# Achieving Fairness in Stochastic Multi-armed Bandit Problem

Vishakha Patil [*] Ganesh Ghalme[†] Vineet Nair [‡]

Y. Narahari[§]

July 19, 2019

**Abstract**

We study an interesting variant of the stochastic multi-armed bandit problem, called the Fair-SMAB problem, where each arm is required to be pulled for at least a given fraction of the total available rounds. We investigate the interplay between *learning* and *fairness* in terms of a pre-specified vector denoting the fractions of guaranteed pulls. We define a *fairness-aware regret*, called $r$-Regret , that takes into account the above fairness constraints and naturally extends the conventional notion of regret. Our primary contribution is characterizing a class of Fair-SMAB algorithms by two parameters: the unfairness tolerance and learning algorithm used as a black-box. We provide a fairness guarantee for this class that holds uniformly over time irrespective of the choice of learning algorithm. In particular, when the learning algorithm is UCB1, we show that our algorithm achieves $O(\ln T)$ $r$-Regret . Finally, we evaluate the *cost of fairness* in terms of the conventional notion of regret.

## 1 Introduction

In a classical stochastic multi-armed bandit (S-MAB) problem, a decision maker is faced with $k$ choices (henceforth referred to as *arms*). At each time $t$, a decision maker decides which choice to select (referred to as pulling an arm). Once a decision maker pulls an arm, she gets a random reward drawn from a fixed reward distribution unknown to her. The arms which are not pulled do not give any reward. The goal of a decision maker at each round is to pull an arm so that the sum of the total expected reward from $T$ pulls is maximized. The challenge faced by the decision maker is famously known in literature as the exploration vs. exploitation dilemma i.e. whether to explore the arms to find the best arm in terms of expected rewards or to pull an arm that has given the best average reward so far.

In this paper we consider Fair S-MAB, a variant of the S-MAB problem where, in addition to the above objective of maximizing the sum of the expected rewards (or equivalently minimizing the cumulative regret), the algorithm also needs to ensure that each arm is pulled for at least a given

[*] Indian Institute of Science. patilv@iisc.ac.in

[†] Indian Institute of Science. ganeshg@iisc.ac.in

[‡] Indian Institute of Science. vineet@iisc.ac.in

[§] Indian Institute of Science. narahari@iisc.ac.in

fraction of the total number of rounds, in any round. This imposes an additional constraint on the algorithm. Such a constraint will be referred to as a *fairness constraint*. The fairness constraint is specified in terms of a vector of size $k$, where each component is the minimum fraction of the total number of time steps for which the corresponding arm has to be pulled. The goal is to minimize the regret while satisfying the fairness requirement of each arm.

Such fairness constraints are natural in many real world resource allocation problems where the arms are individuals or agents competing for a common resource. In the context of the SMAB setting, fairness constraints ensure that no individual starves from the lack of opportunities irrespective of her quality. This objective, which at times is at odds with the objective of maximizing efficiency, conforms with the *veil of ignorance* doctrine of Rawls [1] wherein each individual has equal claim to the resource without the knowledge of their true qualities in original position (refer [2; 3] for detailed discussion). For concreteness, we next present several motivating examples for the work done in this paper.

**Sponsored Search:** An advertiser, characterized by a click-through rate (CTR), competes for an ad-space on a search engine such as Google, Bing, etc. In the absence of any regulatory measures to ensure equitable allocation of ad-space, the new and/or local businesses run a risk of being starved for publicity by big corporations. Fairness regulations ensure that the local businesses get required visibility on online platforms to sustain their business.

**Wireless Communication**[[4]]: Consider a wireless communication system where a central access point allocates the channel to one of the transmitters for some fixed amount of time, called a time slot. For each successful transmission, a reward is generated that depends in some way on the transmitter (e.g. the quality of information transmitted). In addition to maximizing reward, the access point also needs to guarantee a certain minimum quality of service to each transmitter irrespective of the reward it generates.

**Crowdsourcing:** Consider a crowdsourcing scenario where a central planner assigns several micro tasks to the crowdworkers (or agents). The goal is to ensure high quality work from the agents. As the agents are heterogeneous in terms of their qualities, the goal is to find the best quality agents. However, in order to induce participation from the agents, the algorithm has to ensure that each agent is guaranteed a certain number of tasks beforehand. In this work we capture this constraint in terms of the fraction of tasks to be assigned to each agent.

**Our contributions:** In this paper, we study the FAIR-SMAB problem, a variant of the SMAB problem where, in addition to the goal of maximizing expected cumulative reward, an algorithms also has to ensure that each arm is pulled for at least a given fraction of the total number of time step in any round. After formally defining the FAIR-SMAB problem, we define the notion of fairness that we use in this paper. Further, we evaluate the regret of our algorithm with the fairness-aware regret notion called $r$-Regret . This regret notion is a natural extension of the conventional notion of regret, and is defined with respect to an optimal policy that has to also satisfy the fairness constraints. We then define a class of FAIR-SMAB algorithms, called FAIR-ALG characterized by two parameters: the unfairness tolerance, and the learning algorithm used as a black-box by FAIR-ALG . We prove a fairness guarantee for FAIR-ALG that holds uniformly over time, independent of the choice of the learning algorithm. Further, when the learning algorithm is UCB1, we show that $O(\sum_{i \neq 1} \frac{\ln T}{\Delta_i})$ $r$-Regret bound can be achieved. We then evaluate the cost of fairness in FAIR-SMAB with respect to the conventional notion of regret. We conclude by providing detailed experimental results to validate our theoretical guarantees.

**Outline of the Paper:** In the next section we discuss the related work in the area of fairness in machine learning and fairness in multi-armed bandits in specific. In Section 3 we discuss the model considered in the paper. In this section we introduce the notions of $\alpha$-fairness and $r$-Regret. In Section 4 we propose T-aware algorithms that guarantee $\alpha$-fairness at the end of $T$ rounds. In Section 5 we introduce a fair learning framework which guarantees $\alpha$-fairness at any time $t$. Further, the proposed framework can be used as a blackbox for any learning algorithm. We use UCB algorithm as a plugin algorithm and show that FAIR-UCB is $r$-Regret optimal(upto problem dependent constant). In Section 6 we compare UCB algorithm with FAIR-UCB based on the conventional notion of regret. In Section 7 we show via extensive simulations the tradeoff between fairness vector $r$, and unfairness tolerance value $\alpha$. We also compare the performance of proposed FAIR-UCB with LFG algorithm proposed in [4]. We conclude the paper with Section 8 with a brief discussion on the future work.

## 2 Related Work

There has been a surge in research efforts aimed at ensuring fairness in decision making by machine learning algorithms such as classification algorithms [5; 6; 7; 8], regression algorithms [9; 10], ranking and recommendation systems [11; 12; 13; 14; 15], online learning algorithms [4; 16; 17], etc. Here, we present the relevant work in the context of online learning, particularly in the SMAB setting.

Joseph et al. [18] propose a variant of the upper confidence bound algorithm that ensures what the authors call meritocratic fairness i.e. an arm is never preferred over a better arm irrespective of the algorithm's confidence over the mean reward of each arm. This guarantees individual fairness (see [19]) for each arm while achieving efficiency in terms of sub-linear regret. In contrast, we consider that the fairness constraints are exogenously specified and the choices made by the algorithm must adapt to these constraints so as to minimize the regret while satisfying these constraints. The work by Liu et al. [17] aims at ensuring "treatment equality", wherein similar individuals are treated similarly in the SMAB setup. This outcome based notion of fairness considers that the fairness constraints are built into the problem. Gillen et al. [16] consider individual fairness guarantees with respect to an unknown fairness metric.

A recent paper by Li et al. [4] considers a combinatorial, sleeping SMAB setup with fairness constraints similar to the ones considered in this paper. The algorithm in [4] controls the trade-off between minimizing regret and satisfying fairness constraints using a tuning parameter. In our simulations we consider the algorithm proposed in [4] as a baseline to compare the performance of our algorithm in terms of both, fairness and regret. In addition to proving a $O(\sqrt{T})$ instance independent regret bound as in [4], we also show a $O(\sum_{i\neq 1} \frac{\ln T}{\Delta_i})$ regret bound with finer dependence on the instance parameters. Further, we provide an explicit dependence of regret on fairness constraints. We provide a stronger fairness guarantee that holds uniformly over time as compared to the asymptotic fairness guarantee in [4]. A detailed comparison of the two algorithms is given in Section 7.

# 3 Model

In this section we formally define the FAIR-SMAB problem followed by defining the notions of fairness and regret which are used in this work.

## 3.1 The FAIR-SMAB Problem

An instance of a FAIR-SMAB problem is a tuple $\langle T, [k], (\mu_i)_{i \in [k]}, (r_i)_{i \in [k]} \rangle$, where $T$ is the time horizon, $[k] = \{1, 2, \ldots, k\}$ is the set of arms, $\mu_i \in [0, 1]$ represents the mean of the reward distribution $\mathcal{D}_i$ associated with arm $i$, and $(r_i)_{i \in [k]}$ represents the fairness constraint vector. Given a fairness constraint vector $r = (r_1, r_2, \ldots, r_k)$, $r_i$ is the fairness constraint for arm $i$ and denotes the minimum fraction of times arm $i$ needs to be pulled in $T$ rounds, for any $T$. Note that $r_i \in [0, 1]$ and $\sum_{i \in [k]} r_i \leq 1$.

In each round $t \leq T$, a FAIR-SMAB algorithm pulls an arm $i_t \in [k]$ and collects the reward $X_{i_t} \sim \mathcal{D}_{i_t}$. We assume that the reward distributions are $Bernoulli(\mu_i)$ for each arm $i \in [k]$. This assumption holds without loss of generality since one can reduce the SMAB problem with general distributions supported on [0,1] to an SMAB problem with Bernoulli rewards using the extension provided in [20]. Note that the true value of $\mu = (\mu_1, \mu_2, \ldots, \mu_k)$ is *unknown* to the algorithm. Throughout this paper we assume without loss of generality that $\mu_1 > \mu_2 > \ldots > \mu_k$ and arm $1$ is called the *optimal* arm.

