One for All and All for One: Distributed Learning of Fair Allocations with Multi-player Bandits

Ilai Bistritz\textsuperscript{1}, Tavor Z. Baharav\textsuperscript{1}, Amir Leshem\textsuperscript{2}, Nicholas Bambos\textsuperscript{1}
\textsuperscript{1}Department of Electrical Engineering, Stanford University
\textsuperscript{2}Faculty of Engineering, Bar-Ilan University
\{bistritz, baharav, bambos\}@stanford.edu, leshema@biu.ac.il

Abstract—Consider \( N \) cooperative but non-communicating players where each plays one out of \( M \) arms for \( T \) turns. Players have different utilities for each arm, represented as an \( N \times M \) matrix. These utilities are unknown to the players. In each turn, players select an arm and receive a noisy observation of their utility for it. However, if any other players selected the same arm in that turn, all colliding players will receive zero utility due to the conflict. No communication between the players is possible. We propose two distributed algorithms which learn fair matchings between players and arms while minimizing the regret. We show that our first algorithm learns a max-min fairness matching with near-\( O(\log T) \) regret (up to a \( \log \log T \) factor). However, if one has a known target Quality of Service (QoS) (which may vary between players) then we show that our second algorithm learns a matching where all players obtain an expected reward of at least their QoS with constant regret, given that such a matching exists. In particular, if the max-min value is known, a max-min fairness matching can be learned with \( O(1) \) regret.

I. INTRODUCTION

Resource allocation is typically accomplished via a centralized protocol that dictates how the agents in a system share its resources. As systems grow larger, resource allocation becomes an increasingly challenging task, which precludes a centralized entity from aggregating the parameters needed to solve the allocation problem and disseminating the solution back to the agents. Aggregating parameters is even less practical when the agents themselves do not know their local parameters a priori and have to learn them on the fly. Hence, scaling up resource allocation calls for distributed protocols where the cooperative agents make local decisions on which resources to use. These distributed protocols must be designed to allow the agents to learn their environment while accounting for the fact that one agent’s actions affect all the others.

The emerging field of multi-player bandits\textsuperscript{[1]}–\textsuperscript{[15]} provides a framework for distributed learning for resource allocation problems. In multi-player multi-armed bandits we think of each player as one of the agents, and the arms as the system resources. In the standard model, each player faces choices of all players. While the players are fully cooperative since they follow the designed protocol, conflicts still arise due to the players’ preferences over the limited resources. To model this, a reward of zero is assigned to players that select the same arm. Therefore, by learning which arms to pull, the agents learn together an optimal resource allocation in a distributed and online manner. The only information a player receives regarding her peers is through the collisions that occur when one or more players select the same arm as her. This collision model captures channels in communication networks, computation resources on servers, consumers and indivisible items, etc.

A common network performance objective is the sum of rewards of the players. As such, works on multi-player bandits have primarily focused on maximizing the sum of rewards\textsuperscript{[17]}–\textsuperscript{[25]}. However, in the broader literature of network optimization, the sum of rewards is only one possible objective. One very significant drawback of this objective is that it has no fairness guarantees. As such, the maximal sum of rewards assignment might starve some users. In many applications, the designer wants to make sure that all users will enjoy at least minimal target QoS.

In the context of maximizing the sum of rewards, many works on multi-player bandits have considered a model where all players have the same vector of expected rewards\textsuperscript{[24]}, \textsuperscript{[26]}–\textsuperscript{[28]}. While relevant in some applications this model is not rich enough to study fairness, as then the worst off player is simply the one that was allocated the worst resource. To study fairness a heterogeneous model is necessary, where players have different expected rewards for the arms, representable as a matrix. In this case, a fair allocation may prevent some players from getting their best arm in order to significantly improve the allocation for less fortunate players.

Despite being a widely-applied objective in the broader resource allocation literature\textsuperscript{[29]}–\textsuperscript{[31]}, fairness in multi-player bandits has only recently been studied\textsuperscript{[32]}. Some multi-player bandit works have studied alternative objectives that can potentially exhibit some level of fairness\textsuperscript{[33]},\textsuperscript{[34]}. Several recent works have studied fairness constrained sequential learning for a single player\textsuperscript{[35]}–\textsuperscript{[37]}. In this paper, we consider two notions of fairness for multi-player bandits with heterogeneous arms and no communication between the players.

The first fairness notion addressed here is the well-known max-min fairness, where the goal is to maximize the expected reward of the worst off player. We provide a distributed
algorithm that has provably order optimal regret, with respect to the max-min value, up to a \( \log \log T \) factor (that can be arbitrarily improved to any factor that increases with the horizon \( T \)).

The second fairness notion we consider is ensuring that all players receive at least a target minimal QoS, which can vary between players. This goal is more common in practical applications and services. For example, cellular operators typically want to provide each user reliable communication with the advertised throughput. With compute services, each task requires some compute power and memory specifications. Remarkably, for the target QoS case, we provide a distributed algorithm that achieves constant QoS regret.

The two fairness notions coincide if the max-min value is known, and is set as the target QoS. Given this input, our QoS algorithm learns such an allocation with \( O(1) \) regret instead of near-\( O(\log T) \) which our max-min fairness algorithm will accrue (which is near-optimal if the max-min value is unknown). The significant gap between the expected regret of the two cases shows how much more complicated the problem of distributed learning of the maximal QoS is and quantifies the price that one pays for not knowing the maximal QoS ahead of time. Since the QoS algorithm can work with heterogeneous QoS between players, it can learn other Pareto optimal matchings aside from just max-min ones, when their QoS vector is known.

### A. Previous Work

A celebrated fairness objective is \( \alpha \)-fairness, which for a vector of rewards \( \mathbf{x} \in \mathbb{R}^N \) is evaluated as \( \sum_{i=1}^{N} (1 - \alpha)^{-1} x_i^{1-\alpha} \), or \( \sum_{i=1}^{N} \log x_i \) for \( \alpha = 1 \) [38]. This notion of fairness encompasses several classical ones, where \( \alpha = 0 \) yields sum of rewards, \( \alpha = 1 \) yields proportional fairness [39], and max-min fairness corresponds to the limit as \( \alpha \to \infty \). While for constant \( \alpha \), \( \alpha \)-fairness can be maximized in a similar manner to [20], [21], the case of max-min fairness is fundamentally different. The QoS objective is even further removed and is not covered by any \( \alpha \)-fairness notion.

Learning to play the max-min fairness allocation or an allocation that guarantees some QoS for all players involves major technical challenges that do not arise in the case of maximizing the sum of rewards (or in the case of \( \alpha \)-fairness). The sum-rewards optimal allocation is unique for “almost all” scenarios (randomizing the expected rewards). In contrast, there are typically multiple max-min fairness or QoS fairness allocations. This complicates the distributed learning process since players will have to agree on a specific optimal allocation to play, which is difficult to do without communication. Specifically, this rules out using similar techniques to those used in [20] to solve the sum of rewards case.

Our QoS formulation can be thought of as the multiplayer extension of the scenario in [40] or that in [41] which finds all the “good arms” (instead of minimizing the regret). However, neither [40] nor [41] can be used for the multiplayer case since they rely on i.i.d. rewards, which is no longer the case with collisions between players. Intriguingly, [40] shows that if a number \( \gamma \) between the optimal expected reward and the second best expected reward is known, then \( O(1) \) regret can be achieved for the single player multi-armed bandit problem. In this sense, our QoS algorithm generalizes [40] to multi-player bandits.

A preliminary conference version of this paper was published in [32]. This extended version further solves the QoS setting where we achieve \( O(1) \) regret, which was not part of [32]. The QoS objective can be viewed as a generalization of the max-min fairness in [32] in two ways - it allows for more than just the max-min value to be used as the target QoS, and it allows for heterogeneous QoS values between players. However, in the QoS scenario the desired QoS values need to be known (and feasible), so while the objectives are related, the two scenarios are fundamentally different. Achieving constant regret for the QoS case requires a novel algorithm (Algorithm 4) and analysis (Section VII). A constant regret algorithm can only spend a negligible amount of time exploring and finding a matching, which leads to the technical challenge of an exploitation phase with a random unbounded length. When possible, we draw connections between the two distributed fairness algorithms that both utilize our exploration and matching phases (although slightly differently). We also generalize our results to the setting with unbounded sub-Gaussian rewards, where knowledge of the (different) sub-Gaussian parameters is not needed.

### B. Outline

In Section II we formulate our multi-player bandit scenario, with the collision model and with no communication between players. We then define our two notions of fairness regret: max-min fairness and QoS. In Section III we present our two distributed fairness algorithms and state our regret bounds that are later proved in Theorems 1 and 2. We discuss the similarities and dissimilarities of these two algorithms. Section IV analyzes the exploration phase that is common to both of our algorithms. Section V analyzes the matching phase that is also shared by the two algorithms. Section VI provides the analysis for the max-min fairness case and concludes with proving Theorem 1. Section VII provides the analysis for the QoS case, and concludes with proving Theorem 2. Section VIII presents simulation results that corroborate our theoretical findings and demonstrate that both of our proposed algorithms learn their target matching faster than our analytical bounds suggest. Finally, Section IX concludes the paper.

### II. Problem Formulation

We consider a stochastic game played by a set of \( N \) players \( \mathcal{N} = \{1,...,N\} \) over a finite time horizon \( T \). The strategy space of each player is a common set of \( M \) arms with indices denoted by \( i,j \in \{1,...,M\} \). We assume that \( M \geq N \), since otherwise at least one player in each round will receive a reward of zero, which yields linear regret in \( T \) for either of our fairness notions. The horizon \( T \) is not known by any of the players and is considered to be much larger than \( M \) and \( N \) since we assume that the game is played for a long time, and hence we focus on the behavior of the regret as a function of \( T \). Let \( t \) be the discrete turn index. At each turn \( t \), all
players simultaneously pick one arm each. The arm that player $n$ chooses at turn $t$ is $a_n(t)$ and the strategy profile (vector of arms selected) at turn $t$ is $a(t)$. Players do not know which arms the other players chose, and need not even know the number of players $N$.

