Journal of Machine Learning Research 22 (2021) 1-31 Submitted 7/20; Revised 3/21; Published 6/21

Achieving Fairness in the Stochastic Multi-Armed Bandit Problem

Vishakha Patil PATILV@IISC.AC.IN
Department of Computer Science and Automation
Indian Institute of Science
Bangalore, Karnataka, India

Ganesh Ghalme GANESHG@CAMPUS.TECHNION.AC.IL
Faculty of Industrial Engineering and Management
Technion Israel Institute of Technology
Haifa, Israel

Vineet Nair VINEET@CS.TECHNION.AC.IL
Faculty of Computer Science
Technion Israel Institute of Technology
Haifa, Israel

Y. Narahari NARAHARI@IISC.AC.IN
Department of Computer Science and Automation
Indian Institute of Science
Bangalore, Karnataka, India

Editor: Moritz Hardt

Abstract

We study an interesting variant of the stochastic multi-armed bandit problem, which we call the Fair-MAB problem, where, in addition to the objective of maximizing the sum of expected rewards, the algorithm also needs to ensure that at any time, each arm is pulled at least a pre-specified fraction of times. 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*, which we call r-Regret, that takes into account the above fairness constraints and extends the conventional notion of regret in a natural way. Our primary contribution is to obtain a complete characterization of a class of Fair-MAB algorithms via two parameters: the unfairness tolerance and the learning algorithm used as a black-box. For this class of algorithms, we provide a fairness guarantee that holds uniformly over time, irrespective of the chosen learning algorithm. Further, when the learning algorithm is UCB1, we show that our algorithm achieves constant r-Regret for a large enough time horizon. Finally, we analyze the *cost of fairness* in terms of the conventional notion of regret. We conclude by experimentally validating our theoretical results.

Keywords: Multi-armed Bandits, Fairness, Online Learning, Reinforcement learning, Machine Learning, Upper confidence Bounds.

1. Introduction

The multi-armed bandit (MAB) problem is a classic framework for sequential decision-making in uncertain environments. Starting with the seminal work of Robbins (1952), over the years, a significant body of work has been developed to address both theoretical aspects and practical applications of this problem; see Bubeck and Cesa-Bianchi (2012); Lattimore and Szepesvári (2020); Slivkins et al. (2019) for textbook expositions of the MAB problem. Indeed, the study of the MAB problem and its numerous variants continues to be a central pursuit in multiple fields such as online learning and reinforcement learning. In the MAB set-up, at every round, a decision-maker (an online algorithm) is faced with k choices, which correspond to unknown (to the algorithm) reward distributions. Each choice is referred to as an arm. When the decision-maker pulls a specific arm, she receives a reward drawn from the corresponding (a priori unknown) distribution[1]. The goal of the decision-maker is to maximize the cumulative reward in expectation accrued through a sequence of arm pulls, i.e., if the process repeats for T rounds, then in each round, the decision-maker selects an arm with the objective of maximizing the total expected reward[2].

Several variations of the MAB problem have been extensively studied in the literature. Various papers study MAB problems with additional constraints which include bandits with knapsack constraints (Badanidiyuru et al. (2018)), bandits with budget constraints (Xia et al. (2015)), sleeping bandits (Kleinberg et al. (2010); Chatterjee et al. (2017)), etc. In this paper, we consider Fair-MAB, a variant of the MAB problem where, in addition to maximizing the cumulative expected reward, the algorithm also needs to ensure that uniformly (i.e., at the end of every round) each arm is pulled at least a pre-specified fraction of times. This imposes an additional constraint on the algorithm we refer to as a *fairness constraint*, specified in terms of a vector $r \in \mathbb{R}^k$.

Formally, each component r_i of the given vector r specifies a *fairness-quota* for arm i and the online algorithm must ensure that for all time steps t (i.e. uniformly), each arm i is pulled at least $\lfloor r_i \cdot t \rfloor$ times in t rounds. The online algorithm's goal is to minimize expected regret while satisfying the fairness requirement of each arm. The expected regret in this setting, which we call r-Regret, is computed with respect to the optimal *fair* policy (see Definition 4). We note that the difficulty of this problem is in satisfying these fairness constraints at the end of every round, which in particular ensures fairness even when the time horizon is unknown to the algorithm beforehand.

It is relevant to note that the current work contributes to the long line of work in constrained variants of the MAB problem (Badanidiyuru et al. (2018); Kleinberg et al. (2010); Xia et al. (2015)). The fairness constraints described above naturally capture many real-world settings wherein the arm pulls correspond to the allocation of resources among agents with specified entitlements (quotas). The objective of ensuring an absolute minimum allocation guarantee to each individual is, at times, at odds with the objective of maximizing efficiency, the classical goal of any learning algorithm. However, in many applications, the allocation rules must consider such constraints to ensure fairness. The minimum entitlement over available resources secures the prerogative of individuals. For concreteness, we next present a motivating example.

1. The arms that are not pulled do not give any reward.
2. We study the standard set-up in which T is not known upfront to the online algorithm.

The US Department of Housing and Urban Development recently sued Facebook for engaging in housing discrimination by targeting ads based on attributes such as gender, race, religion, etc. which are protected classes under the US law[3]. Facebook's algorithm that decides which ad should be shown to a particular user inadvertently ends up discriminating because of the objective that it is trying to optimize. For example, if the algorithm learns that it can generate more revenue by displaying an ad to more men than women, it would end up discriminating against women. The proposed Fair-MAB model ensures that both men and women are shown the ad for at least a pre-specified fraction of the total number of ad displays, thereby preserving the fundamental right of equal access to opportunities. In a way, the minimum fraction guarantee also provides a moral justification to the chosen allocation rule by evaluating it to be fair under the veil of ignorance doctrine of Rawls (1971) in which an allocation rule is considered as a hypothetical agreement among free and equal individuals unaware of the natural capabilities and circumstantial advantages and biases they might have, i.e., a socially agreed-upon allocation in the original position (see Freeman (2019); Heidari et al. (2018) for a detailed discussion).

The fairness model in this work naturally captures many resource allocation situations such as the sponsored ads on a search engine where each advertiser should be guaranteed a certain fraction of pulls in a bid to avoid monopolization of ad space; crowd-sourcing where each crowd-worker is guaranteed a fraction of tasks in order to induce participation; and a wireless communication setting where the receiver must ensure a minimum quality of service to each sender. The work by Li et al. (2019) contains a detailed discussion of these applications. We discuss other related results on fairness in Section 2.

Our contributions: We first define the Fair-MAB problem in Section 3. Any Fair-MAB algorithm is evaluated based on two criteria: the fairness guarantees it provides and its r-Regret. The fairness notion that we consider requires that the fairness constraints be satisfied after each round, and the r-Regret notion is a natural extension of the conventional notion of regret, which is defined with respect to an optimal policy that satisfies the fairness constraints. The uniform time fairness guarantee that we seek ensures fairness even in *horizon-agnostic* case, i.e., when the time horizon T is unknown to the algorithm. We remark that, even when the horizon T is known, the intuitive approach of pulling each arm sufficiently many times to satisfy its fairness constraint does not guarantee fairness at the end of each round (see Section 9, Algorithm 2).

As our primary contribution, in Section 4, we define a class of Fair-MAB algorithms, called Fair-Learn, characterized by two parameters: the unfairness tolerance and the learning algorithm used as a black-box. We prove that any algorithm in Fair-Learn satisfies the fairness constraints at any time step t. Thus the fairness guarantee for Fair-Learn holds uniformly over time, independently of the learning algorithm chosen. We note here that our meta-algorithm Fair-Learn, allows any MAB algorithm to be plugged-in as a black-box. One can implement this simple yet elegant framework on top of any existing MAB algorithm to ensure fairness with quantifiable loss in terms of regret. The practical applicability of our algorithm is a notable feature of this work.

When the learning algorithm is UCB1, we prove a sub-logarithmic r-Regret bound for the Fair-ucb algorithm. Additionally, for sufficiently large T, we see that the Fair-ucb incurs

3. https://www.technologyreview.com/s/613274/facebook-algorithm-discriminates-ai-bias/

constant r-Regret. We then evaluate the cost of fairness in Fair-MAB with respect to the *conventional* notion of regret in Section 5. We conclude by providing detailed experimental results to validate our theoretical guarantees in Section 7. In particular, we compare the performance of Fair-ucb with LFG algorithm proposed in Li et al. (2019), which is the work closest to our paper. We remark here that we obtain a much stronger fairness guarantee that holds at any time, unlike the asymptotic fairness guarantee of LFG. We also prove a better regret bound with finer dependence on the problem instance parameters. Section 2 provides a detailed comparison.

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 (Agarwal et al., 2018; Narasimhan, 2018; Zafar et al., 2017a,b), regression algorithms (Berk et al., 2017; Rezaei et al., 2019), ranking and recommendation systems (Singh and Joachims, 2019; Beutel et al., 2019; Singh and Joachims, 2018; Celis et al., 2018; Zehlike et al., 2017), etc. This is true even in online learning, particularly in the MAB setting, and we discuss some relevant works below.

Joseph et al. (2016) propose a variant of the UCB algorithm that ensures what they 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 (Dwork et al., 2012) for each arm while achieving efficiency in terms of sub-linear regret. The work by Liu et al. (2017) aims at ensuring "treatment equality", wherein similar individuals are treated similarly, whereas Gillen et al. (2018) consider individual fairness guarantees with respect to an unknown fairness metric.

