Multi-agent Multi-armed Bandits with Minimum Reward Guarantee Fairness

Piyushi Manupriya
IIT Hyderabad
Hyderabad, India
cs18m20p100002@iith.ac.in

Himanshu
IIT Hyderabad
Hyderabad, India
ai22mtech12008@iith.ac.in

SakethaNath Jagarlapudi
IIT Hyderabad
Hyderabad, India
saketha@cse.iith.ac.in

Ganesh Ghalme
IIT Hyderabad
Hyderabad, India
ganeshghalme@ai.iith.ac.in

ABSTRACT

We investigate the problem of maximizing social welfare while ensuring fairness in a multi-agent multi-armed bandit (MA-MAB) setting. In this problem, a centralized decision-maker takes actions over time, generating random rewards for various agents. Our goal is to maximize the sum of expected cumulative rewards, a.k.a. social welfare, while ensuring that each agent receives an expected reward that is at least a constant fraction of the maximum possible expected reward.

Our proposed algorithm, RewardFairUCB, leverages the Upper Confidence Bound (UCB) technique to achieve sublinear regret bounds for both fairness and social welfare. The fairness regret measures the positive difference between the minimum reward guarantee and the expected reward of a given policy, whereas the social welfare regret measures the difference between the social welfare of the optimal fair policy and that of the given policy.

We show that RewardFairUCB algorithm achieves instance-independent social welfare regret guarantees of $\tilde{O}(T^{1/2})$ and a fairness regret upper bound of $\tilde{O}(T^{3/4})$. We also give the lower bound of $\Omega(\sqrt{T})$ for both social welfare and fairness regret. We evaluate RewardFairUCB's performance against various baseline and heuristic algorithms using simulated data and real world data, highlighting trade-offs between fairness and social welfare regrets.

CCS CONCEPTS

• **Theory of computation → Online learning algorithms**.

KEYWORDS

Multi-armed Bandits, Fairness, Regret Analysis, Multi-agent systems

ACM Reference Format:
Piyushi Manupriya, Himanshu, SakethaNath Jagarlapudi, and Ganesh Ghalme. 2025. Multi-agent Multi-armed Bandits with Minimum Reward Guarantee Fairness. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 9 pages.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

1 INTRODUCTION

In a classical stochastic Multi-armed Bandit (MAB) problem, a central decision maker takes an action or, equivalently, selects an arm from a fixed set of arms at each of T time steps. Each arm pull yields a random reward from an unknown distribution. The objective is to develop a strategy for selecting arms that minimizes the regret; difference between the cumulative rewards of the best possible arm-pulling strategy in hindsight and the cumulative expected rewards achieved by the algorithm's policy.

We study an interesting variant of stochastic MAB problem, first proposed by Hossain et al. [9] and known as Multi-agent Multi Armed Bandits (MA-MAB). In the MA-MAB setting an arm pull generates a vector-valued reward whose each entry is independently sampled from a fixed but unknown distribution denoting reward obtained by corresponding agent. When there is a single agent, this setting reduces to a classical stochastic MAB setting.

The MA-MAB setting captures several interesting real-world applications. Consider, for instance, the problem of distributing a fixed monthly/yearly budget, say one unit, among k different projects. There are n beneficiaries (or agents) who each experience varying levels of benefit from the different projects. Each agent $i \in [n]$ receives a reward sampled independently from distribution $\mathcal{D}(\mu_{i,j})$ when the algorithm pulls an arm j (or equivalently, selects project j) where $\mu_{i,j}$ denotes the mean reward for agent i from arm j. The randomness in the reward received by agents may arise from uncertainty in the assessment of the value of the project by individual agents and randomness in the aggregation/reporting step. Given a distribution $\pi \in \Delta_m$ over the set $[m]$ of arms, the total expected reward to agent i is given by $\sum_{j\in[m]} \mu_{i,j}\pi_j$.

Consider another example where a networked TV channel must decide which movie/program to telecast on a given time slot. The different movie/program genres are the arms, whereas the population group (based on age group, demographics, etc.) are the agents. The reward represents the preferences of the corresponding agent, a.k.a. age group. The decision-maker's problem here is to telecast the most liked program that, at the same time, caters to the preferences of a diverse population. We will return to this example in Section 7.

It is easy to see that when the goal is to maximize social welfare [1], the resulting arm pull strategy/allocation strategy might become skewed. For example, consider a MA-MAB instance with n agents

[1] Defined as the sum of cumulative expected reward.

and $m = 2$ arms with reward distributions such that $\mu_{1,1} = 1$ and $\mu_{1,2} = 0$, and $\mu_{i,1} = 0$ and $\mu_{i,2} = 1/n$ for $i > 1$. In this case, a social welfare maximizing policy would allocate the entire budget to the first project. Similarly, select the movie genre most preferred by the entire population in the second example. However, this policy benefits only the first agent, leaving the vast majority of $n - 1$ agents without any reward. Such winner-takes-all allocations can be considered unfair in many applications and can lead to undesirable long-term dynamics leading to mistrust towards the algorithm [9]. The MA-MAB framework with fairness constraints facilitates the simultaneous optimization of both individual rewards and overall societal welfare.

Consider a thought experiment where a single agent dictates the arm-pulling policy. [2] This dictatorial agent would choose to pull her most rewarding arm at every time step. However, this policy completely disregards the preferences of other agents and thus fails to ensure any minimum reward for them. In this paper, we address the problem of ensuring a minimum reward guarantee for each agent as an explicit constraint. Specifically, each agent i is guaranteed at least a certain fraction $C_i \in [0, 1]$ of the maximum possible reward they could receive.

1.1 Related Work

The stochastic multi-armed bandits (MAB) problem has been extensively studied with the goal of designing algorithms that optimally trade-off exploration and exploitation for maximizing the expected cumulative reward [5, 13, 20]. The multi-agent multi-armed bandits (MA-MAB) variant [14] involves multiple agents simultaneously solving a given instance of the MAB problem. Such a setting often demands providing reward fairness guarantees to each agent, besides maximizing the sum of expected cumulative rewards obtained by the agents. Several works have emerged focusing on fairness for the MAB problem [3, 12, 15–19, 21]. However, these approaches do not generalize to provide reward guarantees for different agents involved in the MA-MAB setup. Moreover, these formulations either focus on guaranteeing a certain fraction of arm pulls [16, 18, 19], constrain the deviation of the policy to a specific closed-form optimal policy [3, 21] or focus on meritocratic criteria in online resource allocation setting [15, 17].

The closest work to ours is by Hossain et al. [9] who proposed learning a policy over the m-arms that maximizes the Nash Social Welfare (NSW) involving the n agents. Jones et al. [10] proposed a more efficient algorithm for the NSW-based MA-MAB problem and recently Zhang et al. [22] tried improving the corresponding regret bounds. While NSW objective is known to satisfy desirable fairness and welfare properties (see [6, 11] for details), the fairness guarantees in NSW are implicit and cannot be specified externally. This may not always be desired. Consider a case with 2 agents with $\mu_{1,1} = \mu_{2,2} = 1$ and $\mu_{1,2} = \mu_{2,1} = 0$ where the agents demand at least one-third and two-thirds of their maximum possible reward. Maximizing the NSW results in the policy $(1/2, 1/2)$ which does not satisfy the specified reward requirement for the second agent. In contrast, our proposed MA-MAB formulation finds a policy that respects the reward allocations demanded by each agent, whenever it is possible to do so.