The performance of a FAIR-SMAB algorithm is evaluated based on the regret that it incurs and the fairness guarantee that the algorithm can satisfy. In the next section, we formalize the notions of fairness and regret that we use in this paper.

## 3.2 Notion of Fairness

In the FAIR-SMAB setting, the fairness constraints are exogenously specified to the algorithm in the form of a vector $r \in [0, 1]^k$ where $\sum_{i \in [k]} r_i \leq 1$ and $r_i$ denotes the minimum fraction of times an arm $i \in [k]$ has to be pulled in $T$ rounds, for any $T$. We first begin with the definition of fairness put forth by Li et al. [4] and then define our notion of fairness.

**Definition 1** ([4])**.** A FAIR-SMAB algorithm $\mathcal{A}$ is called fair if $\liminf_{t \to \infty} \mathbb{E}_{\mathcal{A}}[r_i - \frac{N_{i,t}}{t}] \leq 0$ for all $i \in [k]$.

We refer to the above notion of fairness as *asymptotic fairness* for reasons that are clear from the definition itself. In our work we prove a stronger notion of fairness that holds uniformly over time. In addition to this, we define our fairness in terms of the *unfairness tolerance* allowed in the system which is denoted by a constant $\alpha \geq 0$ and is given to the algorithm. Formally, we introduce the following notion of fairness.

**Definition 2.** A FAIR-SMAB algorithm $\mathcal{A}$ is called *$\alpha$-fair* if $\lfloor r_i t \rfloor - N_{i,t} \leq \alpha$ for all $t \leq T$ and for all arms $i \in [k]$.

In particular, if the above guarantee holds for $\alpha = 0$, then we call the FAIR-SMAB algorithm *fair*. Note that our fairness guarantee holds uniformly over the time horizon and and for any sequence of arm pulls $(i_t)_{t \leq T}$ by the algorithm. Hence it is much stronger than the guarantee in [4] which only guarantees asymptotic fairness.

### 3.3 Notions of Regret

The performance of an SMAB algorithm is measured based on the cumulative regret it incurs in $T$ rounds. The expected regret of a SMAB algorithm is defined as the difference between the cumulative reward of the optimal policy and that of the algorithm. In the SMAB setting, the optimal policy is the one which pulls the optimal arm in every round.

**Definition 3.** The expected regret of an algorithm $\mathcal{A}$ after $T$ rounds is defined as:

$$\mathcal{R}_{\mathcal{A}}(T) = \mu_1.T - \mathbb{E}_{\mathcal{A}}\Big[\sum_{t\in[T]} X_{i_t}\Big] \tag{1}$$

The expected regret of $\mathcal{A}$ can equivalently be written in terms of the expected number of pulls of the sub-optimal arms and the expected regret incurred due to playing a sub-optimal arm. In particular, if $\Delta_i = \mu_1 - \mu_i$ and $N_{i,T}$ denotes the number of pulls of an arm $i \in [k]$ by $\mathcal{A}$ in $T$ rounds, then the expected regret of $\mathcal{A}$ after $T$ rounds is defined as:

$$\mathcal{R}_{\mathcal{A}}(T) = \sum_{i\in[k]} \Delta_i \cdot \mathbb{E}[N_{i,T}] \tag{2}$$

We call an algorithm optimal if it attains zero regret. It is easy to see that the above notion of regret does not adequately quantify the performance of a FAIR-SMAB algorithm as the optimal policy here does not account for the fairness constraints. We first characterize the fairness-aware optimal policy that we consider as a baseline.

**Observation 1.** A FAIR-SMAB algorithm $\mathcal{A}$ is optimal iff $\mathcal{A}$ satisfies $\mathbb{E}_{\mathcal{A}}[N_{i,T}] = \lfloor r_i T \rfloor - \alpha$ for all $i \neq 1$.

From Observation 1 we have that an optimal FAIR-SMAB algorithm that knows the value of $\mu$ must play sub-optimal arms (arms $i \in \{2, \dots, k\}$) exactly $\lfloor r_i \cdot T \rfloor - \alpha$ times in order to satisfy the fairness constraint and play the optimal arm (arm 1) for the rest of the rounds i.e. for $T - \sum_{i\neq 1}\lfloor r_i \cdot T \rfloor + (k-1)\alpha$ rounds. The regret of an algorithm is compared with such an optimal policy that satisfies the fairness constraints in the FAIR-SMAB setting.

**Definition 4.** Given a fairness constraint vector $r = (r_1, r_2, \dots, r_k)$ and the *unfairness tolerance* $\alpha \geq 0$, the fairness-aware $r$-Regret of a FAIR-SMAB algorithm $\mathcal{A}$ is defined as:

$$\mathcal{R}^r_{\mathcal{A}}(T) = \sum_{i\in[k]} \Delta_i \cdot \Big(\mathbb{E}[N_{i,T}] - \big(\lfloor r_i \cdot T \rfloor - \alpha\big)\Big) \tag{3}$$

Note that for a given $\alpha \geq 0$, the above definition only makes sense for $T$ large enough so that $\lfloor r_i T \rfloor - \alpha \geq 0$. An algorithm that is not aware of the true means of the reward distributions of arms, faces the exploration v/s exploitation dilemma. On one hand, it has to sufficiently explore all the arms so as to find an optimal arm and on the other, it must exploit the information gathered about mean rewards of the arms. The fairness constraints assist in exploration by guaranteeing $\lfloor r_i T \rfloor - \alpha$ samples for each arm $i$. Note that the $\lfloor r_i T \rfloor - \alpha$ pulls of any sub-optimal arm $i$ do not incur any $r$-Regret , as the optimal fair algorithm also has to pull each sub-optimal arm $i$ for $\lfloor r_i T \rfloor - \alpha$ rounds. A learning algorithm that pulls a sub-optimal arm $i$ for more than $\lfloor r_i T \rfloor - \alpha$ rounds, incurs a regret of

$\Delta_i = \mu_1 - \mu_i$ for each extra pull. The technical difficulties in designing an optimal algorithm for the FAIR-SMAB problem are the conflicting constraints on the quantity $N_{i,T} - \lfloor r_i T \rfloor$ for a sub-optimal arm $i \neq 1$: for the algorithm to be fair we want $N_{i,T} - \lfloor r_i T \rfloor$ to be at least $\alpha$ whereas to minimize the regret we want $N_{i,T} - \lfloor r_i T \rfloor$ to be close to $\alpha$.

## 4 T-aware Algorithms

An algorithm that has access to time horizon $T$ can trade-off fairness and regret more effectively. To see this, notice that in order to identify the best arm quickly it is important that an algorithm should explore the arms in the initial rounds. This observation along with Observation 1 gives us that if the arms are pulled initially to satisfy the fairness constraints, the algorithm incurs no regret and at the same time learns the rewards from each arm. In other words the algorithm incurs no regret for first $T' := \sum_{i \in [k]} r_i \cdot T$ number of rounds. If $r$ is such that the $T'$ is sufficient to explore each arm and find the best arm with high probability then one can pull the best arm for rest of the $T - T'$ rounds.[1] Guided by this intuition we propose two *T-aware* FAIR-SMAB algorithms that achieve sub-linear regret.

**Warming up – NAIVE Algorithm:** We begin with NAIVE(Algorithm 1), a variant of exploration separated policy, EXPSEP [21] that achieves sub-linear regret guarantee in terms of time horizon $T$. It is easy to see that NAIVE is fair. We show in Theorem 1 that NAIVE attains sub-linear regret (Proof in Appendix B).

**Theorem 1.** *The regret of* Naive *algorithm for* Fair-SMAB *problem,* $\mathcal{R}^r_{\text{Naive}}(T) = O((T^2 \ln T)^{1/3})$

```
Algorithm 1: NAIVE
Input: $[k], (r_i)_{i \in [k]}, \varepsilon$ where $0 \leq \varepsilon \leq 1$
Initialize: $m \leftarrow 1 - \sum_{i \in [k]} r_i,\ T' \leftarrow \varepsilon \cdot T$
- Set $n_i \leftarrow \left(r_i + \frac{m}{k}\right) \cdot T'$ for each $i \in [k]$
for $t = 1, 2, \ldots, T'$ do
    - Pull each arm $i \in [k]$ exactly $n_i$ times
end
- $j = \text{argmax}_{i \in [k]} \hat{\mu}_i(T')$
- Set $p_i = r_i + m \cdot \mathbb{1}\{i = j\}$ for each $i \in [k]$
for $t = T' + 1, \ldots, T$ do
    - $i_t \sim p$
end
```

```
Algorithm 2: T-FUCB
Input: $[k], (r_i)_{i \in [k]}$
Initialize:
$n_i \leftarrow \max\left(1, r_i \cdot T\right)$ for each $i \in [k]$
$T' = \sum_{i \in [k]} n_i$
for $t = 1, 2, \ldots, T'$ do
    - Pull each arm $i \in [k]$ exactly $n_i$ times
end
for $t = T' + 1, \ldots, T$ do
    - $i_t = \text{argmax}_{i \in [k]} \bar{\mu}_i(t)$
    - Update $\bar{\mu}_i(t + 1)$
end
```

**Figure 1**
*T-aware Algorithms*

**UCB-based Algorithm (T-FUCB ):** We propose a UCB-based T-aware fair algorithm, T-FUCB . This algorithm knows the time horizon $T$, and effectively separates the *fairness constraint satisfaction* phase and the *regret minimization* phase and achieves logarithmic regret in terms of $T$ with dependence on the values of the fairness fractions.

