i(t)\}$$

$$\mathcal{B}^{UCB}(t) = \{\mu_1 \geq \hat{\mu}_1(t) + r_1(t)\}$$

$$\mathcal{C}_i^{UCB}(t) = \{2r_i(t) \geq \Delta_i\}$$

Thus $\mathbb{E}[N_i(T)] \leq \sum_{t=1}^{T}(\Pr[\mathcal{A}_i^{UCB}(t)] + \Pr[\mathcal{B}^{UCB}(t)] + \Pr[\mathcal{C}_i^{UCB}(t)])$.

Notice that $r_i(t) = \sqrt{\frac{2\log t}{N_i(t)}}$, then if $N_i(t) > \frac{8\log T}{\Delta_i^2}$, event $\mathcal{C}_i^{UCB}(t)$ can not happen, which means $\sum_{t=1}^{T} \Pr[\mathcal{C}_i^{UCB}(t)] \leq \frac{8\log T}{\Delta_i^2}$. As for events $\mathcal{A}_i^{UCB}(t)$ and $\mathcal{B}^{UCB}(t)$, we have the following fact given by Chernoff-Hoeffding's inequality.

Fact 3. *For any arm i, we have:*

$$\sum_{t=1}^{T} \Pr[\hat{\mu}_i(t) \geq \mu_i + r_i(t)] \leq \frac{1}{t^2}$$

$$\sum_{t=1}^{T} \Pr[\hat{\mu}_i(t) \leq \mu_i - r_i(t)] \leq \frac{1}{t^2}$$

By Fact 3, we have $\sum_{t=1}^{T}(\Pr[\mathcal{A}_i^{UCB}(t)] + \Pr[\mathcal{B}^{UCB}(t)]) \le \frac{\pi^2}{3}$.

If arm i has been pulled for $N_i(T)$ times, we need to pay compensation for at most $\sum_{k=1}^{N_i(T)} \sqrt{\frac{2 \log T}{k}} \le \sqrt{8 N_i(T) \log T}$, then

$$Com_i(T) \le \mathbb{E}_{N_i(T)}\left[\sqrt{8 N_i(T) \log T}\right] \le \sqrt{8 \mathbb{E}[N_i(T)] \log T} \le \frac{8 \log T}{\Delta_i} + \frac{\pi^2}{3}.$$

As for arm 1, we can see that when $N_1(t) = \max_i N_i(t)$ and $a(t) = 1$, we do not need to pay compensation. The reason is that $\hat{\mu}_1(t) + r_1(t) \ge \hat{\mu}_i(t) + r_i(t)$ and $r_i(t) \ge r_1(t)$ imply $\hat{\mu}_1(t) \ge \hat{\mu}_i(t)$.

Thus, let $N_1'(T) = \max_{i \neq 1} N_i(T)$, we know that we only need to pay compensation for pulling arm 1 when $N_1(t) \le N_1'(T)$.

Notice that we have $\mathbb{E}[N_1'(T)] \le \sum_{i \neq 1} \mathbb{E}[N_i(T)] \le \sum_{i \neq 1} \frac{8 \log T}{\Delta_i^2} + \frac{N\pi^2}{3}$. Thus, the compensation we need to pay on arm 1 satisfies

$$Com_1(T) \le \mathbb{E}_{N_1'(T)}\left[\sqrt{8 N_1'(T) \log T}\right] \le \sqrt{8 \mathbb{E}[N_1'(T)] \log T} \le \sum_{i \neq 1} \frac{8 \log T}{\Delta_i} + \frac{N\pi^2}{3}.$$

Summing over all sub-optimal arms and the optimal arm, in Algorithm 1, we have

$$Com(T) = \sum_i Com_i(T) \le \sum_{i=2}^{N} \frac{16 \log T}{\Delta_i} + \frac{2N\pi^2}{3}.$$

D   Proof for Theorem 3

Notice that only if we choose to explore arm j, we need to pay the compensation. Now consider the expected compensation we need to pay on explore arm j for the k-th time, which can be written as $\mathbb{E}[\max_i(\hat{\mu}_i(t_j^\varepsilon(k)) - \hat{\mu}_j(t_j^\varepsilon(k)))]$.