[2] Alternatively, consider a scenario where there is only one agent, i.e., $n = 1$.

1.2 Main Results and Organization of the Paper

We propose a novel formulation for the multi-agent multi-armed bandits (MA-MAB) problem to maximize social welfare obtained from the rewards while also guaranteeing each agent a specified fraction of their maximum possible reward. In Section 2, we formally define the problem and provide sufficient conditions under which a fair MA-MAB instance is guaranteed to have a feasible solution. Then, in Section 3, we consider an MA-MAB instance with 2 arms and $n > 1$ agents and show that a simple Explore-First algorithm achieves a simultaneous regret bound of $\tilde{O}(T^{2/3})$ for both fairness and social welfare.

In Section 4, we propose the main algorithm of this paper, RewardFairUCB , and show that it achieves the regret guarantee of $\tilde{O}(\sqrt{T})$ for the social welfare regret and $\tilde{O}(T^{3/4})$ for the fairness regret. In Section 5, we prove lower bounds of $\Omega(\sqrt{T})$ for both social welfare regret and fairness regret. These lower bounds hold independently for the regret notions. We then provide a dual formulation based heuristic algorithm in Section 6 that achieves a better regret performance on the simulated data and real-world datasets (Section 7). The main results of the paper are summarized in Table 1.

2 SETTING AND PRELIMINARIES

We write $[n]$ to denote the set $\{1, 2, \cdots, n\}$. Further, we will assume that the set of agents ($[n]$) and the set of arms ($[m]$) are both finite sets. Let $\mathcal{D}(\mu_{i,j})$ denote a probability distribution with finite mean $\mu_{i,j}$. Further, let the random variable $X_{i,j}$ denote a random reward obtained by agent i from arm j, that is $X_{i,j} \sim \mathcal{D}(\mu_{i,j})$. Finally, we X_i to denote a vector of size m with i^{th} entry $X_{i,j}$.

A fair Multi-Agent Multi-Armed Bandit (MA-MAB) instance $\mathcal{I}$ is denoted by a tuple $\langle A, C, T\rangle$ where,

- A denotes an $n \times m$ non-negative matrix with each entry $A_{i,j} := \mu_{i,j}$. Note that A is fixed but unknown to the algorithm. We will assume, without loss of generality [3] that $A_{i,j} \in [0, 1]$ for all $i \in [n]$ and $j \in [m]$. Further, define $\mathbf{A}^\star$ to be a matrix in $[0, 1]^n$ whose i^{th} entry represents the maximum possible expected reward to agent i. $\mathbf{A}^\star_i = \max_{j\in[m]} A_{i,j}$.
- C denotes $n \times n$ non-negative diagonal matrix. The entry $C_i := C_{i,i} \in [0, 1]$ specifies the fraction of maximum possible rewards to be guaranteed to agent i. Note that C is a predefined constant and does not change with time.
- T denotes the stopping time of the algorithm. We assume that T is known a priori. However, all our results can be extended to unknown time horizon setting using a doubling trick (see [4]) with an additional constant multiple factor increase in the regret.

We now formally define the notion of minimum-reward guarantee fairness regret for a given MA-MAB instance $\mathcal{I}$. We begin by first defining the notion of minimum-reward guarantee.

Definition 1 (Minimum Reward Fairness Guarantee)**.** *Let $\mathcal{I} = \langle A, C, T\rangle$ be a MA-MAB instance and let $\mathbf{A}^\star$ be the vector of maximum values from the corresponding row of A. We say that a policy π satisfies*

[3] It is easy to see that if all the entries are divided by the largest row entry, the optimal strategy does not change.

	Social Welfare Regret	Fairness Regret	Remark
Lower Bound (Sec 5)	$\Omega(\sqrt{T})$	$\Omega(\sqrt{T})$	The lower bounds hold individually for social welfare regret and fairness regret.
Explore-First (Sec 3)	$\tilde{O}(T^{2/3})$	$\tilde{O}(T^{2/3})$	For two arms, $C_i = c \leq 0.5\ \forall i \in [n]$.
RewardFairUCB (Sec 4)	$\tilde{O}(\sqrt{T})$	$\tilde{O}(T^{3/4})$	For any finite number of arms. Optimal (up to logarithmic factor) social welfare regret.

Table 1: Key findings of the paper.

minimum reward fairness guarantee for $\mathcal{I}$, if

$$A\pi \geq C\mathbf{A}^\star. \tag{1}$$

Throughout the paper, we will call a policy π fair if it satisfies minimum fairness guarantees for any fair MA-MAB instance $\mathcal{I}$.

We observe that there may not always exist a fair policy. Consider, an example with 2 agents and 2 arms with $A_{1,1} = A_{2,2} = 1$ and $A_{1,2} = A_{2,2} = 0$ and $C_1 = C_2 = c \in [0, 1]$. It is straightforward to see that no policy π satisfies the minimum reward fairness guarantee for instances with $c > 0.5$. However, a fair policy always exists for $c \leq 0.5$. In particular, $\pi = [0.5, 0.5]$ is one such fair policy.

In our first result of the paper, we provide sufficient conditions that guarantee the existence of fair policy for a given instance $\mathcal{I}$

Theorem 1. *A fair MA-MAB instance $\mathcal{I}$ admits a fair policy if at least one of the below conditions is satisfied*

(1) $\sum_{i\in[n]} C_i \leq 1$,

(2) $C_{\max} := \max_{i\in[n]} C_i \leq \frac{1}{\min(n,m)}$.

For the first condition, observe that the policy $\pi_j = \frac{\sum_{i=1}^n \mathbb{1}(j=j_i)C_i}{\sum_{i=1}^n C_i}$ where j_i being the arm with largest reward to agent i is a fair policy [4]. Under the second condition, uniform arm pull policy i.e., $\pi_j = [1/m, \cdots, 1/m]$ is feasible. A formal proof of Theorem 1 is given in the Appendix.

Under a learning setting where the algorithm is not privy to A, the algorithm must learn the policy (π^t) from the history of past pulls and observed rewards, denoted by $\mathcal{H}^t$. More specifically, for a given time instance t, an arm pull strategy π^t is a mapping, $\pi^t : \mathcal{H}^t \to \Delta_m$. The minimum-reward fairness regret of the policy $\pi := (\pi^t)_{t\geq 1}$ is defined the cumulative *positive* difference between promised expected reward and the expected reward under the policy π^t.