[1] Notice that fairness constraints are satisfied at $T'$.

T-FUCB is presented in Algorithm 2. Note that T-FUCB satisfies the fairness requirements of all arms at $T'$ itself and hence it is fair. Next we show that T-FUCB achieves logarithmic regret.

**Theorem 2.** *For* Fair-SMAB *problem,* T-FUCB *has regret* $\mathcal{R}^r_{\text{T-FUCB}}(T) = O(\ln T)$*. In particular, its r-dependent regret is given by*

$$\mathcal{R}^r_{\text{T-FUCB}}(T) \leq \Big(1 + \frac{\pi^2}{3}\Big) \cdot \sum_{i \in [k]} \Delta_i + \sum_{\substack{i \in S(T) \\ i \neq 1}} \Delta_i \cdot \left(\frac{8 \ln T}{\Delta_i^2} - r_i \cdot T\right)$$

*where* $S(T) = \Big\{ i \in [k] \;\Big|\; r_i \cdot T < \frac{8 \ln T}{\Delta_i^2} \Big\}$.

*Proof Outline.* T-FUCB does not incur any regret in the first $T'$ rounds. After $T'$, T-FUCB decides which arm to play at time $t$ based on the UCBestimates of the arms. For the UCB algorithm, we know that $\mathbb{E}[N_{i,T}] = O\big(\frac{8 \ln T}{\Delta_i^2}\big)$ for any sub-optimal arm $i \neq 1$. Hence, if for any arm we have $r_i \cdot T \geq \frac{8 \ln T}{\Delta_i^2}$, then that arm will be played for only a small constant number of times after $T'$ and hence the regret due to such an arm is bounded by a small value. On the other hand, if for some sub-optimal arm $i$, $r_i \cdot T < \frac{8 \ln T}{\Delta_i^2}$, then we incur a regret equal to $\Delta_i$ for $\mathbb{E}[N_{i,T}] - r_i \cdot T$ rounds i.e. for at most $\frac{8 \ln T}{\Delta_i^2} - r_i \cdot T$ rounds. Hence, the expected regret of T-FUCB , $\mathcal{R}^r_{\text{T-FUCB}}(T) = O(\ln T)$. Proof is provided in Appendix B. □

A FAIR-SMAB algorithm is evaluated based on two criteria: the fairness guarantee it can provide and the $r$-Regret bound of the algorithm. Our main contribution in this paper is proposing a class of FAIR-SMAB algorithms, called FAIR-ALG, characterized by two parameters: the *unfairness tolerance* , and the learning algorithm used as a black-box by FAIR-ALG. In the next section we consider an *"any-time"* version of the algorithm. We consider that the time horizon is not given as an input and hence the fairness guarantee has to be satisfied at all times.

# 5 T-agnostic Algorithms

In this section, we provide the template of our proposed FAIR-SMAB algorithm. Recall from Section 3.2 our definition of an *$\alpha$-fair* FAIR-SMAB algorithm. For an algorithm to be *$\alpha$-fair* , it needs to satisfy $\lfloor r_i t \rfloor - N_{i,t} \leq \alpha$, for all $t \leq T$, for all arms $i \in [k]$, which is equivalent to $r_i t - N_{i,t} < \alpha + 1$. In each round $t$ we're interested in the arms that could possibly violate the fairness constraints and hence look at arms $i \in [k]$ such that $\alpha < r_i(t-1) - N_{i,t-1} < \alpha + 1$. Having provided this intuition, we describe our algorithm.

In particular, when the learning algorithm LEARN(·) = UCB1, we call this algorithm FAIR-UCB. We provide the $r$-Regret bound for FAIR-UCB.

## 5.1 Theoretical Results

We begin by first analyzing the fairness guarantee provided by FAIR-ALG .

**Theorem 3.** *For a given $\alpha \geq 0$ and for any given fairness constraint vector $r = (r_1, r_2, \ldots, r_k)$ where $r_i \in [0, \frac{1}{k})$ for all $i \in [k]$,* Fair-ALG *is $\alpha$*-fair *irrespective of the choice of the learning algorithm* Learn(·)*.*

```
Algorithm 3: Fair-ALG
Input: $[k], (r_i)_{i\in[k]}, \alpha \geq 0$, Learn$(\cdot)$
Initialize:
- $N_{i,0} = 0$ for all $i \in [k]$
for $t = 1, 2, \ldots, T$ do
    - Define : $A(t) = \left\{ i \mid r_i \cdot (t-1) - N_{i,t-1} > \alpha \right\}$
    - Pull arm $i_t = \begin{cases} \mathrm{argmax}_{i\in[k]} \left( r_i \cdot (t-1) - N_{i,t-1} \right) & \text{If} A(t) \neq \emptyset \\ \textsc{Learn}(N_t, S_t) & \text{Otherwise} \end{cases}$
end
- Update parameters $N_t$ and $S_t$
```

*Proof.* After each round $t$ (and before round $t+1$), we consider the $k+1$ sets, $M_{1,t}, M_{2,t}, \ldots, M_{k,t}$, and $S_t$, as defined below:

- arm $i \in M_{j,t} \iff \alpha + \frac{(k-j)}{k} \leq r_i t - N_{i,t} < \alpha + \frac{(k-j+1)}{k}$, $\forall j \in [k]$
- arm $i \in S_t \iff r_i t - N_{i,t} < \alpha$

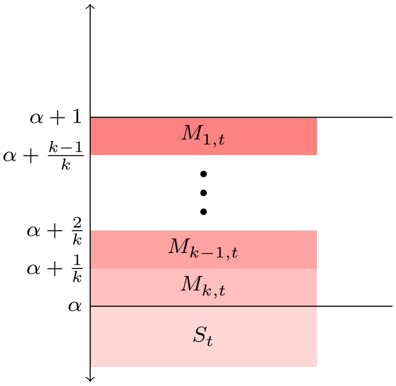

**Figure 2**
*Partition of the arms*

Let $V_{j,t} = \uplus_{\ell=1}^{j} M_{\ell,t}$, for all $j \in [k]$. Then the following lemma guarantees the fairness of the algorithm and is at the heart of the proof. It is proved immediately after the the proof of the Theorem.

**Lemma 1.** *For $t \geq 1$, we have*

1. $V_{k,t} \uplus S_t = [k]$
2. $|V_{j,t}| \leq j$*, for all* $j \in [k]$

Condition 1 in Lemma 1 ensures that at any time $t \geq 1$, the $k+1$ sets $M_{1,t}, M_{2,t}, \ldots, M_{k,t}, S_t$ form a partition of the set $[k]$ of arms. Hence the arm played at the $(t+1)$-th round by the algorithm is from one of these $k+1$ sets. As a part of the proof of Lemma 1, in Observation 2 we show that if $i_{t+1}$ is the arm played at the $(t+1)$-th round then after $t+1$ rounds $i_{t+1} \in M_{k,t+1} \uplus S_{t+1}$. Also in Observation 3 we show that if an arm $i \in M_{j,t}$ is not played in the $(t+1)$-th round then after $t+1$ rounds arm

$i \in M_{j-1,t+1} \uplus M_{j,t+1}$ for all $j \in [2,k]$. We note that the two conditions in Lemma 1 are true after the first round, and then the two observations together ensure that these conditions remain true for all $t > 1$. Hence, all arms $i \in [k]$ satisfy $r_i t - N_{i,t} < \alpha + kr < \alpha + 1$ for all $t \geq 1$, which implies $\lfloor r_i t \rfloor - N_{i,t} \leq \alpha$. In particular, we have $\lfloor r_i t \rfloor - N_{i,t} \leq \alpha$, for all $t \geq 1$, for all $i \in [k]$, which by Definition 2 proves that FAIR-ALG is *$\alpha$-fair* . □

*Proof of Lemma 1.* We begin with two complementary observations and then prove the lemma by induction.

**Observation 2.** Let $i$ be the arm pulled by FAIR-ALG in round $t+1$.

1. if $i \in S_t$, then $i \in S_{t+1}$
2. if $i \in M_{j,t}$ for some $j \in [k]$, then $i \in M_{k,t+1} \uplus S_{t+1}$

*Proof. Case 1:* $i \in S_t \implies r_i t - N_{i,t} < \alpha$. Then after round $t+1$, we have

$$\begin{aligned} r_i(t+1) - N_{i,t+1} &= r_i t + r_i - N_{i,t} - 1 \\ &< \alpha - (1 - r_i) \\ &< \alpha && \text{(Since } 1 - r_i > 0) \end{aligned}$$

$$\implies i \in S_{t+1}$$

*Case 2:* $i \in M_{j,t}$ for some $j \in [k] \implies r_i t - N_{i,t} < \alpha + \frac{(k-j+1)}{k}$. Then after round $t+1$, we have

$$\begin{aligned} r_i(t+1) - N_{i,t+1} &= r_i t + r_i - N_{i,t} - 1 \\ &< \alpha + \frac{(k-j+1)}{k} - (1 - r_i) \\ &\leq \alpha - \frac{j}{k} + \frac{1}{k} + r_i \\ &< \alpha + r_i && \text{(Since } j \geq 1) \\ &< \alpha + \frac{1}{k} && \text{(Since } r_i < \tfrac{1}{k}) \\ \implies i &\in M_{k.t+1} \uplus S_{t+1} \end{aligned}$$

□

**Observation 3.** Let $i \in [k]$ be any arm not pulled at time $t+1$.

1. If $i \in S_t$, then $i \in S_{t+1} \uplus M_{k,t+1}$
2. If $i \in M_{j,t}$ for some $j \in [k]$, then $i \in M_{j-1,t+1} \uplus M_{j,t+1}$