Define the no-collision indicator of arm $i$ in strategy profile $a$ to be

$$
\eta_i(a) = \begin{cases} 
0 & |\mathcal{N}_i(a)| > 1 \\
1 & \text{otherwise.}
\end{cases}
$$

where $\mathcal{N}_i(a) = \{n | a_n = i\}$ is the set of players that chose arm $i$ in strategy profile $a$. The instantaneous utility of player $n$ at time $t$ with strategy profile $a(t)$ is

$$
\nu_{n}(a(t)) = r_{n,a_n(t)}(t) \eta_{a_n(t)}(a(t))
$$

(2)

where $r_{n,a_n}(t)$ is a random reward. We assume that the sequence of rewards of arm $i$ for player $n$, $\{r_{n,i}(t)\}_{t=1}^T$, is i.i.d. with a continuous distribution with expectation $\mu_{n,i}$ that is sub-Gaussian with an unknown parameter $\sigma_{n,i}$, meaning

$$
\mathbb{E}\{e^{\lambda(r_{n,i} - \mu_{n,i})}\} \leq e^{\frac{\lambda^2 \sigma_{n,i}^2}{2}} \text{ for all } \lambda \in \mathbb{R}.
$$

We define $\sigma = \max_{n,i} \sigma_{n,i}$. Many common distributions are sub-Gaussian, including all bounded distributions, and of course the Gaussian distribution. This allows us to model many scenarios, from random Gaussian estimation error for the expected rewards to Bernoulli rewards with expectation $\mu_{n,i}$ that model the success of using the resource (see Remark 1).

An immediate motivation for the collision model above is channel allocation in wireless networks, where the transmission of one user creates interference for other users on the same channel and causes their transmission to fail. Since coordinating a large number of devices in a centralized manner is infeasible, distributed channel allocation algorithms are desirable in practice. In this context, our distributed algorithms learn over time how to assign the channels (arms) such that a target (or maximal) QoS guarantee is maintained for all users. Nevertheless, the collision model is relevant to many other resource allocation scenarios where the resources are discrete items that cannot be shared.

**Remark 1** (Continuously distributed rewards). Rewards with a continuous distribution are natural in many applications (e.g. signal-to-noise ratio in wireless networks). However, this assumption is only used to argue that since the probability for zero reward in a non-collision is zero, players can properly estimate their expected rewards. In the case where there is a positive probability of receiving zero reward, we can assume instead that each player can observe their no-collision indicator in addition to their reward. This alternative assumption requires no modifications to our algorithms or analyses. Observing one bit of feedback signifying whether any other player chose the same arm is significantly less than other common feedback models, such as observing the actions of other players. In wireless networks, this could mean that the ACK is not received at the transmitter over the reverse control channel and therefore the transmitter knows there was a collision on its chosen channel.

Next we define the total expected max-min fairness regret. This is the expected regret that the cooperating but non-communicating players accumulate over time from not playing the optimal max-min allocation.

**Definition 1.** The expected max-min fairness regret is defined as

$$
R_{\gamma^*}(T) = \sum_{t=1}^{T} \left( \gamma^* - \min_n \mathbb{E}\{\nu_{n}(a(t))\} \right)
$$

(3)

where $\gamma^* = \max_n \mathbb{E}\{\nu_{n}(a)\}$. The expectation is over the random rewards $\{r_{n,i}(t)\}$ that dictate the random actions $\{a_{n}(t)\}_t$.

Note that replacing the minimum in (3) with a sum over the players yields the regret for the sum of rewards objective, after redefining $\gamma^*$ to be the optimal sum of rewards [19], [20], [22]–[24].

In practice, it is more common to want user $n$ to achieve at least a target QoS $\gamma(n) > 0$ for all $n$ than to maximize the minimal expected reward. The leading example is in wireless networks, where we want a self-organizing network that can guarantee, for example, a throughput of 1Mbps to all users to support their functionality (and perhaps 10Mbps to a specific type of users). Another example is allocating each user the required compute resources for the task they are running. For this case, we define the QoS regret as follows:

**Definition 2.** Let $\gamma = (\gamma^{(1)}, \ldots, \gamma^{(N)})$. The total expected regret for achieving QoS of at least $\gamma(n)$ for all $n$, which we call QoS regret, is defined as

$$
R_{\gamma}(T) = \sum_{t=1}^{T} \max_n \left( \gamma(n) - \mathbb{E}\{\nu_{n}(a(t))\} \right)^+
$$

(4)

where $(x)^+ = \max(x, 0)$. The expectation is over the random rewards $\{r_{n,i}(t)\}_t$ that dictate the random actions $\{a_{n}(t)\}_t$.

For $\gamma(n) = \gamma^* = \max_n \mathbb{E}\{\nu_{n}(a)\}$ for all $n$, the QoS regret coincides with the max-min fairness regret.

### III. Two Distributed Fairness Algorithms

In this section, we describe our two distributed multiplayer bandit algorithms for the two fairness objectives. Both algorithms divide the unknown horizon of $T$ turns into consecutive epochs. Each epoch is further divided into three phases: exploration, matching, and exploitation.

The exploration and matching phases (Algorithms 1 and 2 respectively) are common to both algorithms, and occur in every epoch. The phases’ lengths increase as a function of the epoch index $k$. Their combined goal is to distributively identify (exploration) and converge to (matching) a $\gamma$-matching, which is a matching of players to arms such that the expected reward of each player $n$ is at least $\gamma(n)$, as defined next:

**Definition 3.** Let $\gamma = (\gamma^{(1)}, \ldots, \gamma^{(N)})$. An allocation of arms $a$ is a $\gamma$-matching if and only if $\mathbb{E}\{\nu_{n}(a)\} \geq \gamma(n)$ for all $n$.

The exploration phase of the $k$-th epoch lasts $[c_1 \log(k+1)]$ turns for some constant $c_1 \geq 4$ and is used for estimating the expectations of the arms. Note that the exploration phase, detailed in Algorithm 1, updates $\{V_{n,i}, S_{n,i}\}$ externally as
“global variables”. The matching phase of the $k$-th epoch lasts $[c_2 \log(k + 1)] + M$ turns for some constant $c_2 \geq 1$. In this phase, players attempt to converge to a $\gamma$-matching, up to the confidence intervals of the exploration phase. To find such a matching, the players follow distributed dynamics that induce an absorbing Markov chain with the strategy profiles as states. The absorbing states of this chain are the desired matchings. When the matching phase grows long enough, the probability that a matching exists but is not found is small. If a matching does not exist, the matching phase naturally does not converge. At the end of the matching phase, players use collisions to signal to other players whether the matching phase ended with a matching. Every player who did not end the matching phase with a collision plays their matched arm for $M$ turns, while players that ended the matching phase with a collision sequentially play each of the $M$ arms for the next $M$ turns. If players deduce that they collectively converged to a matching, they note this for future reference in the matching indicator $s_k$.

**Algorithm 1  Explore**

1: **Input:** Number of turns $T_{\text{explore}}$
2: for $T_{\text{explore}}$ turns do
3: Play arm $i$ uniformly at random from all $M$ arms
4: Receive $r_{n,i}(t)$ and $y_{i}(a(t))$
5: if $y_{i}(a(t)) = 1$ then ▷ No collision
6: Update $V_{n,i} \leftarrow V_{n,i} + 1$ and $S_{n,i} \leftarrow S_{n,i} + r_{n,i}(t)$
7: end if
8: end for
9: Estimate $\mu_{n,i}$ as $\hat{\mu}_{n,i} \leftarrow \frac{S_{n,i}}{V_{n,i}}$, for each $i = 1, ..., M$
10: Construct confidence intervals as $C_{n,i} \leftarrow \frac{1}{\log \frac{4}{\delta} V_{n,i}}$, with $C_{n,i} = +\infty$ for $V_{n,i} = 0, 1$
11: return $\{\hat{\mu}_{n,i}, C_{n,i}\}_{i=1}^M$

**A. The My Fair Bandit Algorithm**

In this subsection, we present the My Fair Bandit Algorithm which learns a max-min fairness allocation and is detailed in Algorithm 3. The key idea behind this algorithm is a global search parameter $\gamma_k$ that all players track together (with no communication required). Since we are interested in maximizing the minimal reward received by any player, we refer to a $\gamma_k$-1-matching as a $\gamma_k$-matching in the context of the My Fair Bandit Algorithm. Essentially, the players want to find the maximal $\gamma_k$ for which there still exists a $\gamma_k$-matching. However, players do not know their expected rewards, and their coordination is extremely limited.

**Algorithm 2  Match**

1: **Input:** Number of turns $T_{\text{match}}$, QoS $\gamma(n)$, arm information $\{\hat{\mu}_{n,i}, C_{n,i}\}_{i=1}^M$
2: $E_n \leftarrow \{i | \hat{\mu}_{n,i} \geq \gamma(n) - C_{n,i}\}$
3: Pick $a_n(t)$ uniformly at random from $E_n$
4: for $T_{\text{match}} - 1$ turns do ▷ Play absorbing Markov chain
5: if $y_{a_n(t)}(a(t)) = 1$ then ▷ No collision
6: Keep playing the same arm, $a_n(t + 1) \leftarrow a_n(t)$
7: else ▷ Collision
8: Pick $a_n(t + 1)$ uniformly at random from $E_n$
9: end if
10: end for
11: Set $\hat{a}_n \leftarrow a_n(t)$
12: if $y_{a_n}(\bar{a}) = 1$ then ▷ Reach consensus
13: Play $a_n$ for the next $M$ turns
14: else
15: Play $a_n = 1, ..., M$ sequentially
16: end if
17: If you experienced a collision in the last $M$ turns then set $s \leftarrow 0$, else set $s \leftarrow 1$
18: return $\hat{a}_n, s$

The $k$-th exploitation phase of the My Fair Bandit Algorithm has a deterministic length of $\lfloor c_3 (\frac{4}{\delta})^k \rfloor$ turns for some constant $c_3 \geq 1$. At the beginning of this phase, if the last matching phase succeeded then the search parameter is updated as $\gamma_{k+1} = \gamma_k + \varepsilon_k$. The step size $\varepsilon_k$ is decreasing such that if even a slightly better matching exists, it will eventually be found. However, it might be that no $\gamma_k$-matching exists. Hence, once in a while with decreasing frequency, the players reset $\gamma_{k+1} = 0$ in order to allow themselves to keep finding new matchings. During all turns of the exploitation phase, players play the best recently found matching $\hat{a}_{k^*}$, where $k^*$ is the epoch $k$ within the last $\frac{k}{2}$ epochs with the largest $\gamma_k$ that resulted in a matching.

The epoch structure is designed to address the primary challenge arising from having multiple players: coordinating between players without communication. To this end, the players in our algorithm try together to find a $\gamma$-matching, where $\gamma$ is a mutual parameter that they can all update simultaneously but independently, obviating the need for a central entity. Then the main measure of multi-player “problem hardness” is the absorption time $\hat{\tau}$ of the matching Markov chain (see Lemma 4), which is unknown. To find a matching we then need a matching phase with increasing length (to eventually surpass $\hat{\tau}$), taken to be of length $[c_2 \log(k + 1)] + M$ for the $k$-th
phase, which alone contributes $O((M + \log \log T) \log T)$ to the expected max-min fairness regret. Hence, the coordination challenge dominates the regret of our algorithm.