The papers discussed above combine the conventional goal of maximizing cumulative reward with that of simultaneously satisfying some additional constraints. Variants of the MAB problem with added constraints have been widely studied in the literature. For example, Badanidiyuru et al. (2018) and Immorlica et al. (2019) study the MAB problem with knapsack constraints, where some arm-specific budget limits the number of times that an algorithm can pull a particular arm. The works of Xia et al. (2015); Amin et al. (2012); Tran-Thanh et al. (2014) consider the MAB problem in which there is some cost associated with pulling each arm, and the learner has a fixed budget. The work by Lattimore et al. (2014, 2015); Talebi and Proutiere (2018) investigates bandit optimization problems with resource allocation constraints. We next discuss works that study the MAB problem with fairness constraints similar to those considered in our work.

Recent work by Li et al. (2019) studies the combinatorial sleeping multi-armed bandit problem with similar fairness constraints as those considered in our work. In addition to proving a $O(\sqrt{T \ln T})$ distribution-free r-Regret bound as in Li et al. (2019), we show a $O(\ln T)$ r-Regret bound with finer dependence on the instance parameters. Our fairness guarantee holds uniformly over time and hence is much stronger than the asymptotic fairness guarantee in Li et al. (2019). Moreover, as our fairness guarantee is independent of the learning algorithm used in Fair-Learn, it holds for the setting considered in Li et al. (2019).

Recent work by Celis et al. (2019) considers a personalized news feed setting, where at any time step t, for a given context (user), the arm (i.e., ad to be displayed) is sampled from a distribution p_t over the set $[k]$ of arms (ads), and fairness is achieved by ensuring a pre-specified probability mass on each arm which restricts the allowable set of distributions to a subset of the simplex. The algorithm by Celis et al. (2019), when applied the classical stochastic multi-armed bandit setting considered in our work, ensures any-time fairness only in *expectation* over the random pulls of arms by the algorithm. In contrast, our algorithm (Theorem 5) provides a much stronger deterministic any-time fairness guarantee. Further, we also provide an explicit trade-off (in terms of the unfairness tolerance α) between fairness and regret. Also, the computational overhead of our algorithm is just $O(1)$, whereas the algorithms by Celis et al. (2019) need to solve at least one linear program in each round. We also note that our model can directly be adapted to capture the setting in (Celis et al., 2019).

3. The Model

In this section, we formally define the Fair-MAB problem, the notion of fairness, and the concept of r-regret used in this work.

3.1 The Fair-MAB Problem

An instance of the Fair-MAB 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. In the Fair-MAB setting, the fairness constraints are endogenously specified to the algorithm in the form of a vector $r = (r_1, r_2, \ldots, r_k)$ where $r_i \in [0, \frac{1}{k-1})$, for all $i \in [k]$, and consequently $\sum_{i\in[k]} r_i < 1$ and r_i denotes the minimum fraction of times an arm $i \in [k]$ has to be pulled in T rounds, for any T. Though our results hold for $r_i \in [0, \frac{1}{k-1})$, we are primarily interested in the case where $r_i \in [0, 1/k)$ to be consistent with the notion of *proportionality* wherein, guaranteeing any arm a fraction greater than its proportional fraction, which is $1/k$, is *unfair* in itself. However, we show that our proposed framework satisfies the same fairness guarantee even with $r_i \in [0, \frac{1}{k-1})$ for all $i \in [k]$. We remark here that the problem of achieving fairness guarantee with $r_i \geq 1/(k-1)$ for some i remains open (see Section 8 for discussion).

In each round t, a Fair-MAB 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 $i \in [k]$. The results in this work can be easily extended to a MAB problem with general distributions supported on [0,1] via reduction to a MAB problem with Bernoulli rewards using the extension provided in Agrawal and Goyal (2012). 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. Next, we formalize the notions of fairness and regret used in the paper.

3.2 Notions of Fairness

Let $N_{i,t}$ denote the number of times arm i is pulled in t rounds. We first present the definition of fairness proposed by Li et al. (2019) and then define the stronger notion of fairness considered in this paper.

Definition 1 (Li et al. (2019)) *A* Fair-MAB *algorithm $\mathcal{A}$ is called (asymptotically) fair if for all $i \in [k]$ we have*

$$\liminf_{t\to\infty} \mathbb{E}_{\mathcal{A}}\Big[r_i - \frac{N_{i,t}}{t}\Big] \leq 0.$$

We refer to the above notion of fairness as *asymptotic fairness*. Note that this fairness guarantee is weak as it holds asymptotically and only in expectation. In other words, this fairness notion tolerates *prohibitively high value of unfairness* in the system for any reasonably large values of time-horizons. We now define a much stronger notion of fairness that holds over all rounds and is parameterized by the *unfairness tolerance* allowed in the system which is denoted by a constant $\alpha \geq 0$.

Definition 2 *Given an* unfairness tolerance $\alpha \geq 0$*, a* Fair-MAB *algorithm $\mathcal{A}$ is said to be α-fair if $\lfloor r_i t \rfloor - N_{i,t} \leq \alpha$ for all $t \leq T$, for all arms $i \in [k]$, and for any time horizon $T \geq 1$.*

In particular, if the above guarantee holds for $\alpha = 0$, then we call the Fair-MAB algorithm *fair*. Note that *α-fairness* guarantee holds uniformly over the time horizon and for any sequence of arm pulls $(i_t)_{t\leq T}$ by the algorithm. Hence, it is much stronger than the *asymptotic fairness* guarantee, which only guarantees fairness asymptotically (Definition 1). We also remark that α does not depend on the time horizon or the fairness vector and, for any $\alpha \geq 0$, α-fairness (Definition 2) implies asymptotic fairness (Definition 1).

3.3 Notions of Regret

In the MAB setting, the optimal policy in hindsight is the one that pulls the optimal arm in every round. The regret of a MAB algorithm is defined as the difference between the cumulative reward of this optimal policy and the algorithm.

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

$$\mathcal{R}_{\mathcal{A}}(T) = \sum_{i\in[k]} \Delta_i \cdot \mathbb{E}[N_{i,T}] \tag{1}$$

Here, $\Delta_i = \mu_1 - \mu_i$ and $N_{i,T}$ denotes the number of pulls of an arm $i \in [k]$ by $\mathcal{A}$ in T rounds.

The above notion of regret does not adequately quantify the performance of a Fair-MAB algorithm as the optimal policy here does not account for the fairness constraints. To see this, consider a two-armed bandits instance where arm 1 is the best arm and $r_1 = r_2 = 1/3$. Further, let $\mathcal{A}$ be an algorithm that pulls arm 2 in rounds that are multiples of 3 and pulls arm 1 in the other rounds. Then, it is easy to see that $\mathcal{A}$ is fair. Further, note that no other fair algorithm can have a higher expected reward than $\mathcal{A}$, since $\mathcal{A}$ pulls the sub-optimal arm (arm 2) exactly the number of times required to satisfy its fairness constraint. But the expected cumulative regret of $\mathcal{A}$ according to Definition 3 is in fact $O(T)$. This motivates

the need to define a suitable notion of regret for the Fair-MAB problem, which we call r-Regret , that takes into account the fairness constraints. We begin with the fairness-aware optimal policy that we consider as a baseline.

Observation 1 *A* Fair-MAB *algorithm $\mathcal{A}$ is optimal if and only if $\mathcal{A}$ satisfies the following: if $\lfloor r_i T \rfloor - \alpha > 0$ then $N_{i,T} = \lfloor r_i T \rfloor - \alpha$, else $N_{i,T} = 0$, for all $i \neq 1$.*

From Observation 1, it follows that an optimal Fair-MAB algorithm ensures that a sub-optimal arm is only pulled to satisfy its fairness constraint. Hence, an optimal Fair-MAB algorithm that knows the value of μ must play sub-optimal arms 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, that is, 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-MAB setting.

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

$$\mathcal{R}^r_{\mathcal{A}}(T) = \sum_{i \in [k]} \Delta_i \cdot \Big(\mathbb{E}[N_{i,T}] - \max\big(0, \lfloor r_i \cdot T \rfloor - \alpha\big) \Big) \tag{2}$$

The $\max(0, \lfloor r_i \cdot T \rfloor - \alpha)$ in the above definition accounts for the number of pulls of arm i made by the optimal algorithm to satisfy its fairness constraint. Also, the r-Regret of an algorithm that is not α-fair could be negative but this is an infeasible solution. 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-MAB problem are the conflicting constraints on the quantity $N_{i,T} - \lfloor r_i T \rfloor$ for a sub-optimal arm $i \neq 1$: at any time T, for the algorithm to be α-fair, we want $N_{i,T}$ to be at least $\lfloor r_i T \rfloor - \alpha$ whereas to minimize the r-Regret we require $N_{i,T}$ to be as small as possible.

4. A Framework for Fair-MAB Algorithms

In this section, we provide the framework of our proposed class of Fair-MAB algorithms. Our meta-algorithm Fair-Learn, which is given in Algorithm 1, defines a class of Fair-MAB algorithms characterized by two parameters: α, the unfairness tolerance allowed in the system, and Learn$(\cdot)$, the MAB algorithm used as a black-box. The simplicity of this framework allows for *any* standard MAB algorithm to be adapted into a Fair-MAB algorithm.