*Proof. Case 1:* $i \in S_t \implies r_i t - N_{i,t} < \alpha$. Then after round $t+1$, we have

$$\begin{aligned} r_i(t+1) - N_{i,t+1} &= r_i t - N_{i,t} + r_i && \text{(Since, } N_{i,t+1} = N_{i,t}) \\ &< \alpha + r_i \\ &< \alpha + \frac{1}{k} && \text{(Since } r_i < \tfrac{1}{k}) \end{aligned}$$

$$\implies i \in S_{t+1} \uplus M_{k,t+1}$$

*Case 2:* $i \in M_{j,t}$ for some $j \in [k] \implies \alpha + \frac{k-j}{k} \le r_i t - N_{i,t} < \alpha + \frac{(k-j+1)}{k}$. Then after round $t+1$, we have

$$\begin{aligned}
r(t+1) - N_{i,t+1} &= rt - N_{i,t} + r && \text{(Since, } N_{i,t+1} = N_{i,t}\text{)}\\
&< \alpha + \frac{(k-j+1)}{k} + r\\
&< \alpha + \frac{(k-j+1)}{k} + \frac{1}{k} && \text{(Since } r < \tfrac{1}{k}\text{)}\\
&< \alpha + \frac{(k-(j-1)+1)}{k}\\
&< \alpha + \frac{(k-(j-1)+1)}{k}
\end{aligned}$$

and

$$\begin{aligned}
r_i t - N_{i,t} + r_i &\ge \alpha + \frac{k-j}{k} + r_i\\
&\ge \alpha + \frac{k-j}{k} && \text{(Since } r_i \in [0, 1/k]\text{)}\\
\implies i \in M_{j-1,t+1} \uplus M_{j,t+1}
\end{aligned}$$

□

Induction base case ($t = 1$): Let $i_1$ be the arm pulled at $t = 1$. Then

$$r_i t - N_{i_1,1} = r_i - 1 < 0 <= \alpha$$

$$\implies i_1 \in S_1$$

For all $i \neq i_1$, we have $r_i t - N_{i,1} = r_i < \frac{1}{k} \le \alpha + \frac{1}{k} \implies i \in S_1 \uplus M_{k,1}$. Hence,

$$\begin{aligned}
V_{k,1} \uplus S_1 &= [k]\\
|V_{k,1}| &\le k - 1 < k\\
|V_{j,1}| &= 0 \qquad \text{for all } j \in [k-1]
\end{aligned}$$

Thus, conditions (1) and (2) of the lemma hold.

Inductive Step: Assuming the conditions in the lemma hold after round $t$, we show that they hold after round $t+1$.

*Case 1:* $i_{t+1} \in S_t$. From Observation 2, we know $i_{t+1} \in S_{t+1}$. From Observation 3, we know that for any arm $i \neq i_{t+1}$, $i \in S_{t+1} \uplus M_{k,t+1}$. Hence,

$$\begin{aligned}
V_{k,t+1} \uplus S_{t+1} &= [k]\\
|V_{j,t+1}| &= 0 \qquad \text{for all } j \in [k-1]\\
|V_{k,t+1}| &\le k - 1 < k
\end{aligned}$$

Thus, Conditions (1) and (2) in the lemma hold after round $t+1$.
*Case 2:* $i_{t+1} \in M_{a,t}$, for some $a \in [k]$.

$$\begin{aligned} & i_{t+1} \in M_{a,t} \\ \implies & i_{t+1} \in V_{a,t} \\ \implies & |V_{j,t}| = 0 \qquad \text{for all } j \in [1, a-1] \end{aligned}$$

From Observation 2, we know $i_{t+1} \in S_{t+1} \uplus M_{k,t+1}$. Hence,

$$\begin{aligned} & |V_{j,t}/\{i_{t+1}\}| \leq j-1 \qquad \text{for all } j \in [a,k] \\ \implies & |V_{j,t+1}| \leq j \qquad \text{for all } j \in [k] \end{aligned}$$

Also, $V_{k,t+1} \uplus S_{t+1} = [k]$. Hence, Conditions (1) and (2) of the lemma hold after round $t+1$. □

We proved above that, given an *unfairness tolerance* $\alpha \geq 0$, FAIR-ALG is *$\alpha$-fair*. In particular, note that the guarantee also holds when $\alpha = 0$ and hence FAIR-ALG with $\alpha = 0$ is *fair*. Next, we provide an upper bound on the $r$-Regret of FAIR-UCB.

**Theorem 4.** *For* Fair-SMAB *problem,* Fair-UCB *has r-Regret* $\mathbb{E}[\mathcal{R}^r_{\text{Fair-UCB}}(T)] = O(\sum_{i \neq 1} \frac{\ln T}{\Delta_i})$. *In particular, the r-Regret of* Fair-UCB *is given by*

$$\mathcal{R}^r_{\text{Fair-UCB}}(T) \leq \left(1 + \frac{\pi^2}{3}\right) \cdot \sum_{i \in [k]} \Delta_i + \sum_{\substack{i \in S(T) \\ i \neq 1}} \Delta_i \cdot \left(\frac{8 \ln T}{\Delta_i^2} - \left(r_i \cdot T - \alpha\right)\right)$$

*where* $S(T) = \left\{ i \in [k] \;\middle|\; r_i \cdot T - \alpha < \frac{8 \ln T}{\Delta_i^2} \right\}$.

*Proof.* Recall that $\bar{\mu}_i(t) = \hat{\mu}_{i,N_{i,t-1}}(t-1) + c_{t,N_{i,t-1}}$ is the UCB estimate of the mean of arm $i$, where $\hat{\mu}_{i,N_{i,t-1}}(t-1)$ is the empirical estimate of the mean of arm $i$ when it is played $N_{i,t-1}$ in $t-1$ rounds and $c_{t,N_{i,t-1}} = \sqrt{\frac{2 \ln t}{N_{i,t-1}}}$ is the confidence interval of the arm $i$ at round $t$. Similar to the analysis of the UCB1 algorithm (Appendix A, Theorem 8), we upper bound the expected number of times a sub-optimal arm is pulled. We do this for each sub-optimal arm by considering two cases dependent on the number of times the sub-optimal arm is required to be pulled for satisfying its fairness constraint i.e. on the value of the quantity $r_i T - \alpha$.
Case 1: Let $i \neq 1$ and $r_i \cdot T - \alpha \geq \frac{8 \ln T}{\Delta_i^2}$. Then

$$\begin{aligned} \mathbb{E}[N_{i,T}] &\leq \left(r_i \cdot T - \alpha\right) + \sum_{t=1}^{T} \mathbb{1}\{i_t = i, N_{i,t-1} \geq r_i \cdot T - \alpha\} \\ &\leq \left(r_i \cdot T - \alpha\right) + \sum_{t=1}^{\infty} \sum_{s_1=1}^{t} \sum_{s_i = r_i \cdot T - \alpha}^{t} \mathbb{1}\left\{\hat{\mu}_{1,s_1}(t) + c_{t,s_1} \leq \hat{\mu}_{1,s_i}(t) + c_{t,s_i}\right\} \end{aligned}$$

(Follows from Section A.2)

Since $r_i \cdot T - \alpha \geq \frac{8 \ln T}{\Delta_i^2}$, it follows from the proof of Theorem 8 that $\mathbb{E}[N_{i,T}] \leq r_i \cdot T - \alpha + \left(1 + \frac{\pi^2}{3}\right)$.
Hence, $\mathbb{E}[N_{i,T}] - \left(r_i \cdot T - \alpha\right) \leq \left(1 + \frac{\pi^2}{3}\right)$.

Case 2: Let $i \neq 1$ and $r_i \cdot T < \frac{8 \ln T}{\Delta_i^2}$

Then the proof of Theorem 8 can be appropriately adapted to show that $\mathbb{E}[N_{i,T}] \leq \frac{8 \ln T}{\Delta_i^2} + \left(1 + \frac{\pi^2}{3}\right)$. Hence

$$\mathbb{E}[N_{i,T}] - \left(r_i \cdot T - \alpha\right) \leq \frac{8 \ln T}{\Delta_i^2} + \left(1 + \frac{\pi^2}{3}\right) - \left(r_i \cdot T - \alpha\right)$$

Suppose $S(T) = \left\{ i \in [k] \;\middle|\; r_i \cdot T - \alpha < \frac{8 \ln T}{\Delta_i^2} \right\}$. Then from the two cases discussed above, we can conclude that

$$\mathcal{R}^r_{\text{FAIR-UCB}}(T) \leq \left(1 + \frac{\pi^2}{3}\right) \cdot \sum_{i \in [k]} \Delta_i + \sum_{\substack{i \in S(T) \\ i \neq 1}} \Delta_i \cdot \left(\frac{8 \ln T}{\Delta_i^2} - \left(r_i \cdot T - \alpha\right)\right)$$

Hence, $\mathcal{R}^r_{\text{FAIR-UCB}}(T) = O(\sum_{i \neq 1} \frac{\ln T}{\Delta_i})$. □

Next, we prove that the instance independent regret of FAIR-UCB is $O(\sqrt{T})$.