**Algorithm 3 My Fair Bandit Algorithm**

1. **Initialization:** Set $V_{n,i} = 0$, $S_{n,i} = 0$ for all $i$. Set reset counter $w = 0$ with expiration $e_w = 1$. Set $\varepsilon_0 = 1$.
2. for epoch $k = 1, 2, \ldots$ do
3. $\{\mu_{n,i}, C_{n,i}\}_{i=1}^M \leftarrow \text{Explore}([c_1 \log(k+1)])$
4. Set $S_{n,i} \leftarrow S_{n,i}$ and $V_{n,i} \leftarrow V_{n,i}$ for each $i \in [M]$.
5. Update $w \leftarrow w + 1$
6. if $w = e_w$ then $\triangleright$ Initiate a new search
7. Set $\varepsilon_k \leftarrow 0$, $w \leftarrow 0$, $e_w \leftarrow \left\lfloor \frac{1}{\log k} \right\rfloor$, $\varepsilon_k \leftarrow \max \frac{1}{1+\log k}$.
8. else
9. $\varepsilon_k \leftarrow \varepsilon_{k-1}$.
10. end if
11. $\tilde{a}_{k,n}, s_k \leftarrow \text{Match([c_2 \log(k+1)], \gamma_k, \{\mu_{n,i}, C_{n,i}\})}$
12. if $s_k = 1$ then $\triangleright$ Success: increase $\gamma_k$
13. $\gamma_{k+1} \leftarrow \gamma_k + \varepsilon_k$
14. else
15. $\gamma_{k+1} \leftarrow \gamma_k$
16. end if
17. for $\left\lfloor c_3 \left(\frac{1}{\log k}\right)^{0.3} \right\rfloor$ turns do $\triangleright$ Exploitation Phase
18. Play $\tilde{a}_{k,n}$ for the maximal $k^* \in \arg \max \gamma_k s_k$
19. end for
20. end for

Our main result for the My Fair Bandit Algorithm is summarized in Table I. The additive constant $C_0$ (see (26)) is essentially the regret accumulated during the initial epochs when the confidence intervals were still not small enough compared to the gap $\Delta = \min_{n \neq j} \min \{\mu_{n,i} - \mu_{n,j}\}$ (formalized in (12)) or when the length of the matching phase was not long enough compared to $\tau$ (defined in (13), and the requirement is formalized in (17)). Hence, $C_0$ depends on $\Delta$ and $\tau$ but not on $T$.

**B. QoS Algorithm**

In this subsection, we present the QoS Algorithm which learns a $\gamma$-matching for some target feasible QoS vector $\gamma$, and is detailed in Algorithm 4. The key idea behind the QoS algorithm is that if the designer knows that a target QoS vector $\gamma$ is feasible, then this knowledge can be leveraged to expedite the convergence to a $\gamma$-matching, accumulating significantly less regret in the process. Surprisingly, we prove that in this case the near-$O(\log T)$ regret can be improved to a constant regret of $O(1)$. This constant regret depends on $N, M$, and the expected rewards $\{\mu_{n,i}\}$, but not on the horizon $T$ which we think of as very large since the game is played for a long time. Our result for the QoS setting is summarized in Table I.

To achieve $O(1)$ regret, it is necessary to reduce the portion of time spent on exploration or matching compared to their portion in the My Fair Bandit Algorithm (Algorithm 3). To that end, our QoS Algorithm (Algorithm 4) employs an exploitation phase with a random duration, that terminates only when one of the players no longer believes that the matching being played is a $\gamma$-matching. To enable this mechanism, in this algorithm players use the reward samples from the exploitation phase in addition to their reward samples from all the past exploitation phases. Intuitively, if players play a true $\gamma$-matching during the exploitation phase, they will stop doing so only under the rare event of a failed estimation. If players play a false $\gamma$-matching during the exploitation phase, they will stop doing so only under the rare event of a failed estimation. Hence, the expected duration of a correct exploitation phase becomes long enough to suppress the time spent on exploration, matching or failed exploitations, leading to the desired $O(1)$ QoS regret.

**IV. Exploration Phase**

Over time, players receive stochastic rewards from different arms and average them to estimate their expected reward for each arm. In each epoch, only $[c_1 \log(k+1)]$ turns are dedicated to exploration. However, the estimation of the expected rewards uses all the previous exploration phases, so the number of samples used for estimation at epoch $k$ is $\Theta(k \log k)$. Since players only have estimates of the expected rewards, they can never be sure if a matching is a true $\gamma$-matching. The purpose

**Algorithm 4 QoS Algorithm**

1. **Input:** Target QoS $\gamma(n)$.
2. **Initialization:** $V_{n,i} \leftarrow 0$ and $S_{n,i} \leftarrow 0$ for all $i$.
3. for epoch $k = 1, 2, \ldots$ do
4. $\{\mu_{n,i}, C_{n,i}\}_{i=1}^M \leftarrow \text{Explore([c_1 \log(k+1)])}$
5. Set $S_{n,i} \leftarrow S_{n,i}$ and $V_{n,i} \leftarrow V_{n,i}$ for each $i = 1, \ldots, M$.
6. $\tilde{a}_{k,n}, s_k \leftarrow \text{Match([c_2 \log(k+1)], \gamma(n), \{\mu_{n,i}, C_{n,i}\})}$
7. Initialize $d \leftarrow \text{run}$ and $\tau \leftarrow 0$
8. Set $\tilde{a}_{k,n}$ for the maximal $k^* \in \arg \max s_k$
9. Set $\tilde{S}_n \leftarrow \sum_{n,i} \tilde{S}_{n,i} - S_{n,i}^{[k/2]}$, $\tilde{V}_n \leftarrow \tilde{V}_{n,i} - V_{n,i}^{[k/2]}$
10. while $d = \text{run}$ do
11. for $c_e = 1, \ldots, M$ do $\triangleright$ Subepoch of $M$ turns
12. Play $\tilde{a}_{k,n}$, receive $r_{n,i} \tilde{a}_{k,n}(t)$ and $\eta_{n,i} r_{n,i} \hat{\alpha}(t)$$\triangleright$ No-collision
13. Update $\tau \leftarrow \tau + 1$
14. if $\eta_{n,i} \hat{\alpha}(t)$ then $\triangleright$ Signaling detected
15. Update $\tilde{V}_n \leftarrow \tilde{V}_n^{[k/2]} + r_{n,i} \eta_{n,i} \hat{\alpha}(t)$
16. end if
17. Set $d \leftarrow \text{terminate}$
18. Estimate $\mu_{n,i} \tilde{a}_{k,n}$ as $\tilde{\mu}_{n,i} \leftarrow \tilde{S}_n \tilde{V}_n$
19. Construct $\tilde{C}_{n,i} \tilde{a}_{k,n}(t) = \frac{1}{\log T} \tilde{V}_n$
20. if $\tilde{\mu}_{n,i} \tilde{a}_{k,n} \leq \gamma(n) - \tilde{C}_{n,i} \tilde{a}_{k,n}$ and $d = \text{run}$ then
21. $d \leftarrow \text{signal}$ $\triangleright$ False matching detected
22. end if
23. end if
24. if $d = \text{signal}$ then $\triangleright$ Disseminate
25. Sequentially play arms $1, \ldots, M$.
26. Exit the while loop
27. end if
28. while $\text{end}$
29. end for
30. end for
of the exploration phase is to help the players become more confident over time that the matchings they converge to in the matching phase are indeed $\gamma$-matchings.

In our exploration phase, each player picks an arm uniformly at random. This type of exploration phase commonly appears in multi-player bandit algorithms [20], [26]. However, the nature of what the players are trying to estimate is different in our setting. With a sum of rewards objective, players just need to improve over time the accuracy of the estimation of the expected rewards. With a fairness objective, player $n$ needs to make a hard (binary) decision whether a certain arm has expected reward above or below $\gamma(n)$. After the confidence intervals become small enough, if the estimations do fall within their confidence intervals, players can be confident about this hard decision. In this case, where the confidence intervals are small enough, a matching $\alpha$ is a $\gamma$-matching if all players observe that $\mu_{n,a}^k \geq \gamma(n) - C_{n,a,n}$. The next lemma bounds the probability that this success event does not occur, i.e., the $k$-th exploration failed.

**Lemma 1 (Exploration Error Probability).** Let $\left\{ \mu_{n,i}^k \right\}$ be the estimated reward expectations using all the exploration phases up to epoch $k$, with confidence intervals $\left\{ C_{n,i}^k \right\}$. Define $V_{n,i}^k$, as the number of visits with no collision of player $n$ to arm $i$ up to epoch $k$. Define the $k$-th exploration error event as

$$E_{e,k} = \left\{ \exists n,i : V_{n,i}^k \leq \frac{c_1 k \log \frac{k}{5M}}{\sigma^2} \quad \text{or} \quad \left| \mu_{n,i}^k - \mu_{n,i} \right| \geq C_{n,i}^k \right\}. \quad (5)$$

Then for all $k \geq k_0$ for a large enough constant $k_0$ we have

$$\mathbb{P}(E_{e,k}) \leq 3NMe^{-c_1k}. \quad (6)$$

**Proof.** After the $k$-th exploration phase, the estimation of the expected rewards is based on $T_e(k)$ samples, and

$$T_e(k) \geq c_1 \sum_{i=1}^{k} \log(i+1) \geq c_1 \frac{k \log k}{2}. \quad (7)$$