The key result in this work is Theorem 5, which guarantees that Fair-Learn is α-fair (see Definition 2) independent of the choice of the learning algorithm Learn$(\cdot)$. The two key properties that contribute to the technical difficulty of this result are: 1) the fairness guarantee holds uniformly over the time horizon even when the time horizon is not known beforehand, and 2) it holds for any sequence of arm pulls by the algorithm and not just in expectation over the arm pulls by the MAB algorithm.

Theorem 5 *For a given $\alpha \geq 0$ and for any given fairness constraint vector $r = (r_1, r_2, \ldots, r_k)$ where $r_i \in [0, \frac{1}{k-1})$ for all $i \in [k]$ and $\sum_{i \in [k]} r_i < 1$,* Fair-Learn *is α-fair irrespective of the choice of the learning algorithm* Learn$(\cdot)$.

Algorithm 1: Fair-Learn

Input: $[k], (r_i)_{i\in[k]}, \alpha \geq 0$, Learn$(\cdot)$

Initialize:

$N_{i,0} = 0$ for all $i \in [k]$

$S_{i,0} = 0$ for all $i \in [k]$, where $S_{i,t}$ = total reward of arm i in t rounds

for $t = 1, 2, \ldots$ **do**

  Define : $A(t) = \left\{ i \mid r_i \cdot (t-1) - N_{i,t-1} > \alpha \right\}$

  Pull arm $i_t = \begin{cases} \arg\max_{i\in[k]} \left(r_i \cdot (t-1) - N_{i,t-1} \right) & \text{If} A(t) \neq \emptyset \\ \text{Learn}(N_t, S_t) & \text{Otherwise} \end{cases}$

  Update parameters N_t and S_t

end

The proof of Theorem 5 is given in Section 6. The guarantee in the above theorem also holds when $\alpha = 0$ and hence Fair-Learn with $\alpha = 0$ is *fair*. In particular, when the learning algorithm Learn$(\cdot)$ = UCB1, we call this algorithm Fair-ucb. We provide the r-Regret bound for Fair-ucb.

Theorem 6 *The r-Regret of* Fair-ucb *is given by*

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

where $S(T, \alpha) = \left\{ i \in [k] \mid \alpha > r_i \cdot T - \frac{8 \ln T}{\Delta_i^2} \right\}$. *In particular, for large enough T, we get*

$$\mathcal{R}^r_{\text{Fair-ucb}}(T) \leq \left(1 + \frac{\pi^2}{3}\right) \cdot \sum_{i\in[k]} \Delta_i$$

Note that the upper bound presented in Theorem 6 is a constant r-Regret guarantee. Further, observe that if $r_i > 0$ for all suboptimal arms i,[4] any algorithm that uses a *consistent estimator*[5] of mean rewards to select the arm to be pulled, achieves a constant *asymptotic* r-Regret guarantee. To see this, note that after large enough number of time steps t, the estimate will be close to true mean by virtue of the fairness guarantee to be satisfied. Hence, after time step t, the algorithm will pull suboptimal arms only to satisfy the fairness constraints without incurring any additional r-Regret.

The choice of UCB1 in our framework is motivated by its *any-time* optimality guarantee. The proof of Theorem 6 is presented in Section 6. Observe that the proposed framework is very easy to implement on top of the in-place learning algorithms and hence can easily be made operational in practice. We now define the notion of distribution-free regret and then show that the distribution-free regret bound of Fair-ucb matches that of UCB1.

4. Note here that the algorithm does not know the true qualities of the arms beforehand.
5. A consistent estimator is defined as the sequence of estimates that converges in probability to the true value. Empirical mean, upper confidence bound, Thompson sampling estimate, lower confidence bound are few examples of consistent estimators of true means of reward distributions.

Definition 7 (Slivkins et al. (2019)) *Let the regret bound of* Alg *be denoted by* $\mathcal{R}_{\text{Alg}}(T) = C \cdot f(T)$*, where* $f(.)$ *does not depend on the reward distribution parameters* $(\mu_i)_i$*, and the 'constant'* C *does not depend on* T*. Such a regret bound is called distribution-free (or instance-independent) if* C *does not depend on* $(\mu_i)_i$*.*

We conclude this section by stating the distribution-free r-regret of Fair-ucb in Theorem 8 (proof in Section 6).

Theorem 8 *The distribution-free* r*-Regret of* Fair-ucb *is* $O(\sqrt{T \ln T})$*.*

This shows that the worst-case regret of Fair-ucb, which is independent of the underlying problem instance, is sub-linear in T and in fact matches the distribution-free regret bound of UCB1 itself.

5. Cost of Fairness

Our regret guarantees so far have been in terms of r-Regret. Note that the notion of fairness considered in this work requires that the sub-optimal arms be pulled some pre-specified minimum fraction of the times. Naturally, an algorithm that satisfies such fairness constraints may perform worse (in terms of the expected cumulative reward) than an algorithm that does not take such fairness considerations into account (for ex. UCB1). This leads us to evaluate the *cost of fairness* in terms of the conventional notion of regret. In particular, we show the trade-off between conventional regret and fairness in terms of the *unfairness tolerance* . The cost of fairness quantifies the trade-off in regret due to the introduction of fairness constraints, which could result in a sub-optimal arm being pulled significantly more number of times than that required to estimate its mean reward with sufficient confidence. The following theorem shows that, based on some instance-dependent threshold, the cost of fairness is either logarithmic or linear in T.

Theorem 9 *The expected regret of* Fair-ucb *is given by*

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

Here, $S(T, \alpha) = \{i \mid \alpha > r_i \cdot T - 8 \ln T / \Delta_i^2\}$.

Theorem 9 captures the explicit trade-off between regret and fairness in terms of the *unfairness tolerance* parameter α. If $S(T, \alpha) = \emptyset$, we obtain $O(\ln T)$ regret. This implies that if $\alpha > r_i T - 8 \ln T / \Delta_i^2$ for all $i \neq 1$, then the regret is $O(\ln T)$. However, if $S(T, \alpha) \neq \emptyset$ then for each $i \in S(T, \alpha)$, additional regret equal to $r_i T - \alpha$ is incurred in which results in $O(T)$ regret. We complement these results with simulations in Section 7.

6. Proof of Theoretical Results

This section provides the theoretical analysis of our results.

6.1 Fairness guarantee of Fair-learn

In this section, we show that the Fair-Learn framework achieves α-fairness guarantee irrespective of the choice of learning algorithm (Theorem 5). We begin with an overview of the proof. We will refer to the quantity $r_i t - N_{i,t}$ at the end of t rounds as the fairness potential of the arm at round $t+1$. First, for a given round t, we look at the (hypothetical) partition (induced by the previous pulls of Fair-learn) of the set of arms satisfying the fairness constraint. Each set in this partition contains the arms whose fairness potential at round $t+1$ lies within a specified range (see Figure 1). Theorem 5 is proved by showing that at any time step t and for any history of pulls by Fair-learn, each arm is contained in one of the sets constituting the partition. In particular, the theorem follows using Lemma 10, in which we prove that the number of arms in interesting sets of this partition can be (non-trivially) upper bounded. Lemma 10 is proved inductively using Observations 2 and 3. These two observations together determine how the partition (of arms) changes between two consecutive time steps, and their proof crucially uses the fact that fair-learn pulls the arm with the largest fairness potential. We now give the formal proof.

Theorem 5 *For a given $\alpha \geq 0$ and for any given fairness constraint vector $r = (r_1, r_2, \ldots, r_k)$ where $r_i \in [0, \frac{1}{k-1})$ for all $i \in [k]$ and $\sum_{i\in[k]} r_i < 1$, Fair-Learn is α-fair irrespective of the choice of the learning algorithm Learn$(\cdot)$.*

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

- For $j = 1, 2, \cdots k-1$,

$$M_{j,t} = \left\{ i \in [k] : \alpha + \frac{(k-1)-j}{k-1} \leq r_i t - N_{i,t} < \alpha + \frac{k-j}{k-1} \right\}$$

- $M_{k,t} = \left\{ i \in [k] : r_i t - N_{i,t} < \alpha \right\}$

Let $V_{j,t} = \uplus_{\ell=1}^{j} M_{\ell,t}$, for all $j \in [k]$. The following lemma guarantees the fairness of the algorithm and is at the heart of the proof. The proof of the theorem is immediate from the proof of the lemma.

Lemma 10 *For $t \geq 1$, we have*

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

From Lemma 10 we have that all the arms $i \in [k]$ satisfy $r_i t - N_{i,t} < \alpha + 1$ for all $t \geq 1$, which implies $\lfloor r_i t \rfloor - N_{i,t} \leq \alpha$. This completes the proof of the theorem. □

Proof of Lemma 10: We begin with a few observations and then prove the lemma by induction. First, we note that, if an arm is pulled by the algorithm in some round t, then it moves to either $M_{k,t+1}$ or $M_{k-1,t+1}$ at time step $t+1$ irrespective of its position at time step t. More specifically, we have the following observation.