**Theorem 5.** *The instance-independent r-Regret of* Fair-UCB *is* $O(\sqrt{T})$.

*Proof.* Recall from Definition 4 our expression for the $r$-Regret of a FAIR-SMAB algorithm $\mathcal{A}$. We know,

$$\begin{aligned} \mathbb{E}[\mathcal{R}^r_{\mathcal{A}}(T)] &= \sum_{i \in [k]} \Delta_i \cdot \left(\mathbb{E}[N_{i,T}] - \left(\lfloor r_i \cdot T \rfloor - \alpha\right)\right) \\ &\leq k + \sum_{i \in [k]} \Delta_i \cdot \left(\mathbb{E}[N_{i,T}] - \left(r_i \cdot T - \alpha\right)\right) \end{aligned}$$

Note that, given any instance with $k$ arms, $\mu = (\mu_1, \mu_2, \ldots, \mu_k)$, and a constant $\alpha \geq 0$,

$$\begin{aligned} \mathbb{E}[\mathcal{R}^r_{\mathcal{A}}(T)] &\leq \max_{\substack{r_i \in [0,1]^k \\ \sum_{i \in [k]} r_i < 1}} k + \sum_{i \in [k]} \Delta_i \cdot \left(\mathbb{E}[N_{i,T}] - \left(r_i \cdot T - \alpha\right)\right) \\ &\leq k + \sum_{i \in [k]} \Delta_i \cdot \mathbb{E}[N_{i,T}] \end{aligned}$$

The last inequality follows from the fact that $r_i \geq 0$ for all $i \in k$, and $\alpha$ is a constant. This implies that the regret for any instance with given value of $r = (r_1, r_2, \ldots, r_k)$ is bounded by the regret of the same instance for $r_1 = r_2 = \ldots = r_k = 0$. But when, $r_1 = r_2 = \ldots = r_k = 0$, FAIR-UCB is the same as UCB1. Hence, from the instance independent regret bound of UCB1 (See Appendix 8), the result follows. Thus we can bound the instance independent regret of FAIR-UCB as $O(\sqrt{T})$. □

## 6 Cost of Fairness

Our regret guarantees until now have been in terms of the extended notion of regret i.e. $r$-Regret. In the previous section we showed that FAIR-UCB achieves $O(\ln T)$ $r$-Regret . We now evaluate the *cost of fairness* in terms of the conventional notion of regret i.e. how much do we lose in terms of regret in comparison to a SMAB algorithm without any fairness constraints. In particular, we show the trade-off between regret and fairness in terms of the *unfairness tolerance* .

**Theorem 6.** *For the* Fair-SMAB *problem where* Learn($\cdot$) = *UCB1, the regret of* Fair-ALG *is given by*

$$\mathcal{R}(T) = \sum_{i \in S(T)} (r_i \cdot T - \alpha) \cdot \Delta_i + \sum_{\substack{i \notin S(T) \\ i \neq 1}} \Big(\frac{8 \ln T}{\Delta_i}\Big) + \sum_{i \in [k]} \Big(1 + \frac{\pi^2}{3}\Big) \cdot \Delta_i$$

*where* $S(T) = \Big\{ i \mid \big(r_i \cdot T - \alpha\big) \geq \frac{8 \ln T}{\Delta_i^2} \Big\}$

*Proof.* From Section 3, Equation 2 we know that $\mathcal{R}_{\mathcal{A}}(T) = \sum_{i \in [k]} \Delta_i \cdot \mathbb{E}[N_{i,T}]$ and hence, we can bound the expected regret of an algorithm by bounding the expected number of pulls of a sub-optimal arm. In particular, we want to bound the quantity $\mathbb{E}[N_{i,T}]$ for every sub-optimal arm $i \neq 1$. We do this by considering two cases dependent on how many times the arm $i$ has been pulled to satisfy the fairness constraint, i.e. on how large is the quantity $r_i \cdot T - \alpha$.
<u>Case 1:</u> Let $i \neq 1$ and $r_i \cdot T - \alpha \geq \frac{8 \ln T}{\Delta_i^2}$. Then

$$\begin{aligned} \mathbb{E}[N_{i,T}] &\leq (r_i \cdot T - \alpha) + \sum_{t=1}^{T} \mathbb{1}\{i_t = i, N_{i,t-1} \geq r_i \cdot T - \alpha\} \\ &\leq (r_i \cdot T - \alpha) + \sum_{t=1}^{\infty} \sum_{s_1=1}^{t} \sum_{s_i = r_i \cdot T - \alpha}^{t} \mathbb{1}\Big\{\hat{\mu}_{1,s_1}(t) + c_{t,s_1} \leq \hat{\mu}_{1,s_i}(t) + c_{t,s_i}\Big\} \end{aligned}$$

(Follows from Section A.2)

Since $(r_i \cdot T - \alpha) \geq \frac{8 \ln T}{\Delta_i^2}$, it follows from the proof of Theorem 8 that $\mathbb{E}[N_{i,T}] \leq (r_i \cdot T - \alpha) + \Big(1 + \frac{\pi^2}{3}\Big)$.
<u>Case 2:</u> Let $i \neq 1$ and $r_i \cdot T - \alpha < \frac{8 \ln T}{\Delta_i^2}$
Then the proof of Theorem 8 can be appropriately adapted to show that $\mathbb{E}[N_{i,T}] \leq \frac{8 \ln T}{\Delta_i^2} + \Big(1 + \frac{\pi^2}{3}\Big)$.
Hence

$$r_i \cdot T - \alpha \leq \mathbb{E}[N_{i,T}] \leq \frac{8 \ln T}{\Delta_i^2} + \Big(1 + \frac{\pi^2}{3}\Big)$$

Then from the two cases discussed above, we can conclude that

$$\mathcal{R}(T) \leq \sum_{i \in S(T)} (r_i \cdot T - \alpha) \cdot \Delta_i + \sum_{\substack{i \notin S(T) \\ i \neq 1}} \Big(\frac{8 \ln T}{\Delta_i}\Big) + \sum_{i \in [k]} \big(1 + \frac{\pi^2}{3}\big) \cdot \Delta_i$$

where $S(T) = \Big\{ i \in [k] \mid r_i \cdot T - \alpha \geq \frac{8 \ln T}{\Delta_i^2} \Big\}$. □

Theorem 6 capture the explicit trade-off in regret in terms of $\alpha$ which characterizes the fairness constraints. Notice the trade-off between fairness guarantees achieved by the algorithm and the asymptotic regret guarantees. If $S(T) = \emptyset$ we have that the regret is $O(\ln T)$. This implies that for $\alpha > r_i T - \frac{8 \ln T}{\Delta_i^2}$ the regret is $O(\ln T)$. However, if $S(T) \neq \emptyset$ then for each $i \in S(T)$, an additional regret equal to $r_i T - \alpha$ is incurred. Note in this case that the regret can be of $O(T)$. We complement these results with simulations in Section 7.

# 7 Experimental Results

In this section we show the results of simulations that validate our theoretical guarantees. First, we represent the cost of fairness by showing the trade-off between regret and fairness with respect to the *unfairness tolerance* $\alpha$. Second, we evaluate the performance of our algorithms in terms of $r$-Regret and fairness guarantee by considering the algorithm by Li et al. [4], called Learning with Fairness Guarantee(LFG), as a baseline.

## 7.1 Trade-off: Fairness vs. Regret

We consider the following FAIR-SMAB instance: $k = 10$, $\mu_1 = 0.8$, and $\mu_i = \mu_1 - \Delta_i$, where $\Delta_i = 0.01i$, and $r = (0.05)^k$. We show the results for $T = 10^6$. Figure 3 shows the trade-off between regret in terms of the conventional notion and fairness. As can be seen, the cost of fairness can be linear in terms of regret up to a certain value of $\alpha$. This implies that until the threshold for $\alpha$ is reached where regret drop from linear to logarithmic, the fairness constraints cause some sub-optimal arms to pulled more number of times as compared to number of times an arm needs to be pulled to determine its mean reward with sufficient confidence. On the other hand, for values of $\alpha$ beyond this threshold, the regret reduces drastically, and we recover logarithmic regret as could be expected from the classic UCB1 algorithm. Note that threshold for $\alpha$ is in this case is problem-dependent.

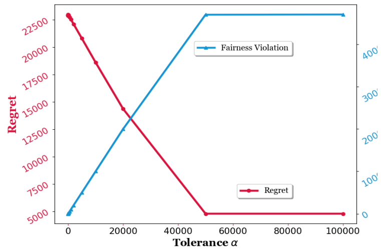

**Figure 3**
*Trade-off: Regret vs. Fairness*

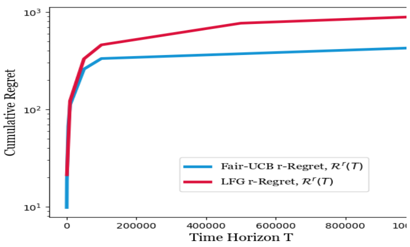

**Figure 4**
*r-Regret:* Fair-UCB *vs.* LFG

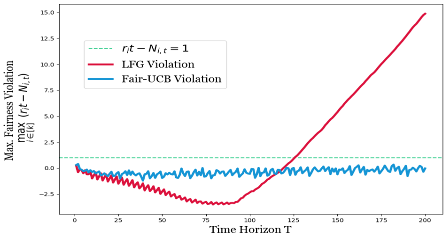

**Figure 5**
*r-Regret:* Fair-UCB *vs.* LFG

## 7.2 Comparison: FAIR-UCB vs. LFG

As we detailed in Section 2, the work closest to ours is that by Li et al. [4] and their algorithm, which is called *Learning with Fairness Guarantee* (LFG) is used as a baseline in the following simulation results. The simulation parameters that we consider for comparing $r$-Regret are the same as in the previous section. Figure 4 shows the plot of time vs. $r$-Regret for FAIR-UCB and LFG. Note that FAIR-UCB and LFG perform comparably in terms of the $r$-Regret suffered by the algorithm. Also, the simulation results validate our theoretical claim of logarithmic $r$-Regret bound.

We next contrast the fairness guarantee of FAIR-ALG with that of LFG. To show this comparison we consider an instance with $k = 3$, $\mu = (0.7, 0.5, 0.4)$, $r = (0.2, 0.3, 0.25)$, and $\alpha = 0$. Even though we tested the fairness guarantee for $T = 10^6$, we show the plot for $T = 200$ as it turns out to be the appropriate scale to compare the performance of FAIR-ALG and LFG without losing any details in terms of the fairness violation. Figure 5 shows the plot of time vs. $\max_{i \in [k]}(r_i t - N_{i,t})$ i.e. the maximum value of the quantity that captures possible fairness violation among all arms.