Let $A_{n,i}(t)$ be the indicator that is equal to one if only player $n$ chose arm $i$ at time $t$. The exploration phase consists of uniform and independent arm choices, so $\mathbb{P}(A_{n,i}(t) = 1) = \frac{1}{M} \left( 1 - \frac{1}{M} \right)^{N-1}$. Let $V_{m,n}^k = \min_{n,i} V_{n,i}^k$. We show that each player pulls each arm many times without collisions. Formally:

$$\mathbb{P}(V_{m}^k \leq \frac{T_e(k)}{5M}) = \mathbb{P}\left( \bigcup_{i=1}^{M} \bigcup_{n=1}^{N} \left\{ V_{n,i}^k \leq \frac{T_e(k)}{5M} \right\} \right) \leq NMP\left( V_{1,1}^k \leq \frac{T_e(k)}{5M} \right) \leq NMe^{-\frac{1}{5M} \left( \left(1 - \frac{1}{M} \right)^{N-1} \right)^2 T_e(k)} \leq NMe^{-\frac{1}{5M} T_e(k)} \quad (8)$$

where (a) is a union bound, (b) is Hoeffding’s inequality for Bernoulli random variables [42, Theorem 2.2.2] and (c) follows since $M \geq N$ and $\left(1 - \frac{1}{M} \right)^{M-1} - \frac{1}{5M} \geq e^{-\frac{1}{5M}} > \frac{1}{M}$, which follows since $(M-1) \log \left( 1 - \frac{1}{M} \right) \geq \frac{1}{2} > -1$. Next, by Hoeffding’s inequality:

$$\mathbb{P} \left( \bigcup_{n=1}^{N} \bigcup_{i=1}^{M} \left\{ \mu_{n,i}^k - \mu_{n,i} \geq C_{n,i}^k \right\} \right) \leq 2M \sum_{n=1}^{N} \sum_{i=1}^{M} e^{-\frac{V_{n,i}^k (C_{n,i}^k)^2}{2\sigma^2}} \leq 2NMe^{-\frac{V_{n,i}^k}{2\sigma^2 \log 2 v_{n,i}}} \leq 2NMe^{- \frac{V_{n,i}^k}{2\sigma^2 \log 2 v_{n,i}}} \quad (9)$$

where (a) is Hoeffding’s inequality for independent sub-Gaussian random variables [42, Theorem 2.6.2] with parameter $\sigma_{n,i} \leq \sigma$ (constant of 2 can be derived by Chernoff bound). Equality (b) uses $C_{n,i}^k = (\log V_{n,i})^{-1/4}$. Inequality (c) follows for $V_{n,i}^k > 1$ for which $\frac{V_{n,i}^k}{\log^2 V_{n,i}^k}$ is increasing in $V_{n,i}^k$. Hence for all $k \geq k_0$ for a sufficiently large $k_0$:

$$\mathbb{P}\left( E_{e,k} \bigg| V_{m}^k > \frac{T_e(k)}{5M} \right) \leq \mathbb{P}\left( \bigcup_{n=1}^{N} \bigcup_{i=1}^{M} \left\{ \mu_{n,i}^k - \mu_{n,i} \geq C_{n,i}^k \right\} \bigg| V_{m}^k > \frac{T_e(k)}{5M} \right) \leq \frac{2NMe^{-\frac{T_e(k)}{10M^2}}}{2\sigma^2 \log 2 v_{n,i}} \quad (10)$$

where (a) uses the law of total probability with respect to $\{V_{m,i}^k\}$ with Bayes’ rule on $\{V_{m}^k \geq \frac{T_e(k)}{5M}\}$, using the bound in (9). We conclude that for all $k \geq k_0$:

$$\mathbb{P}(E_{e,k}) = \mathbb{P}\left( E_{e,k} \bigg| V_{m}^k \leq \frac{T_e(k)}{5M} \right) \mathbb{P}\left( V_{m}^k \leq \frac{T_e(k)}{5M} \right) + \mathbb{P}\left( E_{e,k} \bigg| V_{m}^k > \frac{T_e(k)}{5M} \right) \mathbb{P}\left( V_{m}^k > \frac{T_e(k)}{5M} \right) \leq \mathbb{P}\left( V_{m}^k \leq \frac{T_e(k)}{5M} \right) + \mathbb{P}\left( E_{e,k} \bigg| V_{m}^k > \frac{T_e(k)}{5M} \right) \mathbb{P}\left( V_{m}^k > \frac{T_e(k)}{5M} \right) \leq \frac{NMe^{-\frac{T_e(k)}{10M^2}} + 2NMe^{-\frac{T_e(k)}{10M^2}}}{2\sigma^2 \log 2 v_{n,i}} \quad (11)$$

where (a) uses (8) and (10). Finally, (6) follows by using (7) in (11) for a sufficiently large $k$.

Lemma 1 implies that the confidence intervals will eventually be sufficiently small, as given by the following Corollary.

**Corollary 0.1.** For any constant $\delta > 0$, if $E_{e,k}$ did not happen and $k > k_0$ for some constant $k_0$, then $\max_{n,i} C_{n,i}^k < \frac{\delta}{4}$.

**Proof.** Given that $E_{e,k}$ did not happen, we have for all $k \geq k_0$ for a sufficiently large $k_0$:

$$\max_{n,i} C_{n,i}^k = \max_{n,i} \frac{1}{\log \frac{\log^2 v_{n,i}}{\log^2 v_{n,i}}} \leq \frac{1}{\log^2 \left( \frac{\log^2 v_{n,i}}{\log^2 v_{n,i}} \right)} < \frac{\delta}{4}. \quad (12)$$

□
V. Matching Phase

In this section, we analyze the matching phase, where the goal is to distributely find a $\gamma$-matching (up to confidence intervals) based on the estimated expected rewards from the exploration phases. During this phase, the reward values are ignored, as for player $n$ only the binary decision of whether an arm is better or worse than $\gamma^{(n)}$ matters. These binary decisions induce the following bipartite graph between the $N$ left vertices and $M$ right vertices. During this phase, the reward values are ignored, as for player $n$ only the binary decision of whether an arm is better or worse than $\gamma^{(n)}$ matters. These binary decisions induce the following bipartite graph between the $N$ left vertices and $M$ right vertices.

**Definition 4.** Let $G_k$ be the bipartite graph where edge $(n,i)$ exists if and only if $\mu^k_{n,i} \geq \gamma^{(n)} - C^k_{n,i}$.

It is worth noting that due to the confidence intervals in the definition of $G_k$, the matchings in $G_k$ are in fact only $(\gamma - \max_{n,i} C^k_{n,i})$-matchings. This is sufficient for our purposes, however, as we are interested in the good event when the confidence intervals are sufficiently small and the estimated expected rewards fall within their confidence intervals. For the My Fair Bandit Algorithm (Algorithm 3), in addition $\gamma_k$ is sufficiently close to $\gamma^*$, the matchings in $G_k$ are exactly the optimal $\gamma^*$-matchings. For the QoS Algorithm (Algorithm 4), $\gamma^{(n)} = \gamma^{(n)}$ for all $n$ and $k$, so on this good event the matchings in $G_k$ are exactly the optimal $\gamma^*$-matchings. Note that in the My Fair Bandit Algorithm $G_k$ is used in epoch $k$ while in the QoS Algorithm $G_{[k/2]}$ is used in epoch $k$.

During the matching phase, players follow our dynamics to switch arms in order to find a matching in the appropriate $G_k$, which is a $\gamma$-matching up to confidence intervals. The matchings are absorbing states since the players stop switching arms if they are collectively playing a $\gamma^*$-matching. The dynamics of the players induce the following Markov chain:

**Definition 5.** We define the process $a(t)$ starting from $a(0)$. Let $E^k_n = \{i | \mu^k_{n,i} \geq \gamma^{(n)} - C^k_{n,i}\}$. The transition from $a(t)$ to $a(t+1)$ is dictated by the transition of each player $n$:

1) If $\pi_{n,a(n)}(a(t)) = 1$ then $a_n(t+1) = a_n(t)$ with probability 1.

2) If $\pi_{n,a(n)}(a(t)) = 0$ then $a_n(t+1) = i$ with probability $\frac{1}{|E^k_n|}$ for all $i \in E^k_n$.

Next we prove that if a matching exists in $G_k$, then the matching phase will find it with a probability that goes to one when $k \rightarrow \infty$. Nevertheless, we only need this probability to exceed a large enough constant.

**Lemma 2.** Let $G_{N,M}$ be the set of all bipartite graphs with $N$ left vertices and $M$ right vertices that have a matching of size $N$. Define the random variable $\tau(G,a(0))$ as the first time the process of Definition 5, $\{a(t)\}$, constitutes a matching of size $N$, starting from $a(0)$. Define

$$\bar{\tau} = \max_{a(0),G \in G_{N,M}} \mathbb{E}\{\tau(G,a(0))\}. \tag{13}$$

If $G_k$ permits a matching then the $k$-th matching phase (or $2k$-th for Algorithm 4) converges to a matching with probability $p \geq 1 - \frac{1}{c_2 \log(k+1)}$.

**Proof.** We start by noting that the process $a(t)$ that evolves according to the dynamics in Definition 5 is a Markov chain. This follows since all transitions are a function of $a(t)$ alone, with no dependence on $a(t-1), ..., a(0)$ given $a(t)$. Let $M$ be a matching in $G_k$. Define $\Phi_M(a)$ to be the number of players that are playing in $a$ the arm they are matched to in $M$. Observe the process $\Phi_M(a(t))$. If there are no colliding players, then $a(t)$ is a matching (potentially different from $M$) and no player will ever change their chosen arm. Otherwise, for every collision, at least one of the colliding players is not playing their arm in $M$. There is a positive probability that this player will pick their arm in $M$ at random and all other players will stay with the same arm. Hence, if $a(t)$ is not a matching, then there is a positive probability that $\Phi_M(a(t+1)) = \Phi_M(a(t)) + 1$. We conclude that every non-matching $a$ has a positive probability path to a matching, making $a(t)$ an absorbing Markov chain with the matchings as the absorbing states. By Markov’s inequality

$$\mathbb{P}\left(\tau(G_k,a(0)) \geq c_2 \log(k+1)\right) \leq \mathbb{E}\left\{\tau(G_k,a(0))\right\} \leq \frac{\bar{\tau}}{c_2 \log(k+1)}. \tag{14}$$

This proof also shows that $\bar{\tau} < \infty$, since the absorption time is stochastically dominated by a geometric random variable.

VI. Max-Min Fairness Regret Analysis

In this section, we prove Theorem 1 which shows that the expected max-min fairness regret of the My Fair Bandit Algorithm is near-$\Omega(\log T)$.

According to the seminal work of Lai and Robbins, the optimal regret in the single-player case is $O(\log T)$ [43]. The next Proposition shows that $\Omega(\log T)$ is a lower bound for our multi-player bandit case, since any multi-player bandit algorithm can be used as a single-player algorithm by simulating other players. This establishes that the My Fair Bandit Algorithm is near-optimal.

**Proposition 1.** The max-min fairness regret as defined in (3) of any algorithm is $\Omega(\log T)$.

**Proof.** For $N = 1$, the result directly follows from [43]. Assume that for $N > 1$ there is a policy that results in max-min fairness regret better than $\Omega(\log T)$. Then any single player, denoted player $n$, can simulate $N-1$ other players such that all their expected rewards are larger than her maximal expected reward. Player $n$ can also generate the other players’ random rewards, that are independent of the actual rewards she receives. Player $n$ also simulates the policies for other players and even knows when a collision occurred for herself and can assign zero reward in that case. In this scenario, the expected reward of player $n$ is the minimal expected reward among the non-colliding players. This implies that $\gamma^*$ is the largest expected reward of player $n$. Hence, in every turn $t$ without a collision, the $t$-th term in (3) is equal to the $t$-th term of the single-player regret of player $n$. If there is a collision in turn $t$, then the $t$-th term in (3) is $\gamma^*$, which bounds from above the $t$-th term of the single-player regret of player $n$. Thus, the max-min fairness regret upper bounds the single-player regret of player $n$. Hence, simulating $N-1$ fictitious players is a valid single player algorithm that violates the $\Omega(\log T)$ bound, which is a contradiction. We conclude that the $\Omega(\log T)$ bound is also valid for $N > 1$. \(\square\)
To prove our regret upper bound, we bound the probability that a $γ^*$ matching is not played during the $k$-th exploitation phase. This is accomplished by noting that a $γ^*$ matching must have been found in the last $k/2$ epochs if the past $k/2$ exploration phases succeeded, the step sizes and confidence intervals are sufficiently small (which happens when $k > k_0$), and enough of the last $k/2$ matching phases succeeded so that $γ_κ$ approached $γ^*$. Treating each matching phase as a Bernoulli trial, we notice that these Bernoulli trials are dependent, as after enough successes (that increase $γ_κ$), there will no longer be a matching in $G_κ$, yielding success probability 0. The next Lemma shows that Hoeffding’s inequality for binomial random variables still applies as long as there are few enough successes, such that there is still a matching in $G_κ$.