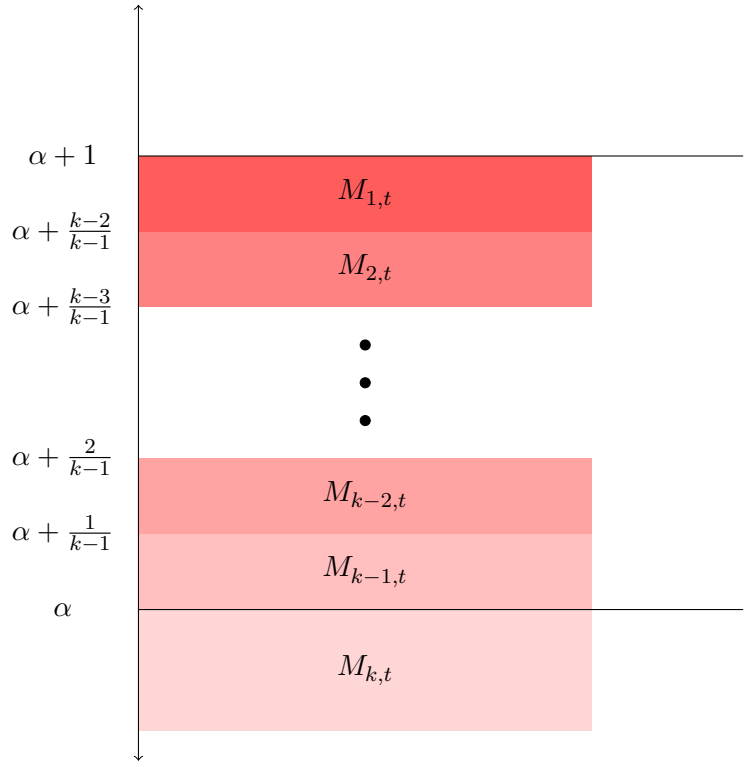


Figure 1
A k-partition of the set of arms after t rounds is given by $\{M_{i,t}\}_{i=1}^{k}$. Here, $M_{k,t}$ is the set of arms whose fairness potential after round t is strictly less than α and $M_{i,t}$ for $1 \leq i < k$ is the set of arms whose fairness potential lies in range $[\alpha + 1 - \frac{i}{k-1}, \alpha + 1 - \frac{i-1}{k-1})$.

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

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

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

$$\begin{aligned} r_i(t+1) - N_{i,t+1} &= r_i t + r_i - N_{i,t} - 1 && (3)\\ &< \alpha - (1 - r_i) && (4)\\ &< \alpha && (\text{Since } 1 - r_i > 0)\\ \implies i &\in M_{k,t+1} \end{aligned}$$

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

$$\begin{aligned} r_i(t+1) - N_{i,t+1} &= r_i t + r_i - N_{i,t} - 1 && (5)\\ &< \alpha + \frac{k-j}{k-1} - (1 - r_i) && (6)\\ &= \alpha + \frac{1-j}{k-1} + r_i && (7)\\ &\leq \alpha + r_i \;\; < \;\; \alpha + \frac{1}{k-1} && (\text{Since } r_i < \tfrac{1}{k-1})\\ \implies i &\in M_{k-1,t+1} \uplus M_{k,t+1} && (8) \end{aligned}$$

■

Next, consider the arms that are not pulled in round t. If an arm i is in set $M_{j,t}$ (for some $j \geq 2$) in round t and is not pulled then in round $t+1$ it either stays in set $M_{j,t+1}$ or moves to $M_{j-1,t+1}$. This leads to our next observation.

Observation 3 *Let $i \in [k]$ be any arm that is not pulled at time step $t+1$ and $i \in M_{j,t}$ for some $j \in [2,k]$, then $i \in M_{j,t+1} \uplus M_{j-1,t+1}$.*

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

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

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

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

and $r_i(t+1) - N_{i,t+1} = r_i t - N_{i,t} + r_i \geq \alpha + \frac{(k-1)-j}{k-1} + r_i \;\; \geq \;\; \alpha + \frac{(k-1)-j}{k-1}$

$$\implies i \in M_{j-1,t+1} \uplus M_{j,t+1}$$

■

Finally, observe that at any round t, there is always an arm whose fairness potential is less than or equal to α.

Observation 4 *For all $t \geq 1$ we have $M_{k,t} \neq \emptyset$*

Proof It is easy to see that $M_{k,1}$ is non-empty. In particular, arm pulled at time step $t = 1$ is in $M_{k,1}$. We now show that $M_{k,t} \neq \emptyset$ for $t \geq 2$. For contradiction, let, at some time step t, we have $r_i t - N_{i,t} \geq 0$ for all $i \in [k]$ this implies that $\sum_{i=1}^{k} r_i t - N_{i,t} \geq 0 \implies \sum_{i=1}^{k} r_i t \geq \sum_{i=1}^{k} N_{i,t} = t \implies \sum_{i=1}^{k} r_i \geq 1$. The last inequality contradicts the assumption that $\sum_{i\in[k]} r_i < 1$. ■

With above observations we complete the proof of the lemma using induction.
Induction base case ($t = 1$): Let i_1 be the arm pulled at $t = 1$. Then

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

$$\implies i_1 \in M_{k,1}$$

For all $i \neq i_1$, we have $r_i t - N_{i,1} = r_i < \frac{1}{k-1} \leq \alpha + \frac{1}{k-1} \implies i \in M_{k,1} \uplus M_{k-1,1}$. Hence, $V_{k,1} = [k]$, $|V_{k-1,1}| \leq k-1$, and $|V_{j,1}| = 0$ for all $j \in [k-2]$. 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: $i_{t+1} \in M_{k,t}$. From Observation 2, we know $i_{t+1} \in M_{k,t+1}$. As $i_{t+1} \in M_{k,t}$, from Observation 3, we have for any arm $i \neq i_{t+1}$ that $i \in M_{k,t+1} \uplus M_{k-1,t+1}$. Hence, $V_{k,t+1} = [k]$ and $|V_{j,t+1}| = 0$ for all $j \in [k-2]$. Furthermore from Observation 4 we have $|V_{k-1,t+1}| \leq k-1$. Thus, Conditions (1) and (2) in the lemma hold after round $t+1$.
Case 2: $i_{t+1} \in M_{a,t}$, for some $a \in [k-1]$.

$$\begin{aligned} & i_{t+1} \in M_{a,t} \implies i_{t+1} \in V_{a,t} \\ \implies & |V_{j,t}| = 0 \quad \text{for all } j \in [1, a-1] \text{ if } a > 1 \end{aligned} \tag{9}$$

From Observation 2, we know $i_{t+1} \in M_{k-1,t+1} \uplus M_{k,t+1}$. Thus, from Observation 3, we infer that $V_{j-1,t+1} \subset V_{j,t} \setminus \{i_{t+1}\}$ for all $j \in [2, k-1]$. Also,

$$\begin{aligned} & |V_{j,t} \setminus \{i_{t+1}\}| \leq j-1 \qquad \text{for all } j \in [a, k-1] \\ \implies & |V_{\ell,t+1}| \leq \ell \qquad \text{for all } \ell \in [a-1, k-2] \end{aligned} \tag{10}$$

Further, note that $\{i : r_i(t+1) - N_{i,t+1} > \alpha + 1\} = \emptyset$. This is true as, from induction argument, we have $\{i : r_i t - N_{i,t} > \alpha + 1\} = \emptyset$ and if $|M_{1,t}|$ is nonempty[6] then $M_{1,t} = \{i_{t+1}\}$ and as $r_i < \frac{1}{k-1}$ we have for all $i \neq i_{t+1}$ that $\{i : r_i t - N_{i,t} \leq \alpha + 1\} = \emptyset$. Hence, Conditions (1) of the lemma hold after round $t+1$. Condition (2) is established by Equation 9, Equation 10 and Observation 4 together with Condition (1). ■

6.2 Distribution Dependent r-Regret guarantee of Fair-Learn

The regret analysis of Fair-ucb builds on the regret analysis of UCB1 which we give in the Section 9.3.1. In Section 9.3.1 we also introduce the notations used in this proof.

6. If $M_{1,t} = \emptyset$ then we are done.

Theorem 6 *The r-Regret of* Fair-ucb *is given by*

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

where $S(T,\alpha)=\left\{i\in[k]\;\middle|\;\alpha > r_i\cdot T-\frac{8\ln T}{\Delta_i^2}\right\}$. *In particular, for large enough* T, *we get*

$$\mathcal{R}^r_{\text{Fair-ucb}}(T) \leq \left(1+\frac{\pi^2}{3}\right)\cdot\sum_{i\in[k]}\Delta_i$$

Proof The UCB1 estimate of the mean of arm i denoted as $\bar{\mu}_i(t)=\hat{\mu}_{i,N_{i,t-1}}(t-1)+c_{t,N_{i,t-1}}$, where $\hat{\mu}_{i,N_{i,t-1}}(t-1)$ is the empirical estimate of the mean of arm i when it is pulled $N_{i,t-1}$ times 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, we upper bound the expected number of times a sub-optimal arm is pulled. We do this by considering two cases dependent on the number of times the sub-optimal arm is required to be pulled for satisfying its fairness constraint. Note, from the proof of UCB1 (Theorem 12 in Section 9), that the expected number of pulls of any sub-optimal arm i is at-most $O(\frac{8\ln T}{\Delta_i^2})$. We have following two cases.

.