Definition 2 (Minimum-reward Fairness Regret)**.** *Given a MA-MAB instance $\mathcal{I}$ and a policy π, the minimum-reward fairness regret of π on instance $\mathcal{I}$ over T time instances is given as*

$$\mathcal{R}^\pi_{FR}(T) = \sum_{t=1}^{T}\sum_{i=1}^{n} |\underbrace{C_i\mathbf{A}^\star_i}_{I} - \underbrace{\mathbb{E}_{\pi^t}[X^t_i]}_{II}|_+, \tag{2}$$

where $|\cdot|_+ \equiv \max\{\cdot, 0\}$.

The term labelled **I** indicates the minimum rewards as a fraction of the maximum possible expected reward that agent i is guaranteed, while the term labelled **II** represents the expected reward that agent i receives under the policy π^t at time t. The use of the expression $|\cdot|_+$ allows us to capture the scenario where, if the reward received by the agent exceeds the minimum required to satisfy the fairness constraints, the fairness regret incurred is zero. Therefore, the total fairness regret is accumulated across all agents up to time T, reflecting the extent to which the agents' reward under policy π^t deviates from their minimum guarantees.

2.1 Social Welfare Maximization with Minimum-reward Fairness Guarantee

Let $\mathrm{SW}_\pi(T) := \sum_{t=1}^T \sum_{i\in[n]} \langle A_i, \pi^t\rangle$ represent the total expected social welfare achieved by the policy $\pi = (\pi^t)_{t\geq 1}$ over the time horizon, T.

P1

$$\begin{aligned} \text{Maximize}_{\pi=(\pi_1,\pi_2,\cdots,\pi_T)} \quad & \mathrm{SW}_\pi(T) \\ \text{subject to} \quad & A\pi^t \geq C\cdot\mathbf{A}^\star \quad \forall t \in [T] \end{aligned} \tag{3}$$

It is easy to see that the optimal fair policy π^* pulls each arm with the same probability in each round, i.e., $\pi^*_i = \pi^t_i$ for all t since matrices A and C are fixed.

We further assume that the conditions in Theorem 1 are satisfied, i.e., π^* is well defined. The reward regret is defined as the cumulative loss in social welfare by not following the policy π^* at each time instant.

Connection with Nash Social Welfare: The Nash Social Welfare (NSW) objective is known to satisfy fairness guarantees in resource allocation scenarios (see [6, 9] and references therein). However, it falls short in accommodating user-defined fairness requirements. Additionally, the primary aim of NSW is not to maximize social welfare, which is the central objective of our work. Interestingly, our proposed formulation reveals an equivalence with the Nash product. Specifically,

$$P1 \equiv \arg\max_{\pi\in\Delta_m} \underbrace{\underbrace{\Pi^n_{i=1}\left(e^{\langle A_i,\pi\rangle}\right)}_{\text{Nash product for rewards}} \underbrace{\Pi^n_{i=1}\left(\mathbb{1}_{\langle A_i,\pi\rangle - C_i\mathbf{A}^\star_i \geq 0}\right)}_{\text{Nash product for fairness}}}_{\text{Nash product for rewards and fairness}}.$$

In this formulation, the Nash product for fairness reaches its maximum value of one only when the fairness guarantees are met

[4] If $\sum_i C_i = 0$, then every policy $\pi \in \Delta_m$ is feasible.

for all agents. Provided that a feasible policy exists, the Nash product for rewards in our formulation can be interpreted as an NSW configuration, where the agents' rewards are exponentiated.

Next, we define the social welfare regret as the additional loss incurred by the algorithm as compared to the optimal fair policy in hindsight.

Definition 3 (Social Welfare Regret). *Let π^* be an optimal policy (solution of problem P1) for a given MA-MAB instance $\mathcal{I}$. Further, let $\pi = (\pi^t)_{t\geq 1}$ be an arm pull strategy. The social welfare regret of π on instance $\mathcal{I}$ over time horizon T is defined as*

$$\mathcal{R}^{\pi}_{SW}(T) = T \cdot SW(\pi^*) - \sum_{t=1}^{T} SW(\pi^t) \tag{4}$$

We drop the superscript in the notation of fairness regret and social welfare regret whenever the arm-pull strategy π is clear from the context. Note that the expected cumulative SW regret could well be negative, in which case the policy π generates more social welfare than an optimal fair policy at the cost of fairness regret.

Throughout the paper, we will consider that the reward functions $X_{i,j}$'s are sub-gaussian random variables with finite and positive mean.

Definition 4 (Sub-gaussian Rewards). *We call X a sub-gaussian random variable if there is a positive constant σ such that for every $\lambda \in \mathbb{R}$, we have*

$$\mathbb{E}\left[\exp\left(\lambda(X - \mathbb{E}[X])\right)\right] \leq \exp(\lambda^2\sigma^2/2). \tag{5}$$

Sub-gaussian random variables encompass a diverse range of distributions, including Bernoulli random variables. More generally, any random variable bounded in $[a, b]$ is σ-sub-gaussian with $\sigma = \frac{(b-a)}{2}$. The sub-gaussian property ensures that the probability of extreme reward values is minimized, which contributes to better reward guarantees. This characteristic is particularly advantageous for designing and analyzing learning algorithms, allowing us to consider a more general class of reward distributions.

3 WARMUP: TWO ARMS CASE

In this section, we consider a simple MA-MAB setup with 2 arms and n agents. This setup allows us to write the optimal fair policy in tractable mathematical form. We also provide our first algorithm, Explore-First.

Consider the MA-MAB instance with 2 arms and n agents. Index the agents such that the first n_1 agents prefer arm 1 and the next $n-n_1$ agents prefer arm 2. Note that when $n_1 = n$ (when $n_1 = 0$), we have that all the agents prefer arm 1 (arm 2) and the optimal fair policy, in this case, is straightforward: pull arm 1 (or arm 2 respectively) with probability 1. Hence, without loss of generality, let $0 < n_1 < n$. That is, $\mathbf{A}^\star_i = A_{i,1} (\geq A_{i,2})$ for all $i \in [n_1]$, $\mathbf{A}^\star_i = A_{i,2} (> A_{i,1})$ for all $i \notin [n_1]$. Further assume without loss of generality that arm 1 is an optimal arm i.e. $\sum_{i\in[n]} A_{i,1} \geq \sum_{i\in[n]} A_{i,2}$ and let $[x^*, 1-x^*]$ be the optimal arm pulling policy.

To characterize the optimal arm pulling strategy in the two-arms case, first observe the following property of the optimal fair policy. An optimal fair policy pulls a sub-optimal arm (arm 2) with a nonzero probability i.e. $1 - x^* > 0$ only when the minimum reward fairness guarantee is violated for some agent. With this intuitive understanding, we now characterize the optimal policy $[x^*, 1-x^*]$.

Let $\Delta := \sum_{i=1}^{n}(A_{i,1} - A_{i,2}) > 0$. The regret of the policy π can be written in terms of Δ as follows

$$\mathcal{R}_{\text{SW}}(T) = \sum_{t=1}^{T} [x^* - x^t]\Delta. \tag{6}$$

Here, x^t is the probability of pulling arm 1 at time t.