As can been seen in the figure, the fairness guarantee of FAIR-UCB holds uniformly over the time horizon $T$. Also note that, even though the fairness violation for LFG appears to be increasing, it does reduce at some point and go to zero which guarantees asymptotic fairness. To summarize, the simulation result reaffirm our theoretical guarantees for both fairness and $r$-Regret of FAIR-ALG in general, and FAIR-UCB in particular.

## 8 Discussion and Future Work

In this paper we discussed the problem of fairness in the SMAB setting. We proposed fairness guarantees that hold uniformly over time and showed that the logarithmic $r$-Regret can be achieved at the same time. One immediate criticism to this work can be that the regret definition is twisted in such a way so as to be advantageous to the proposed algorithms. It is to be noted that we consider situations where fairness is indispensable and must be satisfied even by the optimal algorithm. An immediate direction for future work could be to study the other variants of Multi-armed Bandits such as, adversarial bandits, Combinatorial bandits (with general reward structure), Contextual bandits, Markovian bandits, etc. Finally, an instance independent threshold for the *unfairness tolerance* $\alpha$ could be derived at which point the conventional regret drops from being linear in $T$ to a logarithmic regret.

## References

[1] John Rawls. *A theory of justice*. Harvard university press, 1971. (Not cited.)

[2] Samuel Freeman. Original position. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, summer 2019 edition, 2019. (Not cited.)

[3] Hoda Heidari, Claudio Ferrari, Krishna Gummadi, and Andreas Krause. Fairness behind a veil of ignorance: A welfare analysis for automated decision making. In *Advances in Neural Information Processing Systems 31*, pages 1265–1276. 2018. (Not cited.)

[4] Fengjiao Li, Jia Liu, and Bo Ji. Combinatorial sleeping bandits with fairness constraints. In *Accepted, IEEE INFOCOM*, 2019. (Not cited.)

[5] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification. *arXiv preprint arXiv:1803.02453*, 2018. (Not cited.)

[6] Harikrishna Narasimhan. Learning with complex loss functions and constraints. In *International Conference on Artificial Intelligence and Statistics*, pages 1646–1654, 2018. (Not cited.)

[7] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *26th International Conference on World Wide Web*, pages 1171–1180, 2017. (Not cited.)

[8] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*, pages 962–970, 2017. (Not cited.)

[9] Richard Berk, Hoda Heidari, Shahin Jabbari, Matthew Joseph, Michael Kearns, Jamie Morgenstern, Seth Neel, and Aaron Roth. A convex framework for fair regression. *arXiv preprint arXiv:1706.02409*, 2017. (Not cited.)

[10] Ashkan Rezaei, Rizal Fathony, Omid Memarrast, and Brian D. Ziebart. Fair logistic regression: An adversarial perspective. *CoRR*, abs/1903.03910, 2019. (Not cited.)

[11] Ashudeep Singh and Thorsten Joachims. Policy learning for fairness in ranking. *arXiv preprint arXiv:1902.04056*, 2019. (Not cited.)

[12] Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Li Wei, Yi Wu, Lukasz Heldt, Zhe Zhao, Lichan Hong, Ed H Chi, et al. Fairness in recommendation ranking through pairwise comparisons. *arXiv preprint arXiv:1903.00780*, 2019. (Not cited.)

[13] Ashudeep Singh and Thorsten Joachims. Fairness of exposure in rankings. In *24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2219–2228, 2018. (Not cited.)

[14] L Elisa Celis, Damian Straszak, and Nisheeth K Vishnoi. Ranking with fairness constraints. *arXiv preprint arXiv:1704.06840*, 2017. (Not cited.)

[15] Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. Fa*ir: A fair top-k ranking algorithm. In *ACM Conference on Information and Knowledge Management*, pages 1569–1578, 2017. (Not cited.)

[16] Stephen Gillen, Christopher Jung, Michael Kearns, and Aaron Roth. Online learning with an unknown fairness metric. In *Advances in Neural Information Processing Systems*, pages 2600–2609. 2018. (Not cited.)

[17] Yang Liu, Goran Radanovic, Christos Dimitrakakis, Debmalya Mandal, and David C Parkes. Calibrated fairness in bandits. *arXiv preprint arXiv:1707.01875*, 2017. (Not cited.)

[18] Matthew Joseph, Michael Kearns, Jamie H Morgenstern, and Aaron Roth. Fairness in learning: Classic and contextual bandits. In *Advances in Neural Information Processing Systems*, pages 325–333, 2016. (Not cited.)

[19] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Theoretical Computer Science Conference*, pages 214–226. ACM, 2012. (Not cited.)

[20] Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory*, pages 39–1, 2012. (Not cited.)

[21] Shweta Jain, Satyanath Bhat, Ganesh Ghalme, Divya Padmanabhan, and Y Narahari. Mechanisms with learning for stochastic multi-armed bandit problems. *Indian Journal of Pure and Applied Mathematics*, 47(2):229–272, 2016. (Not cited.)

[22] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002. (Not cited.)

# A Preliminaries

## A.1 Hoeffding's Lemma

**Theorem 7.** *Let $X_1, X_2, \ldots, X_n$ be i.i.d. random variables with $X_i \in [a, b]$ and $\mathbb{E}[X_i] = \mu$. Then*

$$Pr\bigg( \bigg|\frac{1}{n}\sum_{i=1}^{n} X_i - \mu_i\bigg| \geq \epsilon \bigg) \leq 2e^{\frac{-2n\epsilon^2}{(b-a)^2}}$$

## A.2 Upper Confidence Bound (UCB) based Algorithm

In this section we describe the UCBalgorithm that was introduced by [22] and for completeness we also give a proof of its regret bound. In the UCBalgorithm for each arm the algorithm maintains a UCBestimate and at each round the algorithm plays the arm with the highest UCBestimate. Such a UCBestimate for an arm $i \in [k]$ at round $t$ is dependent on the empirical mean of the rewards of arm $i$ and a confidence interval associated with arm $i$. To state it formally let $N_{i,t-1}$ denote the number of times arm $i$ is played in $t-1$ rounds. Then the UCBestimate for arm $i \in [k]$ at round $t \geq 1$ is $\bar{\mu}_i(t) = 0$ if $N_{i,t-1} = 0$, otherwise $\bar{\mu}_i(t) = \hat{\mu}_{i,N_{i,t-1}}(t-1) + \sqrt{\frac{2\ln(t)}{N_{i,t-1}}}$ where $\hat{\mu}_{i,N_{i,t-1}}(t-1)$ is the empirical mean of the rewards of arm $i$ after being played $N_{i,t-1}$ times in $t-1$ rounds and $\sqrt{\frac{2\ln(t)}{N_{i,t-1}}}$ is its associated confidence interval. For ease of notation, we will denote by $c_{t,s_i}$ the confidence interval of arm $i$ at time $t$ when it is played $s_i$ times i.e. $c_{t,s_i} = \sqrt{\frac{2\ln(t)}{s_i}}$ . Technically for the first $k$ rounds the algorithm plays each arm once to compute a non-zero UCBestimate for each arm and for every round $t \geq k+1$ it plays the arm with the highest UCBestimate. The total expected regret of UCBafter $T$ rounds is given by the following theorem, where $\Delta_i = \mu_1 - \mu_i$ for all $i \in [k]$, and $\Delta_i > 0$ as $\mu_1 > \mu_i$ for $i \neq 1$.

**Theorem 8.** *For the* SMAB *problem, the UCB has expected regret* $\mathbb{E}[\mathcal{R}_{\text{UCB}}(T)] \leq \sum_{i\neq 1}\big(\frac{8\ln T}{\Delta_i}\big) + \big(1 + \frac{\pi^2}{3}\big)\sum_{i\in[k]}\Delta_i$ .

*Proof.* To bound the regret of the UCB algorithm, we first upper bound $\mathbb{E}[N_{i,T}]$ for $i \neq 1$, i.e. the expected number of pulls of a sub-optimal arm $i \neq 1$ in $T$ rounds. Denote the arm pulled by the algorithm at the $t$-th round as $i_t$. In the equation below $\mathbb{1}\{i_t = i\}$ is an indicator random variable that is equal to 1 if $i_t = i$ and is 0 otherwise. In general $\mathbb{1}\{\mathsf{E}\}$ denotes an indicator random variable that is equal to 1 if the event $\mathsf{E}$ is true and is 0 otherwise.

$$N_{i,T} = 1 + \sum_{t=k+1}^{T} \mathbb{1}\Big\{i_t = i\Big\}$$

For any positive integer $\ell$ we may rewrite the above equation as

$$N_{i,T} \leq \ell + \sum_{t=\ell}^{T} \mathbb{1}\Big\{i_t = i, N_{i,t-1} \geq \ell\Big\} \tag{4}$$

If $i_t = i$ then $\bar{\mu}_1(t) < \bar{\mu}_i(t)$ i.e. $\hat{\mu}_{1,N_{1,t-1}}(t-1) + c_{t,N_{1,t-1}} < \hat{\mu}_{i,N_{i,t-1}}(t-1) + c_{t,N_{i,t-1}}$ . Hence from

Equation 4

$$
\begin{aligned}
N_{i,T} &\leq \ell + \sum_{t=\ell}^{T} \mathbb{1}\Big\{\hat{\mu}_{1,N_{1,t-1}}(t-1) + c_{t,N_{1,t-1}} < \hat{\mu}_{i,N_{i,t-1}}(t-1) + c_{t,N_{i,t-1}}\ ,\ N_{i,t-1} \geq \ell\Big\} \\
&\leq \ell + \sum_{t=\ell}^{T} \mathbb{1}\Big\{\min_{0<s_1<t}\ \hat{\mu}_{1,s_1}(t-1) + c_{t,s_1} < \max_{\ell \leq s_i < t}\ \hat{\mu}_{i,s_i}(t-1) + c_{t,s_i}\Big\} \\
&\leq \ell + \sum_{t=\ell}^{T} \sum_{s_1=1}^{t} \sum_{s_i=\ell}^{t} \mathbb{1}\Big\{\hat{\mu}_{1,s_1}(t-1) + c_{t,s_1} < \hat{\mu}_{i,s_i}(t-1) + c_{t,s_i}\Big\}
\end{aligned}
$$