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 9 , Theorem 12)

Since $r_i\cdot T-\alpha\geq\frac{8\ln T}{\Delta_i^2}$, it follows from the proof of Theorem 12 in Section 9 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 $r_i\cdot T-\alpha<\frac{8\ln T}{\Delta_i^2}$

Then $\mathbb{E}[N_{i,T}]\leq\frac{8\ln T}{\Delta_i^2}+\left(1+\frac{\pi^2}{3}\right)$. 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 $S(T,\alpha)=\left\{i\in[k]\;\middle|\;\alpha > r_i\cdot T-\frac{8\ln T}{\Delta_i^2}\right\}$.

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

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

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

6.3 Distribution-free r-Regret bound for Fair-ucb

Theorem 8 *The distribution-free r-Regret of* Fair-ucb *is $O(\sqrt{T \ln T})$.*

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

$$\begin{aligned}\mathbb{E}[\mathcal{R}^r_{\mathcal{A}}(T)] &= \sum_{i\in[k]} \Delta_i \cdot \Big(\mathbb{E}[N_{i,T}] - \max\big(0, \lfloor r_i \cdot T \rfloor - \alpha\big)\Big) \\ &\leq k + \sum_{i\in[k]} \Delta_i \cdot \Big(\mathbb{E}[N_{i,T}] - \max\big(0, r_i \cdot T - \alpha\big)\Big)\end{aligned}$$

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

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

The last inequality follows from the fact that $r_i \geq 0$ for all $i \in [k]$ and that α is a constant. This implies that the r-regret for any instance with given value of $r = (r_1, r_2, \ldots, r_k)$ is upper 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. Finally, the said result follows from the distribution-free regret bound of UCB1 (see Section 9.3.2). ■

6.4 Cost of Fairness in the MAB Problem

Theorem 9 *The expected regret of* Fair-ucb *is given by*

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

Here, $S(T, \alpha) = \{i \mid \alpha > r_i \cdot T - 8 \ln T/\Delta_i^2\}$.

Proof From Section 3, Definition 3 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 $r_i \cdot T - \alpha$.

Case 1: Let $i \neq 1$ and $r_i \cdot T - \alpha \geq \frac{8 \ln T}{\Delta_i^2}$. Then

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

(Follows from Section 9.3.1)

Since $(r_i \cdot T - \alpha) \geq \frac{8 \ln T}{\Delta_i^2}$, it follows from the proof of Theorem 12 that $\mathbb{E}[N_{i,T}] \leq (r_i \cdot T - \alpha) + \left(1 + \frac{\pi^2}{3}\right)$.

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

Then the proof of Theorem 12 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

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

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

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

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

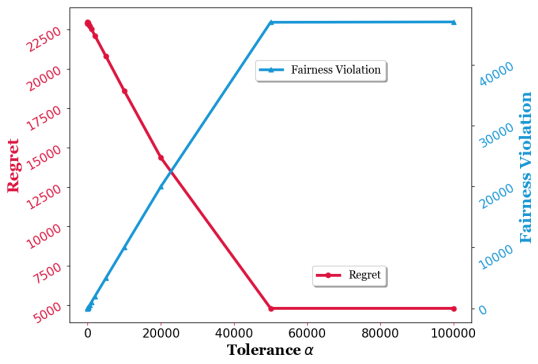
7. Experimental Results

In this section, we show the results of simulations that validate our theoretical findings. First, we represent the cost of fairness by showing the trade-off between regret and fairness with respect to the *unfairness tolerance* α. Second, we evaluate our algorithms' performance in terms of r-Regret and fairness guarantee by comparing them with the algorithm by Li et al. (2019), called Learning with Fairness Guarantee (LFG), as a baseline. We discuss the rationale behind the choice of instance parameters in Section 9.2.

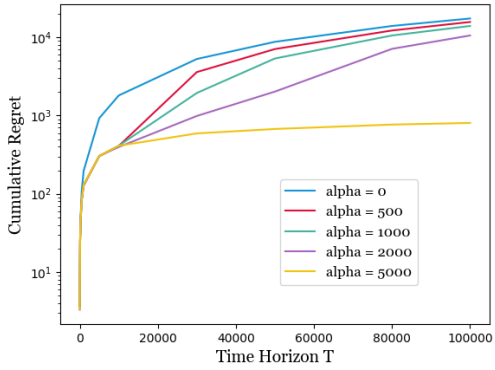
Trade-off between Fairness and Regret: For the experiments in Figure 2a, we consider a Fair-MAB instance with $k = 10$, $\mu_1 = 0.8$, and $\mu_i = \mu_1 - \Delta_i$, where $\Delta_i = 0.01i$, and $r = (0.05, 0.05, \ldots, 0.05) \in [0, 1]^k$. We show the results with regret computed over $T = 10^6$ time steps. Figure 2a shows the trade-off between regret in terms of the conventional regret and maximum fairness violation equal to $\max_{i \in [k]} r_i t - N_{i,t}$, with respect to α, and this in particular captures the *cost of fairness*. As can be seen, the regret decreases,

and maximum fairness violation increases respectively as α increases till a threshold for α is reached. For values of α less than this threshold, the fairness constraints cause some sub-optimal arms to be pulled more than the number of times required to determine its mean reward with sufficient confidence. On the other hand, for values of α more than this threshold, the regret reduces drastically, and we recover logarithmic regret as could be expected from the classical UCB1 algorithm. Note that the threshold for α is problem-dependent.

Next, in Figure 2b, we consider Fair-MAB instance with $k = 10$, $\mu = (0.8, 0.75, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.15, 0.1)$, and $r = (0.05, \ldots, 0.05)$. Note that, to highlight our results in respective settings, we use a different μ vectors in Figures 2a and 2b while keeping the other instance parameters the same. Here, we show how the cumulative regret varies as α takes different values. As expected, the cumulative regret decreases as the unfairness tolerance α increases.



(a) Cost of Fairness

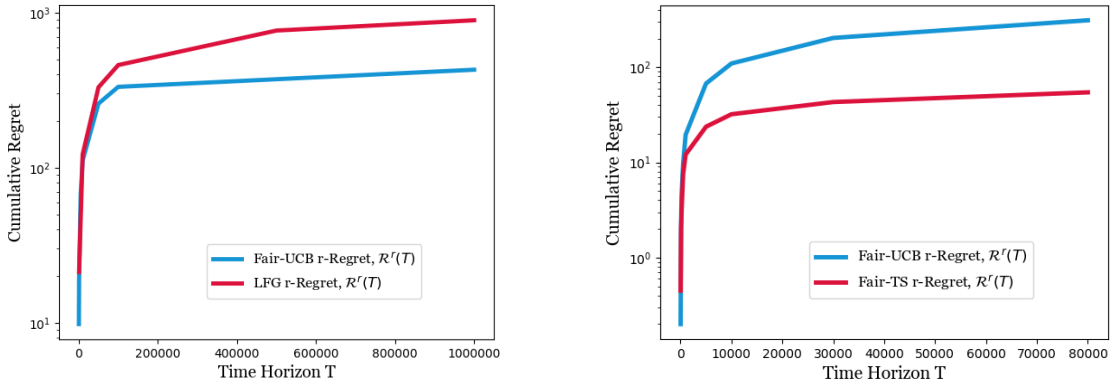


(b) Variation in Regret with α

Figure 2
Figure (a) captures the trade-off between the cumulative regret at $T = 10^6$ and the unfairness tolerance α. Figure (b) shows the growth of cumulative regret over $T = 10^5$ rounds for different values of α.

Comparison of Regret Guarantees: We now compare the r-Regret guarantee of Fair-UCB with other algorithms. The work closest to ours is by Li et al. (2019) 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 Figure 2a. Figure 3a 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 result of logarithmic r-Regret bound. Further, in Figure 3b, we compare the performance of Fair-UCB with Fair-Thompson Sampling (Fair-TS). Fair-TS is an instance of Fair-Learn where the black-box learning algorithm is chosen to be Thompson Sampling Thompson (1933). As in the MAB problem without fairness constraints, the fair variant of Thompson Sampling also converges faster than UCB but provides a comparable r-Regret guarantee in order terms.

We next compare fairness guarantee of Fair-UCB with that of LFG. We consider an instance with $k = 3$, $\mu = (0.7, 0.5, 0.4)$, $r = (0.2, 0.3, 0.25)$ and, $\alpha = 0$. Figure 4a shows



(a) r-Regret : Fair-UCB vs. LFG

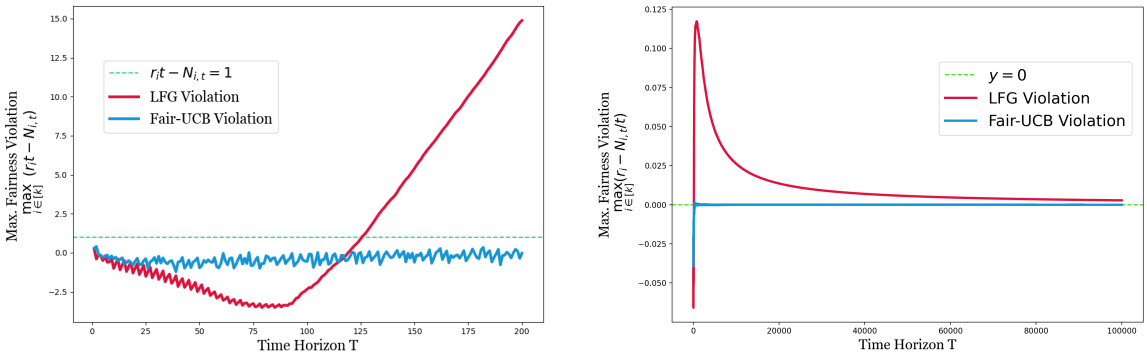
(b) r-Regret : Fair-UCB vs. Fair-TS

Figure 3
In Figure (a), we compare the r-Regret of Fair-ucb and LFG over $T = 10^6$ rounds. In Figure (b), we compare the r-Regret of Fair-ucb and Fair-TS over $T = 8 \times 10^4$. We see that Fair-TS converges faster than Fair-ucb, similar to their respective variants for the MAB problem without fairness constraints. The cumulative regret is plotted on a logarithmic scale.

the plot of time vs. maximum fairness violation in terms of the number of arm pulls, i.e., $\max_{i\in[k]} r_i t - N_{i,t}$. Observe that the fairness guarantee of Fair-ucb holds uniformly over the time horizon T. On the other hand, the fairness violation of LFG can increase up to a significantly large value as they only provide an asymptotic fairness guarantee. This implies that, in the initial rounds, the algorithm could be unfair to some arms. In Figure 4b we plot $\max_{i\in[k]} r_i - N_{i,t}/t$, which is the per round fairness violation. For Fair-ucb (and any other algorithm in Fair-Learn), this quantity immediately approaches zero, whereas for LFG it only asymptotically goes to zero.