Lemma 1. *The optimal feasible policy of a fair MA-MAB instance $\mathcal{I}$ with two arms is given by*

$$x^* = \min\left(1, \min_{i\in[n]\setminus[n_1]} \frac{1-C_i}{1-\frac{A_{i,1}}{A_{i,2}}}\right). \tag{7}$$

The proof of Lemma 1 is given in the Appendix. We are now ready to present our first algorithm that achieves a sublinear regret guarantee.

Algorithm 1 Explore-First

1: **Require:** T, C.
2: **for** $t = 1, 2, \cdots, \lfloor T^\alpha \rfloor$ **do**
3: Pull arm $i = t \mod (2) + 1$.
4: **end for**
5: Compute the estimated reward matrix $\widehat{A}$ of the rewards observed so far.
6: Compute $x' = \min\left(1, \min_{i:\widehat{A}_{i,2}>\widehat{A}_{i,1}} \frac{1-C_i}{1-\frac{\widehat{A}_{i,1}}{\widehat{A}_{i,2}}}\right)$.
7: **for** $t = \lfloor T^\alpha \rfloor + 1, \lfloor T^\alpha \rfloor + 2, \cdots, T$ **do**
8: Pull arm 1 with probability x' and arm 2 with probability $1 - x'$.
9: **end for**

3.1 Regret Analysis of Explore-First Algorithm

The Explore-First algorithm addresses the exploration-exploitation tradeoff effectively by delineating the exploration phase from the exploitation phase. During the exploration phase, the algorithm employs a round-robin strategy to pull each arm for $\lfloor T^\alpha \rfloor$ rounds. This approach ensures that each arm is sampled sufficiently, yielding more accurate estimates of each arm's reward. However, this phase does not prioritize the arm with the highest reward and does not guarantee immediate rewards for the agents.

In the subsequent exploitation phase, the algorithm utilizes these reward estimates to solve an optimization problem P1. For the specific case of two arms, a closed-form solution is given in Line 6. The optimal fair policy derived from this solution is then used to determine the arm pulls for the remaining rounds.

It is important to note that the regrets associated with both social welfare and fairness are influenced by the choice of the parameter α. Specifically, the regret incurred during the exploration phase is proportional to T^α in both cases. Thus, a larger value of α results in higher regret due to the increased duration of the exploration phase. Conversely, if α is too small, the estimates of the arm rewards may not be sufficiently accurate, leading to suboptimal decisions in the exploitation phase and, consequently, higher regret. This tradeoff highlights the importance of carefully choosing α to obtain a balance between accurate reward estimation and minimizing regret.

Theorem 2. *[Informal] The Explore-First algorithm achieves,*

(1) expected social welfare regret of $O\left(\frac{n}{a_{\min}} T^{2/3}\sqrt{\log(T)}\right)$, *and*

(2) expected fairness regret of $O\left(\frac{n}{a_{\min}} T^{2/3}\sqrt{\log(T)}\right)$, *where* $a_{\min} = \min_{i,j} A_{i,j} > 0$.

Detailed proof of Theorem 2 is given in the Appendix. It is easy to see that the Explore-First algorithm is inadequate for both fairness and social welfare. Firstly, observe that the algorithm fails to collect information gathered during the exploit phase and, thus, ceases learning after the exploration phase. This impacts both social welfare and fairness regret guarantees, as inaccurate estimates can result in suboptimal policies. Next, we propose a UCB-based policy that provides a better tradeoff in terms of social welfare regret and fairness regret.

4 THE PROPOSED ALGORITHM AND ANALYSIS

At each time step t, our proposed algorithm RewardFairUCB (refer to Algorithm 2) keeps an Upper Confidence Bound (UCB) estimate and a Lower Confidence Bound (LCB) for every arm-agent pair (i, j). During the initial t' rounds (Lines 2-7), the RewardFairUCB performs exploration, i.e. pulls the arms in a round-robin manner. In the following exploitation phase (i.e., $t \geq t'$), the algorithm keeps UCB and LCB estimates for each arm-agent combination. The UCB index is utilized to provide an optimistic estimate of social welfare, while both UCB and LCB indices are used to assess the fairness requirements to determine the arm-pulling strategy as given in problem P2 below.

P2

$$\begin{aligned} \text{Maximize}_{\pi\in\Delta_m} \quad & \sum_{i=1}^{n} \langle \overline{A}_i, \pi \rangle \\ \text{subject to} \quad & \overline{A}\pi \geq C \cdot \underline{\mathbf{A}^\star} \end{aligned} \tag{8}$$

While using the UCB index to estimate rewards is common in literature and used in virtually all UCB-based algorithms, the use of LCB to estimate fairness constraints is not common. We employ the LCB estimate to ease the fairness constraints in P2, ensuring that the below two properties hold with high probability,

(1) the social welfare guarantees remain intact, and
(2) the fairness constraints are met.

Our proof crucially uses the above two properties of the solution obtained by solving the linear program P2. In particular, we show the optimal solutions of P2 exhibit similar social welfare with a small loss in fairness guarantee in comparison with the solution of P1.

We begin our analysis with a standard result in probability theory.

Lemma 2 (Hoeffding's inequality for sub-Gaussian random variables)**.** *Let $Z_1, Z_2, \ldots, Z_k$ be independent sub-Gaussian random variables, each with sub-Gaussian parameter σ and let $S_k = \frac{1}{k}\sum_{s=1}^{k} Z_s$. Then for all $\varepsilon > 0$, we have*

$$P\left(|S_k - \mathbb{E}[S_k]| > \varepsilon\right) \leq 2\exp\left(-\frac{k\varepsilon^2}{2\sigma^2}\right).$$

Alternatively, for any $\delta \in (0, 1]$, with probability at least $1 - \delta$,

$$|S_k - \mathbb{E}[S_k]| \leq \sigma\sqrt{\frac{2\log\left(\frac{2}{\delta}\right)}{k}}.$$

We now prove an important technical lemma.

Lemma 3. *Let π^* be an optimal feasible solution of P1 and for any $t \geq t'$, π^t be an optimal solution of P2 with $\overline{A} := \overline{A}^t$. Then with probability at-least $1 - 1/\sqrt{T}$, we have*

$$SW_{\pi^t}(\overline{A}) \geq SW_{\pi^*}(A).$$

Proof. Let $\varepsilon^t_{i,j} = \sigma\sqrt{\frac{2\log(4mn\sqrt{T})}{N^t_j}}$ (See Line 9 of RewardFairUCB algorithm) and define $\delta'_{i,j} := \exp\left(\frac{-N^t_j\varepsilon^2}{2\sigma^2}\right) = \frac{1}{4mn\sqrt{T}}$. Further, let $\delta = 1/\sqrt{T}$. Note that $\delta'_{i,j}$ is the probability that $\overline{A}^t_{i,j} = \widehat{A}^t_{i,j} + \varepsilon^t_{i,j} \leq A_{i,j}$ at some time $t \geq t'$.

By symmetry of tail bounds around the mean value given by Hoeffding's inequality (Lemma 2) we have that $\delta'_{i,j}$ is also the probability that $A_{i,j} \geq \underline{A}^t_{i,j}$.