At time $t$, $\hat{\mu}_{1,s_1}(t-1) + c_{t,s_1} < \hat{\mu}_{i,s_i}(t-1) + c_{t,s_i}$ implies that at least one of the following events is true

$$\big\{\hat{\mu}_{1,s_1}(t-1) \leq \mu_1 - c_{t,s_1}\big\} \tag{5}$$

$$\big\{\hat{\mu}_{i,s_i}(t-1) \geq \mu_i + c_{t,s_i}\big\} \tag{6}$$

$$\big\{\mu_1 < \mu_i + 2c_{t,s_i}\big\} \tag{7}$$

The probability of the events in Equations 5 and 6 can be bounded using Hoeffding's inequality as:

$$
\begin{aligned}
\mathbb{P}\Big(\big\{\hat{\mu}_{1,s_1}(t-1) \leq \mu_1 - c_{t,s_1}\big\}\Big) &\leq t^{-4} \\
\mathbb{P}\Big(\big\{\hat{\mu}_{i,s_i}(t-1) \geq \mu_i + c_{t,s_i}\big\}\Big) &\leq t^{-4}
\end{aligned}
$$

The event in equation 7 $\big\{\mu_1 < \mu_i + 2c_{t,s_i}\big\}$ can be written as $\Big\{\mu_1 - \mu_i - 2\sqrt{\frac{2\ln t}{s_i}} < 0\Big\}$. Substituting $\Delta_i = \mu_1 - \mu_i$ and if $s_i \geq \left\lceil \frac{8\ln T}{\Delta_i^2} \right\rceil \geq \left\lceil \frac{8\ln t}{\Delta_i^2} \right\rceil$ then

$$\mathbb{P}\left(\left\{\Delta_i - 2\sqrt{\frac{2\ln t}{s_i}} < 0\right\}\right) = 0 \tag{8}$$

Thus if $\ell = \frac{8\ln T}{\Delta_i^2}$ then

$$
\begin{aligned}
N_{i,T} &\leq \frac{8\ln T}{\Delta_i^2} + \sum_{t=\frac{8\ln T}{\Delta_i^2}}^{T} \sum_{s_1=1}^{t} \sum_{s_i=\frac{8\ln T}{\Delta_i^2}}^{t} \mathbb{1}\Big\{\hat{\mu}_{1,s_1}(t-1) + c_{t,s_1} < \hat{\mu}_{i,s_i}(t-1) + c_{t,s_i}\Big\} \\
\mathbb{E}[N_{i,T}] &\leq \frac{8\ln T}{\Delta_i^2} + \sum_{t=\frac{8\ln T}{\Delta_i^2}}^{T} \sum_{s_1=1}^{t} \sum_{s_i=\frac{8\ln T}{\Delta_i^2}}^{t} 2t^{-4} \quad \leq \quad \frac{8\ln T}{\Delta_i^2} \sum_{t=\frac{8\ln T}{\Delta_i^2}}^{\infty} \sum_{s_1=1}^{t} \sum_{s_i=\frac{8\ln T}{\Delta_i^2}}^{t} 2t^{-4} \\
&\leq \frac{8\ln T}{\Delta_i^2} + 1 + \frac{\pi^2}{3} \qquad \left(\text{as } \sum_{t=\frac{8\ln T}{\Delta_i^2}}^{\infty} \sum_{s_1=1}^{t} \sum_{s_i=\frac{8\ln t}{\Delta_i^2}}^{t} 2t^{-4} \quad \leq \quad 1 + \frac{\pi^2}{3}\right)
\end{aligned}
$$

Recall from Section 3, Equation 3, that

$$
\begin{aligned}
\mathbb{E}[\mathcal{R}_{\text{UCB}}(T)] &= \sum_{i \in [k]} \Delta_i \cdot \mathbb{E}[N_{i,T}] \\
&\leq \sum_{i \neq 1} \frac{8\ln T}{\Delta_i} + \left(1 + \frac{\pi^2}{3}\right) \cdot \sum_{i \in [k]} \Delta_i
\end{aligned}
$$

□

# B Omitted Proofs

## B.1 Regret bound for Naive

**Theorem 1.** *The regret of* Naive *algorithm for* Fair-SMAB *problem,* $\mathcal{R}^r_{\text{Naive}}(T) = O((T^2 \ln T)^{1/3})$

*Proof.* Let $[k]$ be the set of arms and $r = (r_1, r_2, \ldots, r_k)$ be the required fairness fraction vector. Since $\mathbb{E}[\mathcal{R}^r_{\mathcal{A}}(T)] = 0$ when $\sum_{i \in [k]} r_i = 1$, we assume without loss of generality that $\sum_{i \in [k]} r_i < 1$. Also, assume without loss of generality that $\mu_1 > \mu_2 > \ldots > \mu_k$ and let $\Delta_i = \mu_1 - \mu_i$. Let $T' = \varepsilon T$ denote the total number of rounds in the exploration phase (where $0 \leq \varepsilon \leq 1$), and $N_{i,T'}$ be the number of times arm $i$ is pulled until round $T'$. Then $N_{i,T'} = r'_i \cdot T'$ where $r'_i = \left(r_i + \frac{m}{k}\right)$, and $m = 1 - \sum_{i \in [k]} r_i$. Also let $c_i = \sqrt{\frac{\alpha \ln T}{2N_{i,T'}}}$, where $\alpha > 1$ is some constant.

The regret of Naivefrom $t = 1$ to $t = T'$ is given by

$$\mathcal{R}^r_{\text{Naive}}(\varepsilon T) = \frac{m}{k} \cdot \varepsilon T \cdot \sum_{i \neq 1} \Delta_i \tag{9}$$

and that from $t = T' + 1$ to $t = T$ by

$$\mathcal{R}^r_{\text{Naive}}((1-\varepsilon)T) = m \cdot (1-\varepsilon)T \cdot \Delta_j \tag{10}$$

where $j = \text{argmax}_{i \in [k]} \hat{\mu}_i(T')$[2]. From Hoeffding's inequality, for any $i \in [k]$:

$$\mathbb{P}(\mu_i > \hat{\mu}_i(T') + c_i) \leq T^{-\alpha} \tag{11}$$

$$\mathbb{P}(\mu_i < \hat{\mu}_i(T') - c_i) \leq T^{-\alpha} \tag{12}$$

$$\begin{aligned}
\mu_j &\geq \hat{\mu}_j(T') - c_j && \text{(with probability at least } 1 - T^{-\alpha}\text{, from Eq. 12)}\\
\mu_1 - \mu_j &\leq \mu_1 - \hat{\mu}_j(T') + c_j && \text{(Since } \mu_1 \geq \mu_j\text{)}\\
&\leq \hat{\mu}_1(T') + c_1 - \hat{\mu}_j(T') + c_j && \text{(with probability at least } 1 - 2T^{-\alpha}\text{, from Eq. 11)}\\
&\leq \hat{\mu}_j(T') + c_1 - \hat{\mu}_j(T') + c_j && \text{(Since } j = \text{argmax}_{i \in [k]} \hat{\mu}_i(T')\text{)}\\
\Delta_j &\leq c_1 + c_j && \text{(with probability at least } 1 - 2T^{-\alpha}\text{)}
\end{aligned}$$

Now,

$$\begin{aligned}
c_1 + c_j &= \sqrt{\frac{\alpha \ln T}{2N_{1,T'}}} + \sqrt{\frac{\alpha \ln T}{2N_{j,T'}}}\\
&\leq \sqrt{\frac{\alpha \ln T}{2r'_1 \cdot T'}} + \sqrt{\frac{\alpha \ln T}{2r'_j \cdot T'}}\\
&= \sqrt{\frac{\alpha \ln T}{2\varepsilon T}} \cdot \left[\frac{\sqrt{r'_1} + \sqrt{r'_j}}{\sqrt{r'_1 \cdot r'_j}}\right]
\end{aligned}$$

[2] Ties are broken lexicographically.

Hence, with probability at least $1 - 2T^{-\alpha}$

$$\Delta_j \leq \sqrt{\frac{\alpha \ln T}{2\varepsilon T}} \cdot \left[ \frac{\sqrt{r_1'} + \sqrt{r_j'}}{\sqrt{r_1' \cdot r_j'}} \right]$$

and thus, with probability at most $2T^{-\alpha}$, $\Delta_j \leq 1$, which is the trivial upper bound on $\Delta_j$ for any $j \in [k]$.