**Lemma 3.** Let $X_1, \ldots, X_L$ be independent Bernoulli random variables, each with a success probability of at least $p$. For $x < Lp$, consider $S_x = \sum_{i=1}^L X_i \{ \sum_{j<i} X_j < x \}$. Then

$$\mathbb{P}(S_x < x) \leq e^{-2Lp(1-p)^2}. \tag{15}$$

**Proof.** If $S_x = m < x$ then $\sum_{i=1}^L X_i < x$, as otherwise the indicators in $S_x$ of the first $x$ indices $i$ where $X_i = 1$ will be active, and so $S_x \geq x$, contradicting $S_x = m < x$. Therefore

$$\mathbb{P}(S_x < x) \leq \mathbb{P} \left( \sum_{i=1}^L X_i < x \right) \leq e^{-2Lp(1-p)^2}. \tag{16}$$

Next, we provide the main Lemma used to prove Theorem 1. The proof of Lemma 4 is based on the fact that if the past $k/2$ exploration phases succeeded, and enough matching trials succeeded, then a $γ^*$-matching was found within the last $k/2$ matching phases. This $γ^*$-matching is then played during the $k$-th exploitation phase.

**Lemma 4 (My Fair Bandit: Exploitation Error Probability).** Define the $k$-th exploitation error event $E_{κ,ℓ}$ as the event where the actions $A_κ$ played in the $k$-th exploitation phase are not a $γ^*$-matching. Define the minimal gap as

$$Δ ≜ \min_{n \geq k} \min_{i \neq j} |μ_{n,i} - μ_{n,j}|. \tag{17}$$

Let $k_0$ be large enough such that for all $k \geq k_0$

$$ε_{[k/2]} < \frac{Δ}{4} \text{ and } 1 - \frac{\bar{µ}}{c_2 \log(\frac{δ}{2} + 1)} - \frac{1 + \log k}{k/6} \geq \frac{3}{\sqrt{10}}. \tag{18}$$

Then for all $k \geq k_0$ we have

$$\mathbb{P}(E_{κ,ℓ}) \leq 5NMe^{-\frac{ε_{[k/2]}}{2}} + e^{-\frac{3k}{10}}. \tag{19}$$

**Proof.** Define $E_{κ,ℓ}$ as the event where a matching existed in $G_κ$ and was not found in the $ℓ$-th matching phase. By Lemma 2, if there is a matching in $G_κ$, then the $ℓ$-th trial has success probability at least $1 - \frac{\bar{µ}}{c_2 \log(ε+1)}$.

Next we bound from below the number of trials we have between resets. Let $k_w \geq [k/2]$ be the first epoch since $[k/2]$ where a reset occurred (so $γ_{k_0} = 0$). We see that there is at least one period between resets that falls within the last $k/2$ epochs. This follows since in the worst case the algorithm resets in epoch $[k/2] - 1$. Even still, the algorithm will reset again no later than $k_w \leq [k/2] - 1 + \left\lfloor \frac{[k/2]-1}{3} \right\rfloor \leq \frac{2k}{3}$. The subsequent reset will then happen at $k_{w+1}$, where $k_{w+1} \leq \frac{2k}{3} + \left\lfloor \frac{2k}{3} \right\rfloor \leq \left\lfloor \frac{8k}{9} \right\rfloor < k$ for $k > 0$. We conclude that the past $k/2$ epochs must have contained at least one full period (from reset to reset). The length of this period must be at least $k/6$ since the earliest epoch we consider is $\frac{k}{2}$.

Recall the definition of the $ℓ$-th exploration error event $E_{κ,ℓ}$ in (5). Define the event $A_κ = \bigcap_{ε=\lfloor k/2 \rfloor}^{k-1} E_{κ,ℓ}$ for which $A_κ = \bigcup_{ε=\lfloor k/2 \rfloor}^{k-1} E_{κ,ℓ}$. We define $β_{k_0}$ as the number of successful trials needed after reset $w$ to reach $γ_κ \geq γ^* - \frac{Δ}{4}$. Note that $β_{k_0} \leq 1 + \log k$ since no more than $1 + \log k$ steps of size $ε_{k_w} = \frac{1}{1+\log k}$ are needed. Then for all $k \geq k_0$

$$\mathbb{P}(E_{κ} | A_κ) \leq \mathbb{P} \left( γ_κ < γ^* - \frac{Δ}{4} \right) \tag{20}$$

and

$$\mathbb{P} \left( \sum_{ε=\lfloor k/2 \rfloor}^{k-1} \left\{ E_{κ,ℓ} < β_{k_0} \right\} | A_κ \right) \leq e^{-\left(1 - \frac{ε_{[k/2]}}{c_2 \log(\frac{δ}{2}+1)} - \frac{1+\log k}{k_{w+1} - k_w} \right)^2 (k_{w+1} - k_w)} \tag{21}$$

where (a) follows since given $A_κ = \bigcap_{ε=\lfloor k/2 \rfloor}^{k-1} E_{κ,ℓ}$, if $γ_κ \geq γ^* - \frac{Δ}{4}$ then a $γ^*$-matching was found before the $k$-th exploitation phase and $E_{κ,ℓ}$ did not occur. This follows since at the last success at $ℓ \leq k$ we must have had for all $n$ $μ_{n,a_n} ≥ μ_{n,a_n} - C_{ε,a_n} ≥ γ_ℓ - 2C_{ε,a_n}$

$$≥ γ^* - \frac{Δ}{4} - ε_{k_w} - 2C_{ε,a_n} > γ^* - Δ \tag{22}$$

which can only happen if $μ_{n,a_n} ≥ γ^*$. Inequality (b) in (19) follows by noting that the probability that $\max_{[k/2]} ≤ ℓ ≤ k γ_ℓ < γ^* - \frac{Δ}{4}$ with a constant step size (between resets) $ε_{k_w}$ implies fewer than $\left\lfloor \frac{γ^* - \frac{Δ}{4}}{ε_{k_w}} \right\rfloor$ successful trials between $k_w$ and $k_{w+1}$. Given $A_κ$, in any trial $ℓ \in [k_w, k_{w+1}]$ such that there have been fewer than $\left\lfloor \frac{γ^* - \frac{Δ}{4}}{ε_{k_w}} \right\rfloor$ successes in $[k_w, k_{w+1}]$, at least one matching will exist in $G_κ$ (a $γ^*$-matching $A^*$), since

$$μ_{n,a_n} ≥ μ_{n,a_n} - C_{n,a_n} ≥ γ^* - C_{n,a_n} ≥ γ^* - C_{n,a_n} \tag{23}$$

where (1) holds for all $k \geq k_0$, since then $ε_{k_w} < \frac{Δ}{4}$ so $γ_ℓ ≤ γ^* - \frac{Δ}{4} + ε_{k_w} < γ^*$. Inequality (c) in (19) uses Lemma 3 with $p = 1 - \frac{ε_{[k/2]}}{c_2 \log(\frac{δ}{2}+1)}$ and (d) uses $k_{w+1} - k_w ≥ \frac{k}{6}$ and (17).

Finally, (18) is obtained by:

$$\mathbb{P}(E_{κ}) = \mathbb{P}(E_{κ} | A_κ) \mathbb{P}(A_κ) + \mathbb{P}(E_{κ} | A_κ) \mathbb{P}(A_κ) \tag{24}$$

where (a) is a union bound over $A_κ = \bigcup_{ε=\lfloor k/2 \rfloor}^{k-1} E_{κ,ℓ}$ using Lemma 1 and (19), (b) is a geometric sum and (c) uses that $c_1 \geq 1$. \qed
Equipped with Lemma 4, we can now prove our main result for this section.

**Theorem 1 (My Fair Bandit Regret Bound).** Assume that the rewards \( \{r_{n,i}(t)\}_t \) are independent in \( n \) and i.i.d. with \( t \), with a continuous (see Remark 1) \( \mu_{n,i} \)-sub-Gaussian distribution. Without loss of generality, we assume that the expectations \( \{\mu_{n,i}\} \) are in \([0,1]\). Let \( T \) be the finite deterministic horizon of the game, which is unknown to the players. Let each player play according to Algorithm 3 with \( c_1, c_2, c_3 \geq 1 \). Then the expected max-min fairness regret satisfies

\[
R_{\gamma^*} (T) \leq C_0 + \left( M + 2 (c_1 + c_2) \log \log \frac{T}{c_3} \right) \log \frac{T}{c_3} = O ((M + \log \log T) \log T)
\]

where \( C_0 \) is a constant independent of \( T \).

**Proof.** Let \( K \) be the number of epochs that start within the \( T \) turns. Since

\[
T \geq \sum_{k=1}^{K-1} \left( c_1 \log k + c_2 \log k + M + c_3 \left( \frac{4}{3} \right)^k \right)
\]

\[
\geq 3c_3 \left( \frac{4}{3} \right)^{K} - \frac{4}{3}
\]

(24)

then \( K \leq \log \frac{T}{c_3} \left( \frac{4}{3} \right) \). Let \( k_0 \) be a constant epoch index that is large enough for the bounds of Lemma 1, Lemma 4, and inequality (c) in (25) to hold. Intuitively, this is the first epoch where the matching phase duration is long enough, the step size \( \varepsilon_k \) is small enough, and the confidence intervals are sufficiently tight. Define \( E_k \) as the event where a \( \gamma^* \)-matching is not played in the \( k \)-th exploitation phase. We now bound the max-min fairness regret of epoch \( k \geq k_0 \), denoted by \( R_{\gamma^*,k} \):

\[
R_{\gamma^*,k} \leq M + (c_1 + c_2) \log (k+1) + \mathbb{P}(E_k) c_3 \left( \frac{4}{3} \right)^k + 3
\]

\[
\leq M + (c_1 + c_2) \log (k+1) + 3
\]

\[
+ \left( 5NM e^{-(\frac{2}{3}k)} + e^{-(\frac{2k}{3})} \right) c_3 \left( \frac{4}{3} \right)^k
\]

\[
\leq M + 3 + (c_1 + c_2) \log (k+1) + 6NMc_3 \beta^k
\]

\[
\leq M + 2(c_1 + c_2) \log k
\]

(25)

where (a) uses Lemma 4, (b) follows for some \( \beta < 1 \) since \( e^{-(\frac{2}{3}k)} < \frac{2}{3} \) and \( c_1 \geq 1 \) and (c) follows for \( k > k_0 \).