Figure 4a only shows the plot over $T = 200$ time steps to provide a better contrast in the fairness guarantees of Fair-ucb (or any algorithm in Fair-Learn) and LFG. Figure 4b is plotted over $T = 10^5$ to depict the asymptotic fairness guarantee of Li et al. (2019). As the red curve in Figure 4b goes to zero, the red curve in Figure 4a plateaus (not shown in the figure).To summarize, the simulation results reaffirm our theoretical guarantees for both fairness and r-Regret of Fair-Learn in general, and Fair-ucb in particular.

Periodicity Property of allocation: In Figure 5a, we observe that the allocation returned by the proposed algorithm, Fair-Learn, satisfies a certain periodicity property. That is, any arm i with $r_i > 0$ is allocated exactly one round in $\lceil 1/r_i \rceil$ number of rounds after a sufficiently large time T_i. We consider a Fair-MAB instance with $k = 5$ and reward distributions are Bernoulli with means $\mu = (0.9, 0.5, 0.3, 0.1, 0.1)$, $r_i = 0.01$ for all i and $T = 2.5 \times 10^4$. We observe that after a certain T_i, when the optimal arm has been identified with high enough confidence, the sub-optimal arm i is only pulled to satisfy its fairness constraint. That is, in any time window of size $\lceil \frac{1}{r_i} \rceil$, arm i is pulled exactly once by the algorithm. We note that the time T_i depends on the sub-optimality of the arm. The arms with lower values of mean rewards observe periodic allocations sooner than others i.e. $T_i > T_j$ for $i > j$ (since we have assumed $\mu_1 > \mu_2 > \ldots > \mu_k$). This is expected as the arms with lower mean rewards can be separated from the optimal arms with fewer samples.



(a) Fairness Violation

(b) Fairness Convergence

Figure 4

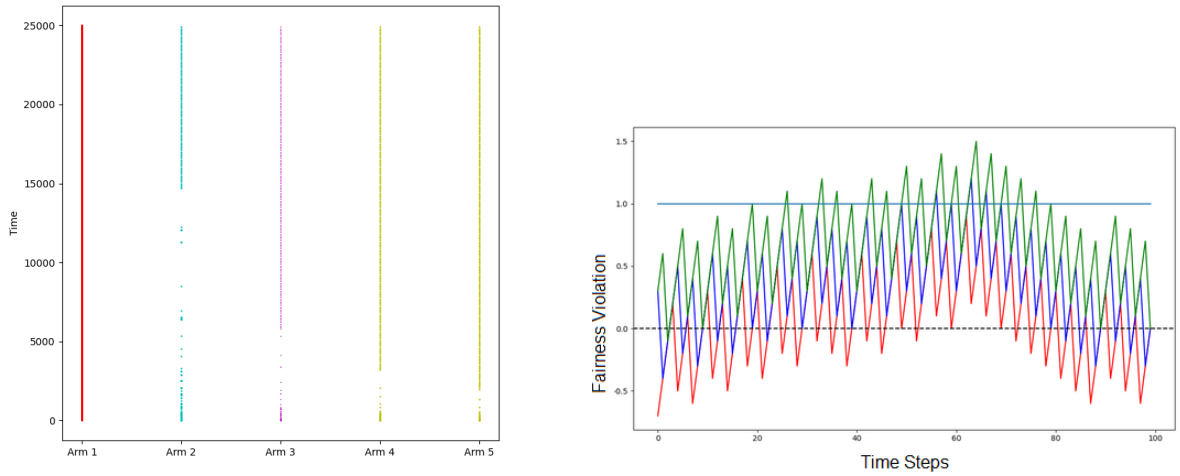
Figure (a) shows the number of rounds in which fairness is violated over time. Figure (b) shows the fraction of rounds in which fairness is violated over time. The two figures are complementary to each other since the red curve in Figure (a) will plateau after some number of rounds (>20000, not shown in the plot for better resolution) which is confirmed by the red curve asymptotically going to zero in Figure (b).

Note that in many real-world applications, it is important to spread out the allocation of opportunities evenly over time. That is, no individual should be starved of opportunities for a long period of time. However, Fair-Learn may not provide this guarantee. In particular, arm 2 is allocated only 9 rounds in the period starting from round 5000 to round 15000. We leave the design of an algorithm that guarantees the uniform allocations over time as an interesting future direction.

As proven in Theorem 5, Fair-Learn is α-fair when $r_i \in [0, 1/(k-1))$ for all $i \in [k]$ and $\sum_{i\in[k]} r_i < 1$. We next show via simulations that, for some instances, when there exists an arm $i \in [k]$ with $r_i > 1/(k-1)$, Fair-Learn may not be α-fair. Figure 5b provides one such instances where the Y-axis shows the value of $r_i t - N_{i,t}$ for the top-3 arms in terms of the fairness violation value, and the X-axis shows the time steps. Recall that for Fair-Learn to be α-fair, we should have $r_i t - N_{i,t} < \alpha + 1$. We consider the following instance: $k = 12$ and $r = (0.3, 0.3, 0.3, 0.01$ repeated 9 times) with $\mu_i = (0.8 - i * 0.001)$. It can be noted in the simulations that, given an unfairness tolerance α ($= 0$ in this case), even though Fair-Learn is not α-fair, it is in fact $(\alpha + 1)$-fair i.e. $r_i t - N_{i,t} < 2 = ((\alpha + 1) + 1)$. We observed this behaviour in all other instances we considered for this simulation. This leads us to conjecture that for a general fairness constraint vector $r \in [0, 1)^k$ with $\sum_{i\in[k]} r_i < 1$, Fair-Learn is in fact $(\alpha + 1)$-fair, which is still an extremely strong fairness guarantee.

8. Discussion and Future Work

The constraints considered in this paper capture fairness by guaranteeing a minimum fraction of pulls to each arm at all times. There are many situations where such fairness constraints are indispensable, and in such cases the r-Regret notion compares the expected loss of any online algorithm with the expected loss of an optimal algorithm that also satisfies such fairness constraints. An important feature of our proposed meta algorithm Fair-Learn is



(a) Periodicity of Fair-Learn

(b) Counter Example for General r

Figure 5
Figure (a) shows the periodic nature of Fair-Learn, and Figure (b) demonstrates one instance where, given that some $r_i > 1/k - 1$, Fair-Learn is not α-fair.

the uniform time fairness guarantee that it provides irrespective of the learning algorithm used. We also elucidate the cost of imposing such fairness constraints by evaluating the trade-off between the conventional regret and fairness in terms of an unfairness tolerance parameter. Additionally, we also provide detailed simulations to validate our theoretical results with respect to the fairness guarantee of Fair-Learn and regret guarantee of Fair-ucb. Finally, with extensive simulations, some important observations are drawn about the periodicity of the allocation returned by the Fair-Learn framework.

Notions of fairness such as disparate impact, statistical parity, equalized odds have been extensively studied in the machine learning literature (see Barocas et al. (2019)). Incorporating such fairness notions in online learning framework, as done by Blum et al. (2018); Blum and Lykouris (2020); Bechavod et al. (2019), is an exciting future direction.

Our proof of uniform time fairness guarantee requires that the fairness quota assured for each arm is at most $\frac{1}{k-1}$. Even though this case is well-motivated in the context of individual fairness, from a theoretical perspective, proving similar uniform time fairness guarantee when the fairness quota allocated for one or more of the arms exceeds $\frac{1}{k-1}$ remains unresolved. We conjecture that given one can achieves a α-fair fairness guarantee in this case.

Another immediate direction for future work could be to study the trade-off between fairness and regret in other variants of Multi-armed Bandits such as adversarial bandits, combinatorial bandits (with general reward structure), contextual bandits, Markovian bandits.

9. Supplimentary Material

9.1 Horizon-aware Algorithms

An algorithm that has access to time horizon T and has to satisfy fairness constraints only at the end of T rounds (and not uniformly at the end of all rounds) 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. Notice that now the fairness constraints are only satisfied after T' rounds. Guided by this intuition we propose a UCB1 based *T-aware* algorithm called T-FAIR-UCB algorithm that satisfies the fairness requirement at the end of T rounds and achieves logarithmic r-Regret.