Let $f(r_j') = \left[ \frac{\sqrt{r_1'}+\sqrt{r_j'}}{\sqrt{r_1' \cdot r_j'}} \right]$. The expected regret of NAIVEafter $T$ rounds is given by,

$$\begin{aligned}
\mathbb{E}[\mathcal{R}^r_{\text{NAIVE}}(T)] &= \frac{m}{k} \cdot \varepsilon T \cdot \sum_{i \neq 1} \Delta_i + m(1-\varepsilon)T \cdot \Delta_j \\
&\leq \frac{m}{k} \cdot \varepsilon T \cdot \sum_{i \neq 1} \Delta_i + m \cdot T \cdot \Delta_j \\
&\leq \frac{m}{k} \cdot \varepsilon T \cdot \sum_{i \neq 1} \Delta_i + m \cdot T \cdot \sqrt{\frac{\alpha \ln T}{2\varepsilon T}} \cdot f(r_j') \cdot (1 - 2T^{-\alpha}) + m \cdot T \cdot (2T^{-\alpha}) \\
&\leq \frac{m}{k} \cdot \varepsilon T \cdot \sum_{i \neq 1} \Delta_i + mT \cdot \sqrt{\frac{\alpha \ln T}{2\varepsilon T}} \cdot f(r_j') + 2m \cdot T^{-(\alpha-1)}
\end{aligned}$$

We observe that if $\varepsilon = \left[ \frac{k}{2} \cdot f(r_j') \cdot \sqrt{\frac{\alpha \ln T}{2T}} \cdot \frac{1}{\sum_{i \neq 1} \Delta_i} \right]^{2/3}$ then $\mathbb{E}[\mathcal{R}^r_{\mathcal{A}}(T)] = O\big((T^2 \ln T)^{1/3}\big)$. □

## B.2 Regret bound for T-FUCB

**Theorem 2.** *For* Fair-SMAB *problem,* T-FUCB *has regret* $\mathcal{R}^r_{\text{T-FUCB}}(T) = O(\ln T)$. *In particular, its r-dependent regret is given by*

$$\mathcal{R}^r_{\text{T-FUCB}}(T) \leq \left(1 + \frac{\pi^2}{3}\right) \cdot \sum_{i \in [k]} \Delta_i + \sum_{\substack{i \in S(T) \\ i \neq 1}} \Delta_i \cdot \left( \frac{8 \ln T}{\Delta_i^2} - r_i \cdot T \right)$$

*where* $S(T) = \left\{ i \in [k] \;\middle|\; r_i \cdot T < \frac{8 \ln T}{\Delta_i^2} \right\}$.

*Proof.* Recall $\bar{\mu}_i(t) = \hat{\mu}_{i,N_{i,t-1}}(t-1) + c_{t,N_{i,t-1}}$ is the UCB estimate of the mean of arm $i$, where $\hat{\mu}_{i,N_{i,t-1}}(t-1)$ is the empirical estimate of the mean of arm $i$ when it is played $N_{i,t-1}$ in $t-1$ rounds and $c_{t,N_{i,t-1}} = \sqrt{\frac{2 \ln t}{N_{i,t-1}}}$ is the confidence interval of the arm $i$ at round $t$. Similar to the proof of Theorem 8 (UCB1 algorithm), we upper bound the expected number of times a sub-optimal arm is pulled. We do this for each sub-optimal arm by considering two cases dependent on the number of times the sub-optimal arm is pulled in the fairness constraint satisfaction phase, i.e. in the first $T'$ rounds.
Case 1: Let $i \neq 1$ and $r_i \cdot T \geq \frac{8 \ln T}{\Delta_i^2}$. Then

$$\begin{aligned}\mathbb{E}[N_{i,T}] &\le r_i \cdot T + \sum_{t=T'+1}^{T} \mathbb{1}\{i_t = i, N_{i,t-1} \ge r_i \cdot T\} \\ &\le r_i \cdot T + \sum_{t=T'}^{\infty} \sum_{s=1}^{t} \sum_{s_i=r_i \cdot T}^{t} \mathbb{1}\Big\{\hat{\mu}_{1,s}(t) + c_{t,s} \le \hat{\mu}_{1,s_i}(t) + c_{t,s_i}\Big\} \qquad \text{(Follows from Section A.2)}\end{aligned}$$

Since $r_i \cdot T \ge \frac{8 \ln T}{\Delta_i^2}$, it follows from the proof of Theorem 8 that $\mathbb{E}[N_{i,T}] \le r_i \cdot T + \Big(1 + \frac{\pi^2}{3}\Big)$. Hence, the expected number of pulls of a sub-optimal arm $i \ne 1$ in the regret minimization phase is $\mathbb{E}[N_{i,T}] - r_i \cdot T \le \Big(1 + \frac{\pi^2}{3}\Big)$.

Case 2: Let $i \ne 1$ and $r_i \cdot T < \frac{8 \ln T}{\Delta_i^2}$

Then the proof of Theorem 8 can be appropriately adapted to show that $\mathbb{E}[N_{i,T}] \le \frac{8 \ln T}{\Delta_i^2} + \Big(1 + \frac{\pi^2}{3}\Big)$. Thus the expected number of pulls of a sub-optimal arm $i \ne 1$ in the regret minimization phase is

$$\mathbb{E}[N_{i,T}] - r_i \cdot T \le \frac{8 \ln T}{\Delta_i^2} + \Big(1 + \frac{\pi^2}{3}\Big) - r_i \cdot T \le \frac{8 \ln T}{\Delta_i^2} + \Big(1 + \frac{\pi^2}{3}\Big)$$

Suppose $S(T) = \{i \in [k] | r_i \cdot T < \frac{8 \ln T}{\Delta_i^2}\}$. Then from the two cases discussed above, we can conclude that

$$\mathcal{R}^r_{\text{T-FUCB}}(T) \le \big(1 + \frac{\pi^2}{3}\big) \cdot \sum_{i \in [k]} \Delta_i + \sum_{\substack{i \in S(T) \\ i \ne 1}} \Delta_i \cdot \bigg(\frac{8 \ln T}{\Delta_i^2} - r_i \cdot T\bigg)$$

Hence, $\mathcal{R}^r_{\text{T-FUCB}}(T) = O(\ln T)$.

□

### B.3 Instance-independent Regret Bound of UCB1 Algorithm

**Theorem 9.** *For the* SMAB *problem, the UCBhas expected (instance-dependent) regret* $\mathbb{E}[\mathcal{R}_{\text{UCB}}(T)] = O\big(\sqrt{T \ln T}\big)$.

*Proof.* Recall from Section A.2 that the expected cumulative regret of the UCB1 algorithm in any round $T$ is given by

$$\mathbb{E}\big[\mathcal{R}_{\text{UCB}}(T)\big] = \sum_{i\ in [k]} \Delta_i \cdot \mathbb{E}[N_{i,T}].$$

To bound the above quantity, we begin by defining the event

$$C := \Bigg\{\big|\hat{\mu}_i(t) - \mu_i\big| \le \sqrt{\frac{2 \ln T}{N_{i,t}}}, \forall i \in [k], \forall t \le T\Bigg\}.$$

By applying Hoeffding's inequality, and taking union bound, we get

$$\mathbb{P}(\bar{C}) \le \frac{2kT}{T^4} \le \frac{2}{T^2}.$$

Next, we will bound the value of $\mathbb{E}\big[\mathcal{R}_{\text{UCB}}(T)\big]$ by conditioning on $C$ and $\bar{C}$. Let us first bound $\mathbb{E}\big[\mathcal{R}_{\text{UCB}}(T)\big|C\big]$. Assume the event $C$ holds and some arm $i_t \neq 1$ is played in round $t \in [T]$. Then, by definition of UCB1 algorithm, we have $\bar{\mu}_1(t) < \bar{\mu}_i(t)$. Then,

$$\begin{aligned}\mu_1 - \mu_{i_t} &\leq \mu_1 - \mu_{i_t} + \bar{\mu}_i(t) - \bar{\mu}_1(t)\\ &= \big(\mu_1 - \bar{\mu}_1(t)\big) + \big(\bar{\mu}_i(t) - \mu_{i_t}\big)\end{aligned}$$

Since event $C$ holds, we have

$$\mu_1 - \bar{\mu}_1(t) = \mu_1 - \hat{\mu}_1(t-1) - \sqrt{\frac{2\ln T}{N_{i,t-1}}} \leq 0.$$

and

$$\bar{\mu}_i(t) - \mu_{i_t} = \hat{\mu}_{i_t}(t-1) - \mu_{i_t} + \sqrt{\frac{2\ln T}{N_{i_t,t-1}}} \leq 2 \cdot \sqrt{\frac{2\ln T}{N_{i_t,t-1}}}.$$

Therefore,

$$\mu_1 - \mu_{i_t} \leq 2 \cdot \sqrt{\frac{2\ln T}{N_{i_t,t-1}}} \tag{13}$$

Now, consider any arm $i \in [k]$ and consider the last round $t_i \leq t$ when this arm was last played. Since the arm has not been played between $t_i$ and $t$, we know $N_{i,t_i} = N_{i,t-1}$. Hence, applying the inequality in Equation 13 to arm $i$ in round $t_i$, we get

$$\mu_1 - \mu_i \leq 2 \cdot \sqrt{\frac{2\ln T}{N_{i,t-1}}}, \text{ for all} t \leq T$$

. Thus, the regret in $t$ rounds is bounded by

$$\mathcal{R}(t) = \sum_{i\in[k]} \Delta_i \cdot N_{i,t} \leq 2\sqrt{2\ln T} \cdot \sum_{i\in[k]} \sqrt{N_{i,t}}.$$

Square root is a concave function, and hence from Jensen's inequality, we obtain

$$\sum_{i\in[k]} \sqrt{N_{i,t}} \leq \sqrt{kt}.$$

Therefor, we have

$$\mathbb{E}\big[\mathcal{R}_{\text{UCB}}(T)\big|C\big] \leq 2\sqrt{2kt\ln T}.$$

Hence, the expected cumulative regret in $t$ rounds can be bounded as

$$\begin{aligned}\mathbb{E}\big[\mathcal{R}_{\text{UCB}}(T) &= \mathbb{E}\big[\mathcal{R}_{\text{UCB}}(T)\big|C\big]\mathbb{P}(C) + \mathbb{E}\big[\mathcal{R}_{\text{UCB}}(T)\big|\bar{C}\big]\mathbb{C}\\ &\leq 2\sqrt{2kt\ln T} + t \cdot \frac{2}{T^2}\\ &= O\big(\sqrt{kt\ln T}\big), \quad \forall t \leq T\end{aligned}$$

Thus, the instance-independent regret bound of UCB1 algorithm at some time $T$ is $O(\sqrt{T\ln T})$. □