Note that arms are pulled in a round-robin fashion in the exploration phase of the RewardFairUCB algorithm. This implies that each arm is pulled for the same number of rounds, i.e., $N^{t'}_j$ is the same for all $j \in [m]$. Hence, we have that $\delta' := \delta'_{i,j} = \frac{1}{4mn\sqrt{T}}$.

We prove the stated claim by showing that with probability at least $1 - \frac{1}{\sqrt{T}}$, every feasible policy π of P1 is also a feasible policy of P2.

Fix $i \in [n]$. Let $k \in \arg\max_{j\in[m]} A_{i,j}$ and $k^t \in \arg\max_{j\in[m]} \underline{A}^t_{i,j}$ be the least indexed arms with maximum value in the i^{th} row of matrices A and $\underline{A}$ respectively.

We have

$$\begin{aligned} \langle \overline{A}_i, \pi \rangle \underbrace{\geq}_{w.p. \geq 1-m\delta'} & \langle A_i, \pi \rangle \geq C_i \cdot A_{i,k} \geq C_i \cdot A_{i,k^t} \\ \underbrace{\geq}_{w.p. \geq 1-m\delta'} & C_i \cdot (\widehat{A}_{i,k^t} - \varepsilon^t_{k^t}) = C_i \cdot \underline{A}_{i,k^t}. \end{aligned} \tag{9}$$

The first inequality (from left) in Equation 9 above follows from Hoeffding's inequality (Lemma 2), the second inequality follows from the feasibility of π for P1, the third inequality follows from the definition of k^t and the last inequality again follows from Hoeffding's inequality (Lemma 2). The first and last inequalities each hold with probability at least $(1 - m\delta')$. Hence we have with probability at-least $1 - 2m\delta'$ that $\langle \overline{A}_i, \pi \rangle \geq C_i \cdot \underline{A}_{i,k^t}$.

Using union bound, we have with probability at-least $1 - 2nm\delta'$ that $\langle \overline{A}_i, \pi \rangle \geq C_i \cdot \underline{A}_{i,k^t}$ for all $i \in [n]$; i.e. every feasible solution π of P1 is also a feasible solution of P2. In particular, π^* is a feasible solution of P2 with probability at least $1 - \delta/2$. This, along with the definition of π^t gives

$$SW_{\pi^t}(\overline{A}) \underbrace{\geq}_{w.p. \geq 1-\delta/2} SW_{\pi^*}(\overline{A}) \underbrace{\geq}_{w.p. \geq 1-\delta/2} SW_{\pi^*}(A)$$

This proved the stated claim. □

Algorithm 2 RewardFairUCB

Require: $T, n, m, C, N_j^t = 0 \; \forall j \in [m]$.
$t' = m\lceil\sqrt{T}\rceil$, $t = 1$, $\widehat{A} = \mathbf{0}_{m\times n}$.
for $t \le t'$ **do**
  Pull arm $j' = t \bmod m + 1$.
  $\forall i \in [n]$, observe reward $X_{i,j'}^t \sim \mathcal{D}(\mu_{i,j'})$.
  $\forall i \in [n], \forall j \in [m]$, $\widehat{A}_{i,j} = \begin{cases} \widehat{A}_{i,j} & \text{if } j \neq j' \\ \frac{(N_j^{t-1})\widehat{A}_{i,j} + X_{i,j}^t}{N_j^t} & \text{if } j = j'. \end{cases}$
  $N_{j'}^t = N_{j'}^{t-1} + 1$.
end for
Compute the confidence matrix $\mathcal{E}$ with entries

$$\epsilon_{i,j}^t = \sigma\sqrt{\frac{2\log(8mnT)}{N_j^t}} \; \forall i \in [n], j \in [m].$$

for $t \le T$ **do**
  Compute $\overline{A} = \widehat{A} + \mathcal{E}$ and $\underline{A} = \widehat{A} - \mathcal{E}$.
  $\forall i \in [n]$, compute $\underline{\mathbf{A}^\star}_i = \max_{j\in[m]} \underline{A}_{i,j}$ (break ties arbitrarily).
  Solve **P2** and let π' be the solution of this LP.
  Sample $j' \sim \pi'$.
  $\forall i \in [n]$, observe reward $X_{i,j'}^t \sim \mathcal{D}(\mu_{i,j'})$.
  $N_{j'}^t = N_{j'}^{t-1} + 1$.
  $\forall i \in [n], \forall j \in [m]$, $\widehat{A}_{i,j} = \begin{cases} \widehat{A}_{i,j} & \text{if } j \neq j' \\ \frac{(N_j^{t-1})\widehat{A}_{i,j} + X_{i,j}^t}{N_j^t} & \text{if } j = j'. \end{cases}$
  Update entries of $\mathcal{E}$.
end for

Lemma 3 implies that by easing the fairness constraints, it is possible to increase social welfare.

Theorem 3. *For any feasible MA-MAB instance $\mathcal{I}$ with $T \ge 32n^2\sigma^2$, expected social welfare regret of RewardFairUCB is upper-bounded by*

$$4n\sqrt{2T}\left(\sigma\log(2m^2T) + m + \sigma\right).$$

The detailed proof of Theorem 3 is provided in the Appendix. We provide a high-level overview of the proof here. First, we break down the regret into two components: R1 and R2, representing the regret from the exploration phase and exploitation phase, respectively. The component R1 encompasses the regret over the initial exploration phase of $m\lceil\sqrt{T}\rceil$ rounds. Following this, using Lemma 3 we argue that the social welfare obtained by solving P2 is at-least that of the original problem with high probability. By applying Hoeffding's inequality and the union bound to the aggregated expected values, we show that R2 is capped by $\tilde{O}(T^{1/2})$. We emphasize here that the social welfare regret of RewardFairUCB is asymptotically optimal (refer to Section 5 for the lower bound).

We now give the fairness regret guarantee of RewardFairUCB algorithm.

Theorem 4. *For any feasible MA-MAB instance $\mathcal{I}$ with $T \ge 32n^2\sigma^2$, expected fairness regret of RewardFairUCB is upper-bounded by*

$$6n(\max_{i\in[n]} C_i)\sigma T^{3/4}\log(2m^2T) + mnO(\sqrt{T}).$$

It is worth noting that while the social welfare regret guarantee of RewardFairUCB is much stronger, this comes at the cost of higher fairness regret. We demonstrate this trade-off in Section 7 with simulations. Next, we show the lower bound on fairness and social welfare regret guarantees of MA-MAB problems,

5 REGRET LOWER BOUNDS

In this section, we prove that every algorithm must suffer an *instance-independent regret* [5] of $\Omega(\sqrt{T})$ in both fairness and social welfare.

Theorem 5. *An instance-independent social welfare and fairness regrets of MA-MAB problem is lower bounded by* $(\Omega(\sqrt{T}), \Omega(\sqrt{T}))$.