Algorithm 2: T-FAIR-UCB

Input: $[k], (r_i)_{i\in[k]}$
Output: $i_1, i_2, \cdots, i_T$
Initialize: $n_i \leftarrow \max\left(1, r_i \cdot T\right)$ for each $i \in [k]$ and $T' = \sum_{i\in[k]} n_i$;
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**
  - $i_t = \arg\max_{i\in[k]} \overline{\mu}_i(t)$
  - Update $\overline{\mu}_i(t+1)$
end

UCB1 based Algorithm (T-FAIR-UCB): This T-FAIR-UCB algorithm knows the time horizon T, and effectively separates the *fairness constraint satisfaction* phase and the *regret minimization* phase and achieves logarithmic r-Regret in terms of T with dependence on the values of the fairness fractions. T-FAIR-UCB is presented in Algorithm 2. Note that T-FAIR-UCB satisfies the fairness requirements of all arms at T' itself, but does not provide uniform time fairness guarantee as FAIR-UCB . Next we show that T-FAIR-UCB achieves logarithmic r-Regret.

Theorem 11 *For* FAIR-MAB *problem,* T-FAIR-UCB *has r-Regret $\mathcal{R}^r_{\text{T-FAIR-UCB}}(T) = O(\ln T)$. In particular, its r-dependent regret is given by*

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

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 12 (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

$$\begin{aligned}\mathbb{E}[N_{i,T}] &\leq r_i \cdot T + \sum_{t=T'+1}^{T} \mathbb{1}\{i_t = i, N_{i,t-1} \geq r_i \cdot T\} \\ &\leq r_i \cdot T + \sum_{t=T'}^{\infty} \sum_{s=1}^{t} \sum_{s_i = r_i \cdot T}^{t} \mathbb{1}\Big\{\hat{\mu}_{1,s}(t) + c_{t,s} \leq \hat{\mu}_{1,s_i}(t) + c_{t,s_i}\Big\}\end{aligned}$$

(Follows from Section 9.3)

Since $r_i \cdot T \geq \frac{8\ln T}{\Delta_i^2}$, it follows from the proof of Theorem 12 that $\mathbb{E}[N_{i,T}] \leq r_i \cdot T + \left(1 + \frac{\pi^2}{3}\right)$. Hence, the expected number of pulls of a sub-optimal arm $i \neq 1$ in the regret minimization phase is $\mathbb{E}[N_{i,T}] - r_i \cdot T \leq \left(1 + \frac{\pi^2}{3}\right)$.
Case 2: Let $i \neq 1$ and $r_i \cdot T < \frac{8\ln T}{\Delta_i^2}$
Then the proof of Theorem 12 can be appropriately adapted to show that $\mathbb{E}[N_{i,T}] \leq \frac{8\ln T}{\Delta_i^2} + \left(1 + \frac{\pi^2}{3}\right)$. Thus the expected number of pulls of a sub-optimal arm $i \neq 1$ in the regret minimization phase is

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

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

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

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

9.2 Rationale for Simulation Parameters

For evaluating the performance of our algorithm, we perform experiments on synthetic data sets as this allows for finer control on the tuning of the parameters of the experiment. In

particular, we consider the following two Fair-MAB instances:

Instance 1: Fairness vs. Regret

- Number of arms: $k = 10$
- Time horizon: $T = 10^6$
- Mean rewards: $\mu_1 = 0.8$ and $\mu_i = \mu_1 - \Delta_i$ where $\Delta_i = 0.01i$
- Fairness constraint: $r = (0.05)_{i \in [k]}$

Instance 2: FUCB vs. LFG

- Number of arms: $k = 3$
- Time horizon: $T = 200$
- Mean rewards: $\mu = (0.7, 0.5, 0.4)$
- Fairness constraints: $r = (0.2, 0.3, 0.25)$

Note that a large value for the time horizon T significantly increases the simulation time. On the other hand, a small value for T does not capture the convergence of a MAB algorithm. In Instance 1, we choose a sufficiently large value of T so that the convergence of Fair-ucb is captured. On the other hand, we use a smaller value of T in Instance 2 because it allows us to capture the maximum fairness violation at each round more clearly.

Next, we use Instance 1 to evaluate the performance of the algorithms in terms of r-Regret , whereas Instance 2 is used to evaluate the fairness guarantee. First, we consider the number of arms chosen. If the number of arms is very small, a learning algorithm will correctly identify the optimal arm very soon. On the other hand, a larger number of arms increases the simulation time required to depict the convergence behaviour of the algorithm. Our choice of k in Instance 1 is sufficient to depict the behaviour of our algorithm in terms of regret without significantly increasing the simulation time. On the other hand, we choose a small value of k in Instance 2 to showcase the fairness guarantee. Keeping k small allows us more flexibility in terms of choosing r_i's (since $\sum_{i \in [k]} r_i < 1$). Thus, by choosing r_i's of sub-optimal arms such that the number of times Fair-ucb is required to play these arms is significantly more than that by classic UCB1, allows us to test the fairness guarantee of Fair-ucb.

Our choice of the expected rewards of the arms, $\mu = (\mu)_{i \in [k]}$ in Instance 1 is such that the difference in the expected rewards of two adjacent arms is small. Consequently, the algorithm needs more time to correctly decide the optimal arm. Furthermore, we also carried out simulations with μ_i's with a higher difference between them. However, our current choice of μ_i's captures the contrast in the regret performance of the Fair-MAB algorithms much better. In contrast, the choice of μ in Instance 2 is because the fairness guarantee of Fair-ucb can be tested more rigorously when the differences in μ_i's is significant as this causes the algorithm to correctly identify the optimal arm quickly. In the standard UCB algorithm, this would lead to the sub-optimal arms being pulled a significantly fewer number of times.

As a result, choosing greater values of r_i's for these arms allows for more strict evaluation of the fairness guarantees of Fair-ucb.

The fairness constraint vector $r = (r_i)_{i\in[k]}$ in Instance 1 is again selected such that it provides a clear depiction of the cost of fairness in terms of the conventional notion of regret. The choice of r in Instance 2 allows for more meticulous assessment of the fairness guarantees of the two algorithms. We have also carried out the experiments with different values of the fairness constraint vector but our choice turns out to be the one suitable for the purpose of representation.

9.3 Additional Preliminaries

The proofs of some of our results (Theorem 6 and Theorem 9) rely on the regret analysis of UCB1 algorithm Auer et al. (2002), which we provide below for completeness.

9.3.1 Upper Confidence Bound (UCB) based Algorithm

In this section we describe the UCB1 algorithm that was introduced by Auer et al. (2002) and for completeness we also give a proof of its regret bound. In the UCB1 algorithm for each arm the algorithm maintains a UCB1 estimate and at each round the algorithm plays the arm with the highest UCB1 estimate. Such a UCB1 estimate 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 pulled in $t-1$ rounds. Then the UCB1 estimate 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 pulled $N_{i,t-1}$ 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 pulled 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 UCB1 estimate for each arm and for every round $t \geq k+1$ it plays the arm with the highest UCB1 estimate. The total expected regret of UCB1 after 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 12 *For the* MAB *problem, the* UCB*1 has expected regret* $\mathbb{E}[\mathcal{R}_{\text{UCB}}(T)] \leq \sum_{i\neq 1}\left(\frac{8\ln T}{\Delta_i}\right) + \left(1+\frac{\pi^2}{3}\right)\sum_{i\in[k]}\Delta_i$.

Proof To bound the regret of the UCB1 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 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 ℓ we may rewrite the above equation as

$$N_{i,T} \leq \ell + \sum_{t=\ell}^{T} \mathbb{1}\Big\{ i_t = i, N_{i,t-1} \geq \ell \Big\} \tag{11}$$

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 11

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

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

$$\big\{ \hat{\mu}_{1,s_1}(t-1) \leq \mu_1 - c_{t,s_1} \big\} \tag{12}$$

$$\big\{ \hat{\mu}_{i,s_i}(t-1) \geq \mu_i + c_{t,s_i} \big\} \tag{13}$$

$$\big\{ \mu_1 < \mu_i + 2c_{t,s_i} \big\} \tag{14}$$

The probability of the events in Equations 12 and 13 can be bounded using Hoeffding's inequality as:

$$\mathbb{P}\Big(\big\{ \hat{\mu}_{1,s_1}(t-1) \leq \mu_1 - c_{t,s_1} \big\} \Big) \leq t^{-4}$$

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

The event in equation 14 $\big\{ \mu_1 < \mu_i + 2c_{t,s_i} \big\}$ 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

$$\mathbb{P}\left(\left\{ \Delta_i - 2\sqrt{\frac{2\ln t}{s_i}} < 0 \right\} \right) = 0 \tag{15}$$