Then we can have:

$$\mathbb{E}[\max_i(\hat{\mu}_i(t_j^\varepsilon(k)) - \hat{\mu}_j(t_j^\varepsilon(k)))]$$

$$= \mathbb{E}[\max_i(\hat{\mu}_i(t_j^\varepsilon(k)) - \mu_i + \mu_i - \mu_j + \mu_j - \hat{\mu}_j(t_j^\varepsilon(k)))] \tag{15}$$

$$\le \mathbb{E}[\max_i(\hat{\mu}_i(t_j^\varepsilon(k)) - \mu_i)] + \mathbb{E}[\max_i(\mu_i - \mu_j)] + \mathbb{E}[(\mu_j - \hat{\mu}_j(t_j^\varepsilon(k)))]$$

$$= \mathbb{E}[\max_i(\hat{\mu}_i(t_j^\varepsilon(k)) - \mu_i)] + \Delta_j + \mathbb{E}[(\mu_j - \hat{\mu}_j(t_j^\varepsilon(k)))] \tag{16}$$

$$= \mathbb{E}[\max_i(\hat{\mu}_i(t_j^\varepsilon(k)) - \mu_i)] + \Delta_j \tag{17}$$

Eq. (16) is because that $\mathbb{E}[\max_i(\mu_i - \mu_j)] = \max_i(\mu_i - \mu_j) = \mu_1 - \mu_j = \Delta_j$, and Eq. (17) is because that whether we choose to explore arm j are independent with the its observations.

Now we consider the value $\mathbb{E}[\max_i(\hat{\mu}_i(t) - \mu_i)]$. It is upper bounded by $\sum_i \mathbb{E}[(\hat{\mu}_i(t) - \mu_i)^+]$. When j is explored for k times, we know every arm must have be chosen for at least k times. Notice that these feedbacks are independent with whether we choose to explore as well. Thus, we have:

$$\sum_i \mathbb{E}[(\hat{\mu}_i(t_j^\varepsilon(k)) - \mu_i)^+] = \frac{1}{2} \sum_i \mathbb{E}[|\hat{\mu}_i(t_j^\varepsilon(k)) - \mu_i|]$$

$$\le \frac{1}{2} \sum_i \sqrt{\mathbb{E}[(\hat{\mu}_i(t_j^\varepsilon(k)) - \mu_i)^2]}$$

$$\le \frac{1}{2} \sum_i \sqrt{\frac{1}{4k}}$$

$$= \frac{N}{4\sqrt{k}}$$

Suppose arm j has been exploration for $n_j(T)$ times until time T. Then

$$\sum_{k=1}^{n_j(T)} \mathbb{E}[\max_i(\hat{\mu}_i(t_j^\varepsilon(k)) - \hat{\mu}_j(t_j^\varepsilon(k)))] \leq \sum_{k=1}^{n_j(T)} \left(\frac{N}{4\sqrt{k}} + \Delta_j\right) \leq n_j(T)\Delta_j + \frac{N}{2}\sqrt{n_j(T)}$$

Notice that when $\epsilon = \frac{cN}{\Delta_2^2}$, $\mathbb{E}[n_j(T)] = \frac{c\log T}{\Delta_2^2}$, then $\mathbb{E}[n_j(T)\Delta_j + \frac{N}{2}\sqrt{n_j(T)}] \leq \frac{c\Delta_j\log T}{\Delta_2^2} + \frac{N}{2\Delta_2}\sqrt{c\log T}$. Thus the total compensation is upper bounded by:

$$\sum_{i=2}^{N} \frac{c\Delta_i\log T}{\Delta_2^2} + \frac{N^2}{2\Delta_2}\sqrt{c\log T}$$

E   Proof for Theorem 4

We first give four important lemmas, which come from the analysis of TS policy in previous works [1, 11]. Their proofs can be modified slightly to work for our Algorithm 3.

Lemma 4. *(Theorem 1 in [1]) In Algorithm 3, summing over all possible rounds $(t, t+1)$, we have that for all $i \neq 1$ and $\varepsilon < \Delta_i$:*

$$\sum_{(t,t+1)} \Pr[a_2(t+1) = i] = \frac{2}{(\Delta_i - \varepsilon)^2}\log T + O\left(\frac{1}{\varepsilon^4}\right)$$

Lemma 5. *(Proposition 1 in [11]) In Algorithm 3, summing over all possible rounds $(t, t+1)$, we have that*

$$\sum_{(t,t+1)} \Pr[N_1(t) \leq t^b] \leq C(\boldsymbol{\mu})$$

holds for some constant $b = b(\boldsymbol{\mu}) \in (0, 1)$.

Lemma 6. *(Lemma 2 in [1]) In Algorithm 3, summing over all possible rounds $(t, t+1)$, for any $i \neq 1$, we have that*

$$\sum_{(t,t+1)} \Pr[a_1(t) = i, \hat{\mu}_i(t) \geq \mu_i + \frac{\Delta_i}{2}] \leq \frac{4}{\Delta_i^2} + 1$$

Lemma 7. *In Algorithm 3,*

$$\forall i, \Pr[|\theta_i(t) - \hat{\mu}_i(t)| \geq r_i(t)] \leq \frac{1}{t^2}$$

where $r_i(t) = \sqrt{\frac{2\log t}{N_i(t)}}$.