We conclude that, for some additive constant \( C_0 \),

\[
R_{\gamma^*} (T) = \sum_{k=1}^{K} R_{\gamma^*,k} \leq MK + 2 \sum_{k=k_0+1}^{K} (c_1 + c_2) \log k
\]

\[
+ \sum_{k=1}^{k_0} \left( (c_1 + c_2) \log (k+1) + c_3 \left( \frac{4}{3} \right)^k + 3 \right)
\]

\[
\leq C_0 + MK + 2 (c_1 + c_2) K \log K
\]

(26)

where (a) completes the \( K \)-th epoch to a full epoch and uses (25). Then, we obtain (23) by using \( K \leq \log \frac{T}{c_3} \left( \frac{T}{c_3} + \frac{4}{3} \right) \leq \log \frac{T}{c_3} \left( \frac{T}{c_3} + \frac{4}{3} \right) \), where the second inequality holds for \( T \geq 2c_3 \) and is only used to simplify (23).

The My Fair Bandit Algorithm can be tuned to have an improved expected regret of \( O((M + f (\log T)) \log T) \) for any increasing function \( f (k) \). This is achieved by replacing the lengths of the exploration and matching phases by \( [c_1 f (k)] \) and \( [c_2 f (k)] + M \) respectively. To prove this, note that there are still \( O(\log (T)) \) epochs as (24) holds regardless of the \( k \) terms (now \( f (k) \) terms) and (25) becomes \( R_{\gamma^*,k} \leq O (f (k)) \), where \( k_0 \) and by extension \( C_0 \) are functions of \( f \). Then, (26) becomes \( R_{\gamma^*} (T) \leq C_0 + MK + 2 (c_1 + c_2) K f (K) \). Similar modifications are required for Lemma 1 and Lemma 4 where \( \log k \) is only used as an increasing term, to argue that a large enough constant \( k_0 \) exists. For example, the confidence intervals would need to be redefined as \( C_{n,i} = (f (V_{n,i}))^{-1/4} \) and the result of Lemma 2 would give \( p \geq 1 - \frac{2c_1}{c_2 f (k)} \).

In the simplified scenario when \( \Delta, \bar{\tau}, \) and \( \sigma \) are known or can be bounded, a regret of \( O (\log T) \) is achievable. This can be accomplished by making the lengths of the exploration and matching phases constant (\( c_1 \) and \( c_2 \) respectively), which necessitates the following modifications:

1. The decreasing confidence intervals in (9) can be made constant instead, as \( C_{n,i} = \frac{\Delta}{2} \). The exponent in (11) then changes from \( \frac{1}{10M \sigma^2 \log^2 \frac{T}{c_3} + 1} \) to \( \frac{1}{160M \sigma^2} \), and now

\[
T_e (k) = \bar{c}_1 k.\]

Setting \( \bar{c}_1 = O \left( \frac{M \sigma^2}{2 \bar{\tau}} + M^2 \right) \), the bound of Lemma 1 in (6) holds for all \( k \).

2. The step sizes do not have to be decreasing and can instead be \( \varepsilon_k = \frac{\Delta}{2} \). Then the \( c_2 \log (k + 1) \) in (17) is replaced with a \( \bar{c}_2 \) and the \( 1 + \log k \) is replaced with \( \frac{2 \bar{\tau}}{\Delta} \). Hence, we can pick \( \bar{c}_2 = 100 \bar{\tau} \) to ensure that (17) holds for all \( k > k_0 \), where \( k_0 \leq \frac{T}{2 \bar{\tau}} + 150 \).

3. For (25), we see that if \( \bar{c}_1 \geq 3M^2 \geq 3NM \) then

\[
\left( 5NM e^{-\frac{2}{3}k} + e^{-\frac{2k}{3}} \right) \left( \frac{4}{3} \right)^k \leq 6 \text{ for all } k.
\]

With these modifications, we have that \( R_{\gamma^*,k} \leq M + O (\bar{c}_1 + \bar{c}_2 + c_3) \) for all \( k > k_0 \), as the terms that previously scaled with \( \log k \) are now constant. We conclude that by choosing \( \bar{c}_1 = O \left( \frac{M \sigma^2}{2 \bar{\tau}} + M^2 \right) \), \( \bar{c}_2 = O (\bar{\tau}) \), and \( c_3 = 1 \), our regret bound in (23) becomes \( O \left( \frac{M \sigma^2}{2 \bar{\tau}} + M^2 + \bar{\tau} \right) \log T \).

During the first \( k_0 = O \left( \frac{1}{\bar{\Delta}} \right) \) rounds, the algorithm as described above accrues an additive constant of \( O \left( e^{\frac{\Delta}{2}} \right) \) regret. This simply happens because there is no reason to run the first \( k_0 = O \left( \frac{1}{\bar{\Delta}} \right) \) exploitation phases, as they may have no chance to succeed. By eliminating these exploitation phases the additive constant becomes \( O \left( \frac{M \sigma^2}{2 \bar{\bar{\tau}}} + M^2 + \bar{\tau} \right) \), which disappears in the big O notation for \( T > e^{\frac{\Delta}{2}} \). This additive constant can eliminated altogether by reducing the lengths of the first \( k_0 \) exploration phases by a factor of \( \bar{\Delta} \).

This simplified scenario provides a better theoretical understanding of the dependence of Algorithm 3 on the problem parameters \( M, N, \bar{\tau}, \) and \( \Delta \). Nevertheless, in practice it is easy to choose large enough \( c_1 \) and \( c_2 \) such that \( k_0 \) and \( C_0 \) are very small across various settings, as we show in Section VIII.
VII. QoS REGRET ANALYSIS

In this section, we prove Theorem 2 that shows that the expected QoS Regret is $O(1)$. To this end, we require the following two lemmas. The first lemma bounds the probability that the exploitation of the $k$-th epoch fails. The second lemma bounds the probability that the exploitation phase lasts for $\tau$ turns, as a function of whether a correct $\gamma$-matching is played.

**Lemma 5** (QoS Exploitation Error Probability). Let $E_k$ be the event that a $\gamma$-matching is not played during the $k$-th exploitation phase. Then for $c_1 \geq 1$ we have

$$P(E_k) \leq 6NMe^{-\frac{k}{4}}.$$  \hspace{1cm} (27)

**Proof.** The proof follows by observing that the $k$-th exploitation phase necessarily succeeds if a matching was found in one of the past $k/2$ matching phases and all exploitation phases these matching phases depend on succeeded, from the $k/4$-th exploration phase up to the $k/2$-th exploration phase. On this event, the matching played in the $k$-th exploitation phase must be a $\gamma$-matching, assuming $k \geq k_0$. Recall that $E_{c,e}$ is the event where the exploitation phase succeeded in the $\ell$-th epoch, in that all estimated means stayed within their confidence intervals and all arms were pulled sufficiently many times. Define the event $A_k = \cap_{\ell=k/4}^{k/2} E_{c,e}$ for which $A_k = \bigcup_{\ell=k/4}^{k/2} E_{c,e}$. Then for all $k \geq k_0$ such that $\frac{k}{2} \log(\frac{k}{\ell+1}) \leq \frac{k}{2}$

$$P(E_k) = P(E_k | A_k) P(A_k) + P(E_k | \bar{A}_k) P(\bar{A}_k) \leq P(A_k) + P(E_k | A_k)$$

$$\leq P(A_k) + \sum_{\ell=k/4}^{k/2} e^{-c_1\ell} + \left(\frac{1}{2}\right)^{k/2}$$

$$\leq 3NM \sum_{\ell=k/4}^{k/2} e^{-c_1\ell} + \frac{1}{2}$$

$$\leq 3NM \frac{e^{-\frac{4k}{5}}}{1 - e^{-c_1}} + e^{-\frac{k}{5}}$$

$$\leq 5NM e^{-\frac{4k}{5}} + e^{-\frac{k}{5}}$$

$$\leq 6NM e^{-\frac{k}{4}}$$  \hspace{1cm} (28)

where (a) uses a union bound of $\bigcup_{\ell=k/4}^{k/2} E_{c,e}$ using Lemma 1 and that $E_k$ can occur only if a $\gamma$-matching was not found in the past $k/2$ epochs, so by Lemma 2, $P(E_k | A_k) \leq \left(\frac{1}{2}\right)^{k/2}$. Inequality (b) is a geometric sum, and (c) uses that $c_1 \geq 1$. \hfill $\square$

In contrast to Algorithm 3, the exploitation phase of Algorithm 4 has a random duration, making the length of each epoch random as well. This poses a synchronization challenge that one does not encounter when all epochs have fixed lengths. We overcome this by breaking the exploitation phase into subepochs of length $M$, and giving each player a discontent indicator $d$ to track her status. Every player starts with $d = run$, indicating that she believes that each player $n$ is receiving at least $\gamma(n)$. If her estimate of the reward of her arm falls below $\gamma(n) - C_n a_k$, then she switches to $d = signal$. This means that in the next subepoch she will signal to everyone to switch to $d = terminate$. A status of $d = terminate$ occurs when a player experiences a collision during the exploitation phase, which can only happen from a player with $d = signal$ signalling her discontent. Note that $terminate$ supersedes $signal$, in that if a player experiences a collision she will always set $d = terminate$, and if $d = terminate$ she will never switch to $d = signal$. We can see that the players all terminate the exploitation phase at the same step time, as no one terminates before the end of a subepoch. Additionally, no player will terminate in an epoch where all players start with $d = run$. In the first subepoch where a player sets $d = signal$, we see that all players will terminate the exploitation phase at the end of the subsequent subepoch.