Proof Outline. The proof of Theorem 5 is given in the Appendix. We provide an intuition here. To show the lower bound on social welfare, consider a class of instances where each row is a non-negative multiple of the first row i.e. $A_i = \beta_i A_1$ for some $\beta_i \ge 0$ and C as a zero matrix. As every agent has the same preferences over arms, the problem of maximizing social welfare now is reduced to the problem of identifying an arm j with the highest $\sum_{i\in[n]} A_{i,j}$. This is equivalent to finding an arm j with largest $\left(1+\sum_{i\neq 1}\beta_i\right)\cdot A_{1,j}$. This problem is the same as the classical stochastic MAB problem with m arms and reward distributions with the mean reward of arm j as $\left(1 + \sum_{i\neq 1}\beta_i\right)\cdot A_{1,j}$. We use the $\Omega(\sqrt{T})$ instance-independent regret lower bound [1, Theorem 5.1] for classical stochastic bandits to lower bound the social welfare regret.

To lower-bound the fairness regret, we construct a MA-MAB instance with $m = 2$ arms and $n = 2$ agents with A being the Identity matrix. For any values of $C_1 > 0$ and $C_2 = 1 - C_1$ satisfying the conditions in Theorem 1, the fairness criteria is satisfied if and only if $x^* = C_1$. Since $A_{1,1} - A_{1,2} = -(A_{2,1} - A_{2,2}) = 1$ we have that the fairness regret can be written (using Eq. 6) as

$$\mathcal{R}_{\text{FR}}(T) = \sum_{t=1}^{T} |C_1 - x^t|_+ \ge |\sum_{t=1}^{T} C_1 - \sum_{t=1}^{T} x^t|_+ \ge \sum_{t=1}^{T} C_1 - \sum_{t=1}^{T} x^t.$$

Now consider a stochastic MAB setup with two arms and rewards 1 and 0 respectively. We again use a lower bound of $\Omega(\sqrt{T})$ for the stochastic MAB setting with this instance to obtain a lower bound on the fairness regret for the MA-MAB setting. □

It is worth noting that the lower bounds presented in Theorem 5 invoke different stochastic MAB instances and, therefore, may not hold simultaneously. The RewardFairUCB algorithm proposed in this paper is asymptotically optimal up to a logarithmic factor; however, the same is not true for the fairness regret.

In the next section, we present a heuristic algorithm that provides better empirical performance for fairness. However, the empirical performance for the social welfare of this heuristic algorithm is worse than that of RewardFairUCB.

[5] A regret guarantee is called instance independent if it holds for every feasible MA-MAB instance $\mathcal{I}$. That is, for all values of fairness constraints matrix C, time horizon T, and mean rewards matrix A provided that $\mathcal{I}$ admits a feasible policy.

6 DUAL-BASED ALGORITHM

We now present a heuristic dual-based algorithm inspired by the dual formulation of our optimization problem P1 (Eq. 3). This algorithm pulls each arm in round-robin fashion for the first $O(\sqrt{T})$ rounds and solves the problem using a Lagrangian dual formulation with the appropriate estimates of A using pulls from the first phase.

We start with formulating the dual of our Problem P1.

$$\textbf{Primal:} - \min_{\pi \in \Delta_m} \sum_{i=1}^{n} -\langle A_i, \pi \rangle$$
$$\text{s.t. } -(\langle A_i, \pi \rangle - C_i \mathbf{A}_i^\star) \leq 0 \; \forall i \in [n]. \tag{10}$$

We derive the Lagrangian dual with Lagrange parameters $\lambda_i|_{i=1}^n$ corresponding to the fairness constraints.

$$\begin{aligned}
\textbf{Dual:} \quad & - \max_{\lambda \in \mathbb{R}^n_{\geq 0}} \min_{\pi \in \Delta_m} \sum_{i=1}^{n} -\langle A_i, \pi \rangle - \sum_{i=1}^{n} \lambda_i (\langle A_i, \pi \rangle - C_i \mathbf{A}_i^\star) \\
& = - \max_{\lambda \in \mathbb{R}^n_{\geq 0}} \min_{\pi \in \Delta_m} -\left\langle \sum_{i=1}^{n} (1+\lambda_i) A_i, \pi \right\rangle + \sum_{i=1}^{n} \lambda_i C_i \mathbf{A}_i^\star \\
& = - \max_{\lambda \in \mathbb{R}^n_{\geq 0}} - \max_{j \in [m]} \left(\sum_{i=1}^{n} (1+\lambda_i) A_i \right)_j + C_i \langle \lambda, \mathbf{A}^\star \rangle \\
& = - \max_{\lambda \in \mathbb{R}^n_{\geq 0}} - \| \, (\mathrm{Diag}(1+\lambda)A)^\top \mathbf{1}_n \|_\infty + C_i \langle \lambda, \mathbf{A}^\star \rangle
\end{aligned} \tag{11}$$

where $\mathrm{Diag}(\cdot)$ denotes the diagonal matrix formed by the entries $(1+\lambda_i)$. Motivated by the simplification we obtain in the last step, our dual algorithm is designed to pick the arms based on the UCB estimate of $\| \, (\mathrm{Diag}(1+\lambda)A)^\top \mathbf{1}_n \|_\infty$ with λ as the solution of Eq. (11).

Algorithm 3 Dual-Inspired Algorithm.

Require: $T, n, m, C, N_j^0 = 0 \; \forall j \in [m]$.
$t' = m\lceil \sqrt{T} \rceil$, $t = 1$, $\widehat{A} = \mathbf{0}_{m \times n}$.
for $t \leq t'$ **do**
  Pull arm $j' = t \bmod m + 1$.
  $\forall i \in [n]$, observe reward $X_{i,j'}^t \sim \mathcal{D}(\mu_{i,j'})$.
  $\forall i \in [n], \forall j \in [m]$, $\widehat{A}_{i,j} = \begin{cases} \widehat{A}_{i,j} & \text{if } j \neq j' \\ \frac{(N_j^{t-1})\widehat{A}_{i,j} + X_{i,j}^t}{N_j^t} & \text{if } j = j'. \end{cases}$
  $N_{j'}^t = N_{j'}^{t-1} + 1$.
end for
Compute $\mathcal{E}$ with entries $\varepsilon_{i,j}^t = \sigma \sqrt{\frac{2 \log(8mnT)}{N_j^t}}$.
Compute $\hat{\lambda}$ by solving the convex program in Eq. (11) with $\hat{A}$.
for $t \leq T$ **do**
  $j' \in \arg\max \left(\left(\mathrm{Diag}(1+\hat{\lambda})\hat{A} \right)^\top \mathbf{1}_n + \mathcal{E} \right)$.
  $\forall i \in [n]$, observe reward $X_{i,j'}^t \sim \mathcal{D}(\mu_{i,j'})$.
  $N_{j'}^t = N_{j'}^{t-1} + 1$.
  $\forall i \in [n], \forall j \in [m]$, $\widehat{A}_{i,j} = \begin{cases} \widehat{A}_{i,j} & \text{if } j \neq j' \\ \frac{(N_j^{t-1})\widehat{A}_{i,j} + X_{i,j}^t}{N_j^t} & \text{if } j = j'. \end{cases}$

  Update entries of $\mathcal{E}$.
end for

7 EXPERIMENTAL EVALUATION

We now empirically validate the sublinear regret guarantees of the proposed algorithms and the efficacy of RewardFairUCB.