We do not provide the proofs of Lemma 4, Lemma 5 and Lemma 6 since their proofs are almost the same as in [1] and [11]. As for Lemma 7, although the proof is similar, the statement is not the same. We provide its proof in the end of this section.

Firstly, we analyze the regret bound of Algorithm 3.

Lemma 4 shows that the regret during sample steps are bounded, now we come to the regret during empirical steps.

We use the following four events to help our analysis:

$$\mathcal{A}_i^{TS}(t) = \{\hat{\mu}_i(t) \geq \mu_i + \frac{\Delta_i}{2}\}$$
$$\mathcal{B}^{TS}(t) = \{\hat{\mu}_1(t) + r_1(t) \leq \mu_1\}$$

$$\mathcal{C}_i^{TS}(t) = \{2r_1(t) \geq \Delta_i\}$$
$$\mathcal{D}^{TS}(t) = \{N_1(t) > t^b\}$$

Then:

$$
\begin{aligned}
\sum_{(t,t+1)} \Pr[a_1(t) = i] \quad \leq \quad & \sum_{(t,t+1)} \Pr[\mathcal{A}_i^{TS}(t) \cap \{a_1(t) = i\}] + \sum_{(t,t+1)} \Pr[\mathcal{B}^{TS}(t) \cap \{a_1(t) = i\}] \\
& + \sum_{(t,t+1)} \Pr[\mathcal{C}_i^{TS}(t) \cap \mathcal{D}^{TS}(t) \cap \{a_1(t) = i\}] + \sum_{(t,t+1)} \Pr[\neg\mathcal{D}^{TS}(t) \cap \{a_1(t) = i\}] \\
& + \sum_{(t,t+1)} \Pr[\neg\mathcal{A}_i^{TS}(t) \cap \neg\mathcal{B}^{TS}(t) \cap \neg\mathcal{C}_i^{TS}(t) \cap \{a_1(t) = i\}])
\end{aligned}
$$

Lemma 6 shows that $\sum_{(t,t+1)} \Pr[\mathcal{A}_i^{TS}(t) \cap \{a_1(t) = i\}] \leq \frac{4}{\Delta_i^2} + 1$. Using Fact 3, $\sum_{(t,t+1)} \Pr[\mathcal{B}^{TS}(t) \cap \{a_1(t) = i\}] \leq \sum_{(t,t+1)} \Pr[\mathcal{B}^{TS}(t)] \leq \frac{\pi^2}{6}$.

Then we consider t such that $\mathcal{C}_i^{TS}(t) \cap \mathcal{D}^{TS}(t)$ happens. By definition, we can see that $\frac{\Delta_i}{2} \leq r_1(t) = \sqrt{\frac{2\log t}{N_1(t)}} \leq \sqrt{\frac{2\log t}{t^b}}$. Thus there exists $t_i = f(i, \boldsymbol{\mu})$ such that for all $t \geq t_i$, $\Pr[\mathcal{C}_i^{TS}(t) \cap \mathcal{D}^{TS}(t)] = 0$. This implies that:

$$\sum_{(t,t+1)} \Pr[\mathcal{C}_i^{TS}(t) \cap \mathcal{D}^{TS}(t) \cap \{a_1(t) = i\}] \leq \sum_{(t,t+1)} \Pr[\mathcal{C}_i^{TS}(t) \cap \mathcal{D}^{TS}(t)] \leq t_i$$

$\sum_{(t,t+1)} \Pr[\neg\mathcal{D}^{TS}(t) \cap \{a_1(t) = i\}]$ is upper bounded by Lemma 5, which is $C(\boldsymbol{\mu})$.

$\neg\mathcal{A}_i^{TS}(t) \cap \neg\mathcal{B}^{TS}(t) \cap \neg\mathcal{C}_i^{TS}(t) \cap \{a_1(t) = i\}$ cannot happen since under the first three events we have:

$$\hat{\mu}_1(t) > \mu_1 - r_1(t) > \mu_1 - \frac{\Delta_i}{2} = \mu_i + \frac{\Delta_i}{2} > \hat{\mu}_i(t),$$

which contradict with $\{a_1(t) = i\}$.