**Lemma 6.** [QoS: Exploitation Error Probability] Assume that there exists a matching $a$ such that $\mu_n a_n \geq \gamma(n)$ for all $n$. Define the set of non-$\gamma$-matchings $A_e = \{a | \eta_i(a) = 1 \forall \ell \land \exists n \text{ s.t. } \mu_n a_n < \gamma(n)\}$. Let $\Delta = \min_{a \in A_e} \min_{n \in [N]} (\gamma(n) - \mu_n a_n) > 0$. Let $L_k$ be the number of turns spent in the exploitation phase of epoch $k$ until one of the players sets $d = signal$. Then

1) For the case where $E_k$ occurred (players were not playing a $\gamma$-matching) we have for all $\tau \geq e^{16/\Delta^4}$ that

$$P(L_k = \tau | E_k) \leq e^{-\frac{\tau^2}{1000}}.$$  \hspace{1cm} (29)

2) For the case where $E_k$ did not occur (players were playing a $\gamma$-matching) we have for all $\tau$ that

$$P(L_k = \tau | \bar{E}_k) \leq \frac{2N^2Me^{-\frac{16\tau}{4M}}}{e^{-\frac{16\tau}{4M}}}.$$  \hspace{1cm} (30)

**Proof.** For compactness of notation, we let $a \triangleq a_k$. For this proof, let $r^k_e(\ell)$ be the $\ell$-th random reward that player $n$ uses to decide if to switch to $d = signal$ for the $k$-th exploitation phase. Note that the first $V_{k,n,a_n} = V_{k/2,n,a_n}$ of these samples were collected during the past $k/2$ exploitation phases.

Throughout the proof, we define $\Delta_n^k = \mu_n a_n - \mu_n a_i$ gre and the corresponding error $\Delta_n^k = \mu_n a_n - \mu_n a_i$ for every $n,i$.\hfill $\square$
If $E_k$ occurred, then there must exist some player $n^*$ with $\mu_{n^*, a^*} < \gamma(n)$, where $n^*$ and $a^*$ are random. Define the event $A_{n, i} = \{n^* = n, a^* = i\}$ for every $n, i$. Then for all $\tau$ such that $\max \tilde{C}_{n, a}(\tau) \leq (\log \tau)^{-1/4} \leq \frac{\Delta_\tau}{2}$, equivalently for all $\tau \geq e^{16/\Delta_\tau^2}$:

$$\Pr\left(L_k = \tau + 1 \mid E_k, \left\{V_{n,i}^{k/[2]} = v_{n,i}\right\}_{n,i}\right) \leq \Pr\left(\hat{\mu}_{n^*, a^*} > \gamma(n) - \tilde{C}_{n, a}(\tau) \mid E_k\right)$$

$$\leq \Pr\left(\Delta_{n^*, a^*} \geq \frac{\Delta_\tau}{2} \mid E_k\right)$$

$$\leq \Pr\left(\Delta_{n^*, a^*} \geq \Delta_\tau \mid E_k\right)$$

$$\leq \sum_{n=1}^{N} \sum_{i=1}^{M} \Pr\left(\Delta_{n, i} \geq \Delta_\tau \mid E_k, A_{n, i}\right) \Pr\left(A_{n, i} \mid E_k\right)$$

$$\leq e^{-\frac{n+\min_{n,i} \Delta_\tau^2}{2\sigma^2}} \tag{31}$$

where (a) follows since if $L_k = \tau + 1$, then player $n^*$ must have had $\hat{\mu}_{n^*, a^*} = \sum \frac{v_{n,i}^{k/[2]} - \tilde{C}_{n, a}(\tau)}{\gamma(n) - \tilde{C}_{n, a}(\tau)} \geq \gamma(n) - \tilde{C}_{n, a}(\tau)$ otherwise we would have had $L_k \leq \tau$ since player $n^*$ would have switched to $d$ signal at or before the $\tau$-th exploitation turn. Inequality (b) follows for all $\tau \leq e^{16/\Delta_\tau^2}$, since then $\frac{\Delta_\tau}{2}$ and $\gamma(n) - \tilde{C}_{n, a}(\tau) \geq \frac{\Delta_\tau}{2}$. Inequality (c) (uses Hoeffding’s inequality for a sum of independent $\sigma$-sub-Gaussian random variables [42, Theorem 2.6.2] (derived via Chernoff). It also uses that $E_k$ is independent of $\{v_{n,i}(t)\}$ since $E_k$ is only a function of samples from before epoch $k/2$, due to the sample splitting. Since (31) holds for every $\min_{n,i} v_{n,i} \geq 0$ then

$$\Pr\left(L_k = \tau + 1 \mid E_k\right) \leq e^{-\frac{\tau+\min_{n,i} \Delta_\tau^2}{\sigma^2}} \tag{32}$$

We now examine the case where $E_k$ did not occur. We bound the probability that after $\tau$ turns of playing a correct $\gamma$-matching in the exploitation phase, the estimation of some player $n$ of her arm $a_n$ will be below $\gamma(n) - \tilde{C}_{n, a_n}(\tau)$:

$$\Pr\left(L_k = \tau \mid E_k, \left\{V_{n,i}^{k/[2]} = v_{n,i}\right\}_{n,i}\right)$$

$$\leq \sum_{n=1}^{N} \sum_{i=1}^{M} \Pr\left(\mu_{n, i} \leq \gamma(n) - \tilde{C}_{n, a_n}(\tau) \mid E_k\right)$$

$$\leq \sum_{n=1}^{N} \sum_{i=1}^{M} \Pr\left(\Delta_{n, i} \leq -\tilde{C}_{n, a_n}(\tau) \mid E_k, a_n = i\right) \Pr\left(a_n = i \mid E_k\right)$$

$$\leq \sum_{n=1}^{N} \sum_{i=1}^{M} e^{-\frac{\tau+\min_{n,i} \Delta_\tau^2}{\sigma^2}} \Pr\left(\Delta_{n, i} \leq \Delta_\tau \mid E_k\right)$$

$$\leq \sum_{n=1}^{N} \sum_{i=1}^{M} e^{-\frac{\tau+\min_{n,i} \Delta_\tau^2}{\sigma^2}} \Pr\left(a_n = i \mid E_k\right)$$

$$\leq \sum_{n=1}^{N} \sum_{i=1}^{M} e^{-\frac{\tau+\min_{n,i} \Delta_\tau^2}{\sigma^2}} \Pr\left(a_n = i \mid E_k\right)$$

$$\leq Ne^{-\frac{\tau+\min_{n,i} \Delta_\tau^2}{4\sigma^2}} \tag{33}$$

where (a) is the union bound and (b) follows since given $E_k$ we have $\mu_{n, a_n} \geq \gamma(n)$ for every $n$. Inequality (c) uses Hoeffding’s inequality for a sum of independent $\sigma$-sub-Gaussian random variables [42, Theorem 2.6.2] (derived via Chernoff). Note that we also used that $\{v_{n,i}(t)\}$ and $\tilde{C}_{n, a_n}(\tau)$ are independent of $E_k$ since they only use samples from after the $k/2$-th exploration phase. Inequality (d) uses $\sqrt{x + y} \geq \frac{1}{2} \sqrt{2x} + \frac{1}{2} \sqrt{2y}$ since $\sqrt{z}$ is concave.

Let $\gamma_{\min} = \min_{n,i} (V_{n,i}^{k/[2]} - V_{n,i}^{k/[2]})$. Using (33) we obtain

$$\Pr\left(L_k = \tau \mid E_k\right)$$

$$\leq Ne^{-\frac{\tau+\min_{n,i} \Delta_\tau^2}{4\sigma^2}} \sum_{n=1}^{N} \sum_{i=1}^{M} e^{-\frac{\tau+\min_{n,i} \Delta_\tau^2}{4\sigma^2}} \Pr\left(V_{n,i}^{k/[2]} = v_{n,i}\right)$$

$$\leq Ne^{-\frac{\tau+\min_{n,i} \Delta_\tau^2}{4\sigma^2}} \sum_{n=1}^{N} \sum_{i=1}^{M} e^{-\frac{\tau+\min_{n,i} \Delta_\tau^2}{4\sigma^2}} \Pr\left(V_{n,i}^{k} = v\right)$$

$$\leq Ne^{-\frac{\tau+\min_{n,i} \Delta_\tau^2}{4\sigma^2}} \left(N M e^{-\frac{\tau+\min_{n,i} \Delta_\tau^2}{4\sigma^2}} \log \frac{1}{\epsilon} + e^{-\sqrt{\frac{\tau+\min_{n,i} \Delta_\tau^2}{4\sigma^2}} \log \frac{1}{\epsilon}}\right)$$

$$\leq 2 N^2 M e^{-\frac{\tau+\min_{n,i} \Delta_\tau^2}{4\sigma^2}} e^{-\sqrt{\frac{\tau+\min_{n,i} \Delta_\tau^2}{4\sigma^2}} \log \frac{1}{\epsilon}} \tag{34}$$

where (a) follows similarly to (8), by

$$\Pr\left(V_{n,i}^{k} \leq T_{e}(k) - T_{e}(k/[2])\right)$$

$$\leq N M e^{-\frac{\tau+\min_{n,i} \Delta_\tau^2}{4\sigma^2}} \left(N M e^{-\frac{\tau+\min_{n,i} \Delta_\tau^2}{4\sigma^2}} \log \frac{1}{\epsilon} + e^{-\sqrt{\frac{\tau+\min_{n,i} \Delta_\tau^2}{4\sigma^2}} \log \frac{1}{\epsilon}}\right)$$

$$\leq N M e^{-\frac{\tau+\min_{n,i} \Delta_\tau^2}{4\sigma^2}} \log \frac{1}{\epsilon}. \tag{35}$$

Equipped with Lemmas 5 and 6, we can now prove Theorem 2, which is the main result of this section. Intriguingly, the proof implies that the expected duration of the $k$-th exploitation length vanishes with $k$. This is possible since, with high probability, the $k$-th exploitation phase never happens. Indeed, epochs with a large $k$ index only occur with exponentially vanishing probability. This means that at some point the algorithm gets stuck in an exploitation phase where a $\gamma$-matching is played forever.

**Theorem 2 (QoS Regret Bound). Assume that there exists a matching a such that $\mu_{n, a_n} \geq \gamma(n)$ for all $n$. Assume that the rewards $\{r_{n,i}(t)\}$ are independent in $n$ and i.i.d. with $t$, with a continuous (see Remark 1) $\sigma_{n,i}$-sub-Gaussian distribution. Without loss of generality, we assume that the expectations $\{\mu_{n, i}\}$ are in $[0, 1]$. Let $T$ be the finite deterministic horizon of the game, which is unknown to the players. Let each player play according to Algorithm 4 with $c_1, c_2, c_3 \geq 1$. Then the QoS regret is upper bounded by a constant with respect to $T$, i.e., $R_T(T) = O(1)$.

**Proof of Theorem 2.** Let $R_{t,k}$ be the QoS regret in epoch $k$, and $R_{0,k}$ be the regret accumulated during the $k$-th exploitation
and matching phases. Then for some constant $C_1 > 0$:

$$\mathbb{E}\{R_{\gamma,k} \mid E_k\} \leq (a) \quad R_{0,k} + \sum_{\tau=1}^{\infty} (\tau + 2M) \mathbb{P}(L_k = \tau \mid E_k)$$

$$\leq R_{0,k} + \sum_{\tau=1}^{\infty} (\tau + 2M) + \sum_{\tau=1}^{\infty} (\tau + 2M) e^{-\frac{\tau^2}{10M\sigma^2}}$$

$$\leq (c_1 + c_2) \log(k + 1) + C_1 M$$

(36)

where (a) follows since if $L_k = \tau$ then no more than $\tau + 2M$ regret is incurred for $\tau + 2M$ turns, since the exploitation phase terminates at most $2M$ turns after a player has set $d = \text{signal}$. Inequality (b) uses (32).