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

$$N_{i,T} \leq \lceil \frac{8 \ln T}{\Delta_i^2} \rceil + \sum_{t=\frac{8 \ln T}{\Delta_i^2}}^{T} \sum_{s_1=1}^{t} \sum_{s_i=\frac{8 \ln T}{\Delta_i^2}}^{t} \mathbb{1}\Big\{ \hat{\mu}_{1,s_1}(t-1) + c_{t,s_1} < \hat{\mu}_{i,s_i}(t-1) + c_{t,s_i} \Big\}$$

$$\begin{aligned} \mathbb{E}[N_{i,T}] &\leq \lceil \frac{8 \ln T}{\Delta_i^2} \rceil + \sum_{t=\frac{8 \ln T}{\Delta_i^2}}^{T} \sum_{s_1=1}^{t} \sum_{s_i=\frac{8 \ln T}{\Delta_i^2}}^{t} 2t^{-4} \\ &\leq \lceil \frac{8 \ln T}{\Delta_i^2} \rceil + \sum_{t=\frac{8 \ln T}{\Delta_i^2}}^{\infty} \sum_{s_1=1}^{t} \sum_{s_i=\frac{8 \ln T}{\Delta_i^2}}^{t} 2t^{-4} \leq \frac{8 \ln T}{\Delta_i^2} + 1 + \frac{\pi^2}{3} \end{aligned}$$

In the last inequality we use $\sum_{t=\lceil \frac{8 \ln T}{\Delta_i^2} \rceil}^{\infty} \sum_{s_1=1}^{t} \sum_{s_i=\lceil \frac{8 \ln t}{\Delta_i^2} \rceil}^{t} 2t^{-4} \leq \sum_{t=1}^{\infty} 2t^{-2} = \frac{\pi^2}{3}$. Recall from Section 3, Equation 3, that

$$\mathbb{E}[\mathcal{R}_{\mathrm{UCB}}(T)] = \sum_{i \in [k]} \Delta_i \cdot \mathbb{E}[N_{i,T}] \leq \sum_{i \neq 1} \frac{8 \ln T}{\Delta_i} + \Big(1 + \frac{\pi^2}{3}\Big) \cdot \sum_{i \in [k]} \Delta_i$$

■

9.3.2 *Distribution-free Regret Bound for* UCB*1*

Theorem 13 *For the* MAB *problem, the* UCB*1 has expected (distribution-free) regret* $\mathbb{E}[\mathcal{R}_{\mathrm{UCB}}(T)] = O\big(\sqrt{T \ln T}\big)$.

Proof Recall from Section 9.3.1 that the expected cumulative regret of the UCB1 algorithm in any round T is given by

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

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

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

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

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

Next, we will bound the value of $\mathbb{E}\big[\mathcal{R}_{\mathrm{UCB}}(T)\big]$ by conditioning on C and $\bar{C}$. Let us first bound $\mathbb{E}\big[\mathcal{R}_{\mathrm{UCB}}(T)\big|C\big]$. Assume the event C holds and some arm $i_t \neq 1$ is pulled 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,

$$\mu_1 - \mu_{i_t} \leq 2 \cdot \sqrt{\frac{2\ln T}{N_{i_t,t-1}}} \tag{16}$$

Now, consider any arm $i \in [k]$ and consider the last round $t_i \leq t$ when this arm was last pulled. Since the arm has not been pulled between t_i and t, we know $N_{i,t_i} = N_{i,t-1}$. Hence, applying the inequality in Equation 16 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

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

Therefore, we have

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

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

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

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

Acknowledgments

VP is grateful for the financial support from the Ministry of Education, Govt. of India. GG is thankful for the financial support from the Israeli Ministry of Science and Technology grant 19400214. VN is thankful to be funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 682203 -ERC-[Inf-Speed-Tradeoff]. Also, a large part of this work was done when GG and VN were graduate students at the Indian Institute of Science.

References

A. Agarwal, A. Beygelzimer, M. Dudik, J. Langford, and H. Wallach. A reductions approach to fair classification. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 60–69, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018.

S. Agrawal and N. Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *Proceedings of the 25th Annual Conference on Learning Theory*, volume 23 of *Proceedings of Machine Learning Research*, pages 39.1–39.26, Edinburgh, Scotland, 25–27 Jun 2012.

K. Amin, M. Kearns, P. Key, and A. Schwaighofer. Budget optimization for sponsored search: Censored learning in mdps. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, page 54–63, 2012.

P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2–3):235–256, May 2002. ISSN 0885-6125.

A. Badanidiyuru, R. Kleinberg, and A. Slivkins. Bandits with knapsacks. *Journal of the ACM*, 65(3), Mar. 2018. ISSN 0004-5411.

S. Barocas, M. Hardt, and A. Narayanan. *Fairness and Machine Learning*. fairmlbook.org, 2019. `http://www.fairmlbook.org`.

Y. Bechavod, K. Ligett, A. Roth, B. Waggoner, and S. Z. Wu. Equal opportunity in online classification with partial feedback. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

R. Berk, H. Heidari, S. Jabbari, M. Joseph, M. Kearns, J. Morgenstern, S. Neel, and A. Roth. A convex framework for fair regression. *arXiv preprint arXiv:1706.02409*, 2017.

A. Beutel, J. Chen, T. Doshi, H. Qian, L. Wei, Y. Wu, L. Heldt, Z. Zhao, L. Hong, E. H. Chi, and C. Goodrow. Fairness in recommendation ranking through pairwise comparisons. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 2212–2220, 2019.

A. Blum and T. Lykouris. Advancing Subgroup Fairness via Sleeping Experts. In *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*, volume 151, pages 55:1–55:24, 2020.

A. Blum, S. Gunasekar, T. Lykouris, and N. Srebro. On preserving non-discrimination when combining expert advice. In *Advances in Neural Information Processing Systems*, pages 8376–8387, 2018.

S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

L. E. Celis, D. Straszak, and N. K. Vishnoi. Ranking with fairness constraints. In *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, volume 107, pages 28:1–28:15, 2018.

L. E. Celis, S. Kapoor, F. Salehi, and N. Vishnoi. Controlling polarization in personalization: An algorithmic framework. FAT* '19, page 160–169, 2019.

A. Chatterjee, G. Ghalme, S. Jain, R. Vaish, and Y. Narahari. Analysis of thompson sampling for stochastic sleeping bandits. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*, 2017.

C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel. Fairness through awareness. In *Theoretical Computer Science Conference*, pages 214–226, 2012.

S. Freeman. Original position. In E. N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, 2019.

S. Gillen, C. Jung, M. Kearns, and A. Roth. Online learning with an unknown fairness metric. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

H. Heidari, C. Ferrari, K. Gummadi, and A. Krause. Fairness behind a veil of ignorance: A welfare analysis for automated decision making. In *Advances in Neural Information Processing Systems*, pages 1265–1276. 2018.

N. Immorlica, K. A. Sankararaman, R. E. Schapire, and A. Slivkins. Adversarial bandits with knapsacks. In *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 202–219, 2019.

M. Joseph, M. Kearns, J. Morgenstern, and A. Roth. Fairness in learning: classic and contextual bandits. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 325–333, 2016.

R. Kleinberg, A. Niculescu-Mizil, and Y. Sharma. Regret bounds for sleeping experts and bandits. *Machine learning*, 80(2):245–272, 2010.

T. Lattimore and C. Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

T. Lattimore, K. Crammer, and C. Szepesvári. Optimal resource allocation with semi-bandit feedback. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, pages 477–486, 2014.

T. Lattimore, K. Crammer, and C. Szepesvári. Linear multi-resource allocation with semi-bandit feedback. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*, pages 964–972, 2015.

F. Li, J. Liu, and B. Ji. Combinatorial sleeping bandits with fairness constraints. *IEEE Transactions on Network Science and Engineering*, 7(3):1799–1813, 2019.

Y. Liu, G. Radanovic, C. Dimitrakakis, D. Mandal, and D. C. Parkes. Calibrated fairness in bandits. *arXiv preprint arXiv:1707.01875*, 2017.

H. Narasimhan. Learning with complex loss functions and constraints. In *International Conference on Artificial Intelligence and Statistics*, pages 1646–1654, 2018.

J. Rawls. *A theory of justice*. Harvard university press, 1971.

A. Rezaei, R. Fathony, O. Memarrast, and B. D. Ziebart. Fair logistic regression: An adversarial perspective. *CoRR*, abs/1903.03910, 2019.

A. Singh and T. Joachims. Fairness of exposure in rankings. In *International Conference on Knowledge Discovery & Data Mining*, pages 2219–2228, 2018.

A. Singh and T. Joachims. Policy learning for fairness in ranking. In *Advances in Neural Information Processing Systems*, 2019.

A. Slivkins et al. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2):1–286, 2019.

M. S. Talebi and A. Proutiere. Learning proportionally fair allocations with low regret. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 2(2):36, 2018.

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.

L. Tran-Thanh, L. Stavrogiannis, V. Naroditskiy, V. Robu, N. R. Jennings, and P. Key. Efficient regret bounds for online bid optimisation in budget-limited sponsored search auctions. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, pages 809–818, 2014.

Y. Xia, H. Li, T. Qin, N. Yu, and T.-Y. Liu. Thompson sampling for budgeted multi-armed bandits. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pages 3960–3966, 2015.

M. B. Zafar, I. Valera, M. Gomez Rodriguez, and K. P. Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *International Conference on World Wide Web*, pages 1171–1180, 2017a.

M. B. Zafar, I. Valera, M. G. Rogriguez, and K. P. Gummadi. Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*, pages 962–970, 2017b.

M. Zehlike, F. Bonchi, C. Castillo, S. Hajian, M. Megahed, and R. Baeza-Yates. Fa*ir: A fair top-k ranking algorithm. In *ACM Conference on Information and Knowledge Management*, pages 1569–1578, 2017.