Thus, we have that $\sum_{i=2}^N \left(\sum_{(t,t+1)} \Pr[a_1(t) = i] \right) \leq N \left(1 + \frac{\pi^2}{6}\right) + \sum_{i=2}^N \left(\frac{4}{\Delta_i^2} + t_i\right) + C(\boldsymbol{\mu}) = F_1(\boldsymbol{\mu})$ for some function F_1, and it is independent with time horizon T.

Adding the regret during sample steps, the total regret of Algorithm 3 is upper bounded by $\sum_i \frac{2\Delta_i}{(\Delta_i - \varepsilon)^2} \log T + O\left(\frac{1}{\varepsilon^4}\right) + F_1(\boldsymbol{\mu})$.

Now we consider the compensation. Notice that in empirical steps we always choose the arm with the largest empirical mean, thus we do not need to pay any compensation in this time slot. Because of this, we can focus on the compensation in sample steps. To do so, we define a event $\mathcal{E}^{TS}(t)$ as following:

$$\mathcal{E}^{TS}(t) = \{\forall i, |\theta_i(t) - \hat{\mu}_i(t)| \leq r_i(t)\}$$

Then the total compensation can be written as

$$
\mathbb{E}[\sum_{(t,t+1):t+1<T} c_i(t+1)] \leq \mathbb{E}[\sum_{(t,t+1):t+1<T} \mathbb{I}[\mathcal{E}^{TS}(t)]c_i(t+1)] + \mathbb{E}[\sum_{(t,t+1):t+1<T} \mathbb{I}[\neg\mathcal{E}^{TS}(t)]c_i(t+1)]
$$
(18)

Lemma 7 shows that $\mathbb{E}[\sum_{(t,t+1):t+1<T} \mathbb{I}[\neg\mathcal{E}^{TS}(t)]] \leq \frac{N\pi^2}{6}$, thus the second term in Eq. (18) has upper bound $\frac{N\pi^2}{6}$ as well.

Now we consider the first term in Eq. (18). Here the compensation we need to pay is $c_i(t+1) = \max_j \hat{\mu}_j(t+1) - \hat{\mu}_{a_2(t+1)}(t+1)$.

If $a_1(t) = a_2(t+1) = i$ and $\hat{\mu}_i(t+1) \geq \hat{\mu}_i(t)$, we have $c_i(t+1) = 0 < \frac{1}{N_i(t)}$.

If $a_1(t) = a_2(t+1) = i$ but $\hat{\mu}_i(t+1) < \hat{\mu}_i(t)$, then we know that $\max_j \hat{\mu}_j(t+1) \leq \hat{\mu}_i(t)$, thus $c_i(t+1) \leq \hat{\mu}_i(t) - \hat{\mu}_i(t+1) \leq \frac{1}{N_i(t)}$.

If $a_1(t) = k, a_2(t+1) = i$ and $k \neq i$, then $c_i(t+1) = \max_j \hat{\mu}_j(t+1) - \hat{\mu}_i(t+1) \leq \hat{\mu}_k(t) + \frac{1}{N_k(t)} - \hat{\mu}_i(t)$.

Thus, if $a_1(t) = a_2(t+1)$, we need to pay at most $\frac{1}{N_i(t)}$ for compensation. Otherwise, we need to pay at most $(\hat{\mu}_{a_1(t)}(t) - \hat{\mu}_{a_2(t+1)}(t)) + \frac{1}{N_{a_1(t)}(t)}$.

Notice that under event $\mathcal{E}(t)$, $\hat{\mu}_{a_2(t+1)}(t) + r_{a_2(t+1)}(t) \geq \theta_{a_2(t+1)}(t) \geq \theta_{a_1(t)}(t) \geq \hat{\mu}_{a_1(t)}(t) - r_{a_1(t)}(t)$. Thus if $a_1(t) \neq a_2(t+1)$, $c_i(t+1) \leq r_{a_1(t)}(t) + r_{a_2(t+1)}(t) + \frac{1}{N_{a_1(t)}(t)}$. Then we can treat the total compensation as following: we first pay $r_{a_1(t}(t) + \frac{1}{N_{a_1(t)}(t)}$ on arm $a_1(t)$ in the empirical step, and then pay $r_{a_2(t+1)}(t)$ on arm $a_2(t+1)$ in the sample step. By this method, we can upper bound the compensation we need to pay on pulling sub-optimal arm i as $\sum_{\tau=1}^{N_i(T)} \sqrt{\frac{2\log T}{\tau}} + \frac{1}{\tau} \leq \log T + \sqrt{8 N_i(T) \log T}$ under the event $\mathcal{E}^{TS}(t)$.