7.1 Common Experimental Setup

The distribution $\mathcal{D}(\mu_{i,j})$ is taken $\mathrm{Ber}(\mu_{i,j})$ i.e. when an arm $j' \in [m]$ is sampled, agent $i \in [n]$ obtains a reward drawn from Bernoulli with parameter $\mu_{i,j}$. We plot the average regrets after simulating with 100 runs for different realizations of randomly sampled rewards. Complementing our analysis for Explore-First in the 2-arm case, we empirically find that the same regret guarantees for $m > 2$ hold with same optimal exploration parameter valued, i.e., $\alpha = 0.67$. For this, we replace Step (6) of Explore-First algorithm with the solution of the linear program P1 obtained with empirical reward estimates $\hat{A}$. The CVXPY [7] library is used to solve the linear programs wherever required in our algorithms. More results with different A matrices are presented in the Appendix.

7.2 Experiments on Simulated Data

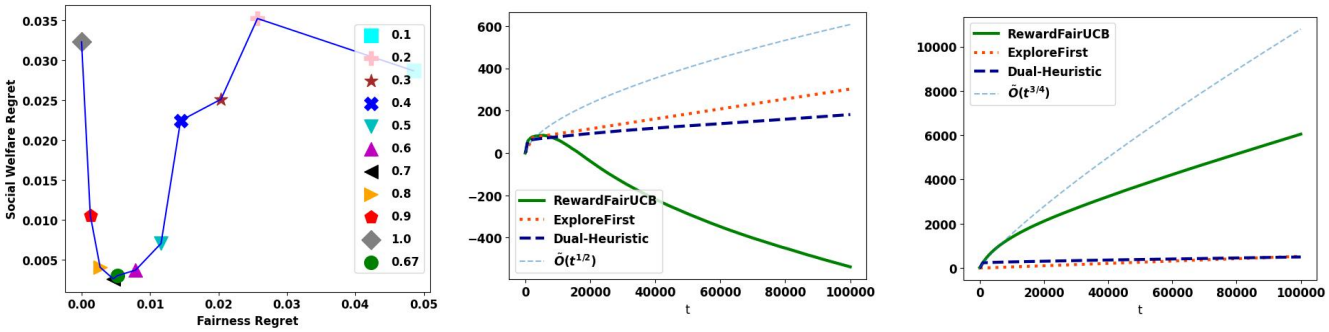
Figure 1 shows results with a mean reward matrix A of size (4, 3), i.e. $n = 4$, $m = 3$. The stopping time $T = 10^5$ and $C_i = c = 0.3 \; \forall i \in [n]$. We first empirically show the trade-off between exploration and exploitation by plotting the social welfare regret and fairness regret of Explore-First algorithm on varying the exploration parameter α. Figure 1a shows the plots comparing the social welfare regret and the fairness regret on varying the exploration parameter α from $\{0.1, 0.2, \cdots, 1.0, 0.67\}$, marked in the legend. The plotted regret values are after normalizing by T. The empirically observed best choice for obtaining low regrets for both social welfare and fairness is $\alpha = 0.67$ which closely matches the theoretically optimal value of $\alpha = 2/3$ derived for the 2-arm case in Sec 3.

Figures 1b and 1c respectively compare the social welfare regrets and the fairness regrets of RewardFairUCB with the Explore-First baseline (with $\alpha = 0.67$) and the dual heuristic (Sec 6). Figure 1b shows that RewardFairUCB not only obtains a sub-linear regret but also outperforms the baselines and heuristics. We can also see sublinear regrets obtained by Explore-First, supporting our theoretical claim derived for the 2-arm case. Figure 1c demonstrates sublinear fairness regret of RewardFairUCB. While the Explore-First baseline and the dual-based heuristic obtain a lower fairness regret, they incur an excess social welfare regret. RewardFairUCB achieves optimal social welfare performance while maintaining a sublinear fairness regret.

7.3 Experiments on Real-World Data

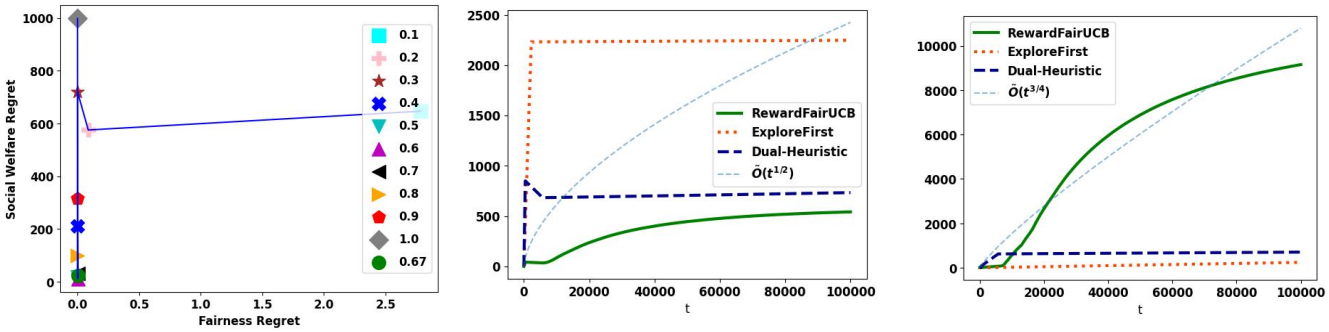
Figure 2 shows the performance of our algorithm on real-world data, MovieLens 1M [8]. MovieLens comprises ratings given by users to different movies. We obtain a user-genre matrix with the average rating that users assign to each movie genre. This matrix is normalized to have each entry in $[0, 1]$ and serves as the mean reward matrix A. For the movies associated with multiple genres, their contribution to each genre was divided equally.

The A matrix for this experiment is of size (6039, 18), i.e. $n = 6039$, $m = 18$. The stopping time $T = 10^5$ and $C_i = c = 1/m \; \forall i \in [n]$. We first empirically show the trade-off between social welfare regret



(a) Explore-Exploit tradeoff with Explore-First. **(b) Social welfare regrets vs timesteps.** **(c) Fairness regrets vs timesteps.**

Figure 1: Experimental results on simulated data ($n = 4, m = 3$). C_i is $0.3\ \forall i \in [n]$.



(a) Explore-Exploit tradeoff with Explore-First. **(b) Social welfare regrets vs timesteps.** **(c) Fairness regrets vs timesteps.**

Figure 2: Experimental results on MovieLens real-world data ($n = 6039, m = 18$). C_i is $1/m\ \forall i \in [n]$.

and fairness regret of Explore-First algorithm on varying the exploration parameter α.