The $k$-th exploitation phase incurs regret under $E_k$ only if it terminates, in which case we accumulate the regret from at most $2M$ signaling turns. Additionally, we incur $R_{0,k+1}$ regret from the exploration and matching phases of the $k$-th epoch. Using this observation, we conclude that for some constants $C_2, C_3, C_R > 0$

$$R_{\gamma}(T) \leq \sum_{k=1}^{\infty} \mathbb{E}\{R_{\gamma,k}\} = \sum_{k=1}^{k_0-1} \mathbb{E}\{R_{\gamma,k}\}$$

$$+ \sum_{k=k_0}^{\infty} \left( \mathbb{P}(E_k) \mathbb{E}\{R_{\gamma,k} \mid E_k\} + \mathbb{P}(\bar{E}_k) \mathbb{E}\{R_{\gamma,k} \mid \bar{E}_k\} \right)$$

$$\leq 6NM \sum_{k=k_0}^{\infty} e^{-\frac{k}{M}} ((c_1 + c_2) \log(k + 1) + C_1 M)$$

$$+ (c_1 + c_2) \log(k_0 + 1) + 2 + 3M + C_0$$

$$+ \sum_{k=k_0}^{\infty} \sum_{\tau=1}^{\infty} (R_{0,k+1} + 2M) \mathbb{P}(L_k = \tau \mid \bar{E}_k)$$

$$\leq C_0 + C_2 NM (c_1 + c_2 + C_3 M)$$

$$+ 2N^2 M \sum_{k=k_0}^{\infty} \sum_{\tau=1}^{\infty} (R_{0,k+1} + 2M) e^{-\frac{\sqrt{\tau^2 + 2M \log(\frac{2M}{\tau})}}{4\sigma^2}} e^{-\frac{\sqrt{\tau^2}}{4\sigma^2}}$$

$$\leq C_0 + C_2 NM (c_1 + c_2 + C_3 M)$$

$$+ 2N^2 M \sum_{k=k_0}^{\infty} \sum_{\tau=1}^{\infty} (R_{0,k+1} + 2M) e^{-\frac{\sqrt{\tau^2 + 2M \log(\frac{2M}{\tau})}}{4\sigma^2}}$$

$$\leq C_0 + O(N^2M^3 + N^2M^2 (c_1 + c_2))$$

(37)

where (a) uses Lemma 5 on $\mathbb{P}(E_k)$, $\mathbb{P}(\bar{E}_k) \leq 1$, (34) and (36). We also defined $C_0 = \sum_{k=1}^{k_0-1} \mathbb{E}\{R_{\gamma,k}\}$. Inequality (b) uses $R_{0,k+1} < (c_1 + c_2) \log(k + 2) + 2 + M$ and then bounds the converging sums using an integral, as in the integral test for convergence.

We conclude that by choosing $c_1 = O\left(\frac{M^2}{\Delta_2^2} + M^2\right)$ and $c_2 = O(\bar{E})$ we obtain $k_0 = 1$ and $C_0 = 0$. Then, by bounding the infinite converging sums in (37) using the integral test for convergence, we obtain that $\sum_{\tau=1}^{\infty} e^{-\frac{\sqrt{\tau^2}}{4\sigma^2}} = O(\sigma^4)$, and since $c_1 = \Omega(M\sigma^2)$ we also obtain that

$$\sum_{k=1}^{\infty} (c_1 + c_2 + 4M) e^{-\frac{\sqrt{\tau^2}}{4\sigma^2}} = O(\sigma^2(\bar{c}_1 + \bar{c}_2 + M)).$$

(39)

Hence, the regret $C_R$ can be bounded as

$$C_R \leq C_u \left( N \left( NM\sigma^6 + M \left( \frac{M^2}{\Delta_2^2} + M^2 + \bar{E} \right) \right) + N M^2 \frac{\sigma^2}{\Delta_2^2} + N M \frac{\sigma^4}{\Delta_2^4} + N^2 M^2 \sigma^6 \right).$$

(40)

for some universal constant $C_u$ that does not depend on any problem parameters.

Nevertheless, the above analysis is mainly theoretical, since our simulations in Section VIII imply that the constant $C_R$ is better than our theoretical bounds suggest. Specifically, $C_R$ does not seem to increase for larger $N, M$ or smaller $\Delta_2$.
We simulated three multi-armed bandit games with the following expected rewards matrices $U_1, U_2, U_3$:

$$U_1 = \begin{bmatrix}
    \frac{1}{10} & \frac{1}{10} & \frac{1}{10} & \frac{1}{10} \\
    \frac{1}{10} & \frac{1}{10} & \frac{1}{10} & \frac{1}{10} \\
    \frac{1}{10} & \frac{1}{10} & \frac{1}{10} & \frac{1}{10} \\
    \frac{1}{10} & \frac{1}{10} & \frac{1}{10} & \frac{1}{10}
\end{bmatrix} \quad U_2 = \begin{bmatrix}
    0 & 0 & 0 & 0 \\
    0 & 0 & 0 & 0 \\
    0 & 0 & 0 & 0 \\
    0 & 0 & 0 & 0
\end{bmatrix}$$

$$U_3 = \begin{bmatrix}
    \frac{1}{4} & \frac{3}{10} & \frac{3}{10} & \frac{3}{10} \\
    \frac{1}{4} & \frac{3}{10} & \frac{3}{10} & \frac{3}{10} \\
    \frac{1}{4} & \frac{3}{10} & \frac{3}{10} & \frac{3}{10} \\
    \frac{1}{4} & \frac{3}{10} & \frac{3}{10} & \frac{3}{10}
\end{bmatrix}$$

$U_1$ has $4! = 24$ matchings: 16 with minimal expected reward $\frac{1}{10}$, 7 with $\frac{1}{4}$, and only one optimal matching with $\frac{3}{10}$. The optimal sum of expected rewards is 2.15, but its matching has a minimal expected reward of $\frac{1}{10}$. Hence, an algorithm that optimizes the sum of expected rewards will have regret $\Omega \left( \frac{t}{4} \right)$.

$U_2$ has $7! = 5040$ matchings: 4792 with minimal expected reward of 0, 198 with $\frac{1}{10}$, 26 with $\frac{1}{4}$, 19 with $\frac{3}{10}$, 3 with $\frac{1}{4}$, and only two optimal matchings with $\frac{3}{10}$. The optimal sum of expected rewards is 5.3, but its matching has a minimal expected reward of 0. Hence, an algorithm that optimizes the sum of expected rewards will have regret $\Omega \left( \frac{t}{4} \right)$.

$U_3$ has $10! = 3628800$ matchings: 2761572 with minimal expected rewards of 0.05, 785048 with 0.1, 62066 with 0.2, 16180 with 0.25, 3798 matchings with 0.3 and only 136 optimal matchings with 0.4. The optimal sum of expected rewards is 7.35, but its matching has a minimal expected reward of 0.3. Hence, an algorithm that optimizes the sum of expected rewards will have regret $\Omega \left( \frac{t}{10} \right)$.

For these simulations, given expectations $\{\mu_{n,i}\}$, the rewards are generated as $r_{n,i}(t) = \mu_{n,i} + z_{n,i}(t)$ where $\{z_{n,i}(t)\}$ are independent random Gaussian variables, each with zero mean and variance $\sigma_{n,i}^2 = 1$ for all $n, i$. We used $c_1 = c_2 = c_3 = 4000$ for all experiments, which ensures that the additive constant $C_0$ is small (since $k_0$ is small), as the exploration and matching phases are long from the beginning.

In Fig. 1, we present the max-min fairness regret of the My Fair Bandit Algorithm versus time, averaged over 100 realizations, for the three scenarios above. In our implementation, at the beginning of the $k$-th matching phase, each player played her action from the last exploitation phase if it is in $E_k$, or a random action from $E_k^*$ otherwise. Although it has no
effect on the theoretical bounds, it improved the performance in practice significantly. Another practical modification was to use $C_{n,i} = \frac{0.01}{\log \pi V_{n,i}}$, which requires larger $c_1$ but does not affect the analysis otherwise. The step size sequence, which is updated only on resets, was chosen as $\epsilon_k = \frac{0.5}{\gamma n \log^2 T}$. The shaded area is bounded between the 1% and 99% quantiles. It can be seen that in all 100 experiments the players learned the max-min optimal matching quickly, in less than 6 epochs for all scenarios. This suggests that $k_0$ is much smaller than our theoretical bound. As expected, the regret scales (approximately) logarithmically as guaranteed by Theorem 1.

In Fig. 2, we present the QoS regret of the QoS Algorithm versus time, averaged over 100 realizations, for the three scenarios above. For each scenario, we chose $\gamma^{(n)} = \gamma^*$ for all $n$, as this is the most challenging scenario. As a practical enhancement, players used the samples from all past exploitation phases to estimate the expected rewards. It can be seen that quite early on, the algorithm finds a correct $\gamma$-matching and all players play this matching forever, leading to constant regret as guaranteed by Theorem 2. This constant is a random variable that varies between realizations depending primarily on how early the correct matching is found.

IX. CONCLUSIONS

We studied a multi-player multi-armed bandit game where players cooperate to learn how to allocate arms, thought of as resources, while guaranteeing fairness between them. To allow for a meaningful notion of fairness, we employed the heterogeneous model where arms can have different expected rewards for each player. Our algorithm operates in the restrictive setting of bandit feedback, where each player only observes the reward for the arm she played and cannot observe the actions or rewards of other players. This reward is zero if one or more other players picked the same arm, modeling a collision. No communication between players is possible.

We presented and studied fully distributed algorithms for two metrics of fairness. One notion of fairness is max-min fairness, where we maximize the minimal expected reward of $\gamma^{(n)} > 0$. For this case, we proposed another fully distributed algorithm. Surprisingly, this algorithm was able to leverage the knowledge of a feasible $\gamma$ to obtain $O(1)$ regret that is independent of $T$. This includes the case of choosing $\gamma^{(n)} = \gamma^*$ for all $n$, if the max-min value is known. As multi-player systems are inherently more complex than their single-player counterparts, this raises the general question of what information or richer feedback model can significantly simplify the learning for distributed agents.

Possible future research directions include generalizing the results of this paper to restless bandits, where the reward process of each arm is Markovian and evolves every turn [44], [45]. Another interesting extension is to contextual multi-player bandits, where all players observe a context every turn, which affects their rewards [46].

REFERENCES