By regret analysis, $\mathbb{E}[N_i(T)] \leq \frac{2}{(\Delta_i - \varepsilon)^2} \log T + O\left(\frac{1}{\varepsilon^4}\right) + t_i + \frac{4}{\Delta_i^2} + \frac{\pi^2+6}{6} + C(\boldsymbol{\mu})$, thus the compensation $Com_i(T)$ is upper bounded by $\frac{4}{\Delta_i - \varepsilon} \log T + O\left(\frac{1}{\varepsilon^4}\right) + t_i + \frac{4}{\Delta_i^2} + \frac{\pi^2+6}{6} + C(\boldsymbol{\mu}) + \log T$.

As for arm 1, when $a_1(t) = a_2(t+1) = 1$, we only need to pay $\frac{1}{N_1(t)}$ for compensation. The expected number of time steps that we do not have $a_1(t) = a_2(t) = 1$ is at most $\sum_{i=2}^{N} \frac{2}{(\Delta_i - \varepsilon)^2} \log T + F(\boldsymbol{\mu}) + O\left(\frac{N}{\varepsilon^4}\right)$, which is given by the regret analysis. This means the compensation on pulling arm 1 is upper bounded by $\sum_{i=2}^{N} \frac{4}{\Delta_i - \varepsilon} \log T + F_1(\boldsymbol{\mu}) + O\left(\frac{N}{\varepsilon^4}\right) + \log T$.

Thus, the first term in Eq. (18) has upper bound $\sum_{i=2}^{N} \frac{8}{\Delta_i - \varepsilon} \log T + F_1(\boldsymbol{\mu}) + O\left(\frac{N}{\varepsilon^4}\right) + N \log T + \sum_{i=2}^{N} \left(t_i + \frac{4}{\Delta_i^2} + \frac{\pi^2+6}{6} + C(\boldsymbol{\mu})\right)$. After adding the upper bound $\frac{N\pi^2}{6}$ of the second term and setting $F_2(\boldsymbol{\mu}) = F_1(\boldsymbol{\mu}) + \frac{N\pi^2}{6} + \sum_{i=2}^{N} \left(t_i + \frac{4}{\Delta_i^2} + \frac{\pi^2+6}{6} + C(\boldsymbol{\mu})\right)$, we have that

$$Com(T) \leq \sum_i \frac{8}{\Delta_i - \varepsilon} \log T + N \log T + O\left(\frac{N}{\varepsilon^4}\right) + F_2(\boldsymbol{\mu}).$$

E.1 Proof of Lemma 7

Since $\theta_i(t)$ only depends on the values of $(\alpha_i(t), \beta_i(t))$ but is independent of the random history, we can fix the pair $(\alpha_i(t), \beta_i(t))$ to prove the inequality, and then the inequalities hold also for $(\alpha_i(t), \beta_i(t))$ as random variables.

$$
\begin{aligned}
\Pr[\theta_i(t) > \hat{\mu}_i(t) + r_i(t)] &= 1 - F_{\alpha_i(t), \beta_i(t)}^{Beta}(\hat{\mu}_i(t) + r_i(t)) \\
&= 1 - (1 - F_{\alpha_i(t)+\beta_i(t)-1, \hat{\mu}_i(t)+r_i(t)}^{B}(\alpha_i(t) - 1)) \quad (19) \\
&= F_{\alpha_i(t)+\beta_i(t)-1, \hat{\mu}_i(t)+r_i(t)}^{B}(\alpha_i(t) - 1) \\
&\leq F_{\alpha_i(t)+\beta_i(t)-1, \hat{\mu}_i(t)+r_i(t)}^{B}(\hat{\mu}_i(t)(\alpha_i(t) + \beta_i(t) - 1)) \\
&\leq \exp(-(\alpha_i(t) + \beta_i(t) - 1)KL(\hat{\mu}_i(t), \hat{\mu}_i(t) + r_i(t))) \quad (20) \\
&\leq \exp(-\frac{N_i(t) r_i(t)^2}{2}) \quad (21) \\
&\leq \frac{1}{t^2}
\end{aligned}
$$

Eq. (19) is given by the following Beta-Binomial Trick (Fact 4), Eq. (20) is given by Chernoff-Hoeffding Inequality, and Eq. (21) follows the fact that $KL(x, y) \geq \frac{|x-y|^2}{2}$.

Fact 4. *(Beta-Binomial Trick) Let $F_{a,b}^{Beta}(x)$ be the cdf of Beta distribution with parameters (a, b), let $F_{n,p}^{B}(x)$ be the cdf of Binomial distribution with parameters (n, p). Then for any positive integers (a, b), we have*

$$F_{a,b}^{Beta}(x) = 1 - F_{a+b-1,x}^{B}(a - 1)$$