We begin by empirically illustrating the effect of exploration and exploitation trade-off, controlled by the exploration parameter α, on the social welfare regret and fairness regret of the Explore-First. Figure 2a shows the two regrets (normalized by T) with α values ranging from $\{0.1, 0.2, \cdots, 1.0, 0.67\}$ as marked in the legend. The empirically determined optimal that minimizes both social welfare regret and the fairness regret is $\alpha = 0.67$ which closely matches the theoretically optimal $\alpha = 2/3$ derived for the 2-arm case in Sec 3. Figures 2b and 2c compare the social welfare and fairness regrets of RewardFairUCB with the baseline algorithm Explore-First and the dual-heuristic. Figures 2b and 2c empirically demonstrate that RewardFairUCB obtains a sublinear regret for both social welfare and fairness. Although the Explore-First baseline and the dual-based heuristic achieve lower fairness regret, they lead to a higher social welfare regret. RewardFairUCB performs optimally in terms of social welfare and still obtains a sublinear fairness regret. The empirical results with Explore-First also support our theoretically sublinear regret claim that was derived for the 2-arm case.

8 DISCUSSION AND FUTURE WORK

Our paper formulates a fair MA-MAB problem where the search is over reward-based social welfare maximizing policy that also ensures fairness to each agent. Our notion of fairness guarantees a pre-specified fraction of the corresponding maximum possible rewards to each agent. We derive the lower bound of $\tilde{O}(\sqrt{T})$ that holds individually for both social welfare and fairness regret. Our proposed algorithm RewardFairUCB obtains an optimal (up to logarithmic constants) social welfare regret and a sub-linear fairness regret. We also propose baseline algorithms/heuristics for the problem, present the exploration-exploitation trade-off and empirically validate the efficacy of the proposed RewardFairUCB algorithm on both simulated and real-world data. Our algorithms can be easily made time-horizon unaware with a doubling trick [4, Theorem 4]. Improving the fairness regret upper bound of $\tilde{O}(T^{3/4})$ to match the lower bound of $\Omega(\sqrt{T})$ would be an interesting future work which would also include theoretically analysing the proposed dual-based heuristics. Another future work could be to extend the lower bounds for the regrets derived individually to hold simultaneously. It will also be interesting to extend our theoretical analysis for Explore-First algorithm (Algo 1) for $m > 2$.

ACKNOWLEDGMENTS

PM thanks Google for the PhD Fellowship. GG thanks support from SERB through grant CRG/2022/007927. The authors also thank the anonymous reviewers for their constructive feedback.

REFERENCES

[1] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. 2002. The Nonstochastic Multiarmed Bandit Problem. *Society for Industrial and Applied Mathematics (SIAM) Journal on Computing* 32, 1 (2002), 48–77.

[2] Siddharth Barman, Arindam Khan, Arnab Maiti, and Ayush Sawarni. 2023. Fairness and welfare quantification for regret in multi-armed bandits. In *Association for the Advancement of Artificial Intelligence (AAAI) (AAAI'23)*. AAAI Press, 6762–6769.

[3] Dorian Baudry, Nadav Merlis, Mathieu Benjamin Molina, Hugo Richard, and Vianney Perchet. 2024. Multi-armed bandits with guaranteed revenue per arm. In *AISTATS (Proceedings of Machine Learning Research, Vol. 238)*. PMLR, 379–387.

[4] Lilian Besson and Emilie Kaufmann. 2018. What Doubling Tricks Can and Can't Do for Multi-Armed Bandits. *arXiv preprint arXiv:1803.06971* (2018).

[5] Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. 2012. *Regret analysis of stochastic and nonstochastic multi-armed bandit problems*. Vol. 5. Now Publishers, Inc. 1–122 pages.

[6] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D Procaccia, Nisarg Shah, and Junxing Wang. 2019. The unreasonable fairness of maximum Nash welfare. *ACM Transactions on Economics and Computation (TEAC)* 7, 3 (2019), 1–32.

[7] Steven Diamond and Stephen Boyd. 2016. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research* 17, 83 (2016), 1–5.

[8] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems* 5, 4, Article 19 (Dec 2015), 19 pages.

[9] Safwan Hossain, Evi Micha, and Nisarg Shah. 2021. Fair Algorithms for Multi-Agent Multi-Armed Bandits. In *Neural Information Processing Systems (NeurIPS)*, Vol. 34. Curran Associates, Inc., 24005–24017.

[10] Matthew Jones, Huy Nguyen, and Thy Nguyen. 2023. An efficient algorithm for fair multi-agent multi-armed bandit with low regret. In *Association for the Advancement of Artificial Intelligence (AAAI) (AAAI'23)*. AAAI Press, Article 916, 9 pages.

[11] Mamoru Kaneko and Kenjiro Nakamura. 1979. The Nash Social Welfare Function. *Econometrica* 47, 2 (1979), 423–435.

[12] Anand Krishna, Philips George John, Adarsh Barik, and Vincent Y. F. Tan. 2024. p-Mean Regret for Stochastic Bandits. arXiv:2412.10751 [cs.LG]

[13] Tor Lattimore and Csaba Szepesvári. 2020. *Bandit Algorithms*. Cambridge University Press.

[14] Keqin Liu and Qing Zhao. 2009. Distributed Learning in Multi-Armed Bandit With Multiple Players. *IEEE Transactions on Signal Processing* 58 (2009), 5667–5681.

[15] Yang Liu, Goran Radanovic, Christos Dimitrakakis, Debmalya Mandal, and David C Parkes. 2017. Calibrated fairness in bandits. *Proceedings of the 4th Workshop on Fairness, Accountability, and Transparency in Machine Learning (Fat/ML 2017)* (2017).

[16] Vishakha Patil, Ganesh Ghalme, Vineet Nair, and Y. Narahari. 2021. Achieving Fairness in the Stochastic Multi-Armed Bandit Problem. *Journal of Machine Learning Research (JMLR)* 22, 174 (2021), 1–31.

[17] Vishakha Patil, Vineet Nair, Ganesh Ghalme, and Arindam Khan. 2023. Mitigating Disparity while Maximizing Reward: Tight Anytime Guarantee for Improving Bandits. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*. International Joint Conferences on Artificial Intelligence Organization, 4100–4108.

[18] Tom Ron, Omer Ben-Porat, and Uri Shalit. 2021. Corporate Social Responsibility via Multi-Armed Bandits. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT '21)* (Virtual Event, Canada). Association for Computing Machinery, 26–40.

[19] Abhishek Sinha. 2024. BanditQ: Fair Bandits with Guaranteed Rewards. In *Proceedings of the Fortieth Conference on Uncertainty in Artificial Intelligence (UAI) (Proceedings of Machine Learning Research, Vol. 244)*. PMLR, 3227–3244.

[20] Aleksandrs Slivkins. 2019. Introduction to Multi-Armed Bandits. *Foundations and Trends® in Machine Learning* 12, 1-2 (2019), 1–286.

[21] Lequn Wang, Yiwei Bai, Wen Sun, and Thorsten Joachims. 2021. Fairness of exposure in stochastic bandits. In *International Conference on Machine Learning (ICML)*. PMLR, 10686–10696.

[22] Mengxiao Zhang, Ramiro Deo-Campo Vuong, and Haipeng Luo. 2024. No-Regret Learning for Fair Multi-Agent Social Welfare Optimization. In *Neural Information Processing Systems (NeurIPS)*, Vol. 37. Curran Associates, Inc.