Combinatorial Sleeping Bandits with Fairness Constraints

Fengjiao Li, Jia Liu, and Bo Ji

Abstract—**The multi-armed bandit (MAB) model has been widely adopted for studying many practical optimization problems (network resource allocation, ad placement, crowdsourcing, etc.) with unknown parameters. The goal of the player (i.e., the decision maker) here is to maximize the cumulative reward in the face of uncertainty. However, the basic MAB model neglects several important factors of the system in many real-world applications, where multiple arms (i.e., actions) can be simultaneously played and an arm could sometimes be "sleeping" (i.e., unavailable). Besides reward maximization, ensuring *fairness* is also a key design concern in practice. To that end, we propose a new *Combinatorial Sleeping MAB model with Fairness constraints*, called *CSMAB-F*, aiming to address the aforementioned crucial modeling issues. The objective is now to maximize the reward while satisfying the fairness requirement of a minimum selection fraction for each individual arm. To tackle this new problem, we extend an online learning algorithm, called *Upper Confidence Bound (UCB)*, to deal with a critical tradeoff between *exploitation* and *exploration* and employ the virtual queue technique to properly handle the fairness constraints. By carefully integrating these two techniques, we develop a new algorithm, called *Learning with Fairness Guarantee (LFG)*, for the CSMAB-F problem. Further, we rigorously prove that not only LFG is *feasibility-optimal*, but it also has a time-average *regret* upper bounded by $\frac{N}{2\eta}+\frac{\beta_1\sqrt{mNT\log T}+\beta_2 N}{T}$, where N is the total number of arms, m is the maximum number of arms that can be simultaneously played, T is the time horizon, β_1 and β_2 are constants, and η is a design parameter that we can tune. Finally, we perform extensive simulations to corroborate the effectiveness of the proposed algorithm. Interestingly, the simulation results reveal an important tradeoff between the regret and the speed of convergence to a point satisfying the fairness constraints.**

I. Introduction

The *multi-armed bandit (MAB)* model has been widely adopted for studying many practical optimization problems (network resource allocation, ad placement, crowdsourcing, etc.) with unknown parameters (see, e.g., [1]). In the basic stochastic MAB setting, there are N arms (i.e., actions), each of which, if played, returns a random reward to the player (i.e., the decision maker). The random reward of each arm takes values in $[0, 1]$ and is assumed to be *independent and identically distributed (i.i.d.)* over time. However, the reward distributions and the mean rewards are unknown *a priori*. The player decides which single arm to play in each round for a given time horizon of T rounds, with a goal of maximizing the cumulative reward in the face of unknown mean rewards.

However, this basic MAB model neglects several important factors of the system in many real-world applications, where multiple actions can be simultaneously taken and an action could sometimes be "sleeping" (i.e., unavailable). Take wireless scheduling for example: multiple clients compete for a shared wireless channel to transmit packets to a common access point (AP). The AP decides which client(s) can transmit at what times. A successfully delivered packet will generate a random reward, which could represent the value of the information contained in the packet. In each scheduling cycle, multiple clients could be scheduled for simultaneous transmissions as the channel can typically be divided into multiple "sub-channels" using multiplexing technologies [2]. On the other hand, some clients may be unable to transmit packets when experiencing a poor channel condition (due to fading or mobility). Furthermore, in addition to maximizing the reward, ensuring *fairness* among the clients or providing *Quality of Service (QoS)* guarantees to the clients is also a key design concern in wireless scheduling [3], [4], as well as in network resource allocation in general [5]. These important factors (i.e., combinatorial actions, availability of actions, and fairness) are commonly shared by many other applications too (see more detailed discussions in Section VI). However, it remains largely unexplored in the literature to carefully integrate all these factors into a unified MAB model.

To that end, in this paper we propose a new *Combinatorial Sleeping MAB model with Fairness constraints*, called *CSMAB-F*, aiming to address the aforementioned modeling issues, which are practically important for a wide variety of applications. Compared to the basic MAB setting, in the proposed framework the set of available arms follows a certain distribution that is assumed to be *i.i.d.* over time and is unknown *a priori*. However, the information of available arms will be revealed at the beginning of each round. The player can then play multiple, but no more than m, available arms and receives a compound reward being the weighted sum of the rewards of the played arms. We also impose fairness constraints that the player must ensure a (possibly different) minimum selection fraction for each individual arm. The goal is now to maximize the reward while satisfying the fairness requirement. We summarize our main contributions as follows.

First, to the best of our knowledge, *this is the first work that integrates all three critical factors of combinatorial arms, availability of arms, and fairness into a unified MAB model.* The proposed CSMAB-F framework successfully addresses

This work was supported in part by the NSF under Grants CNS-1651947, CCF-1657162, ECCS-1818791, CCF-1758736, CNS-1758757, and CNS-1446582, the ONR under Grant N00014-17-1-2417, and the AFRL under Grant FA8750-18-1-0107.

Fengjiao Li (fengjiao.li@temple.edu) and Bo Ji (boji@temple.edu) are with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA, USA. Jia Liu (jialiu@iastate.edu) is with the Department of Computer Science, Iowa State University, Ames, IA, USA.

these crucial modeling issues. This new problem, however, becomes much more challenging. In particular, integrating fairness constraints adds a new layer of difficulty to the combinatorial sleeping MAB problem that is already quite challenging. This is because not only the player encounters a *fundamental tradeoff* between *exploitation* (i.e., staying with the currently-known best option) and *exploration* (i.e., seeking better options) when attempting to maximize the reward, but she is also faced with a *new dilemma*: how to manage the balance between maximizing the reward and satisfying the fairness requirement? Several well-known MAB algorithms can successfully handle the exploitation-exploration tradeoff, but none of them was born with fairness constraints in mind.

To address this new challenge, we extend an online learning algorithm, called *Upper Confidence Bound (UCB)*, to deal with the exploitation-exploration tradeoff and employ the *virtual queue technique* to properly handle the fairness constraints. By carefully integrating these two techniques, we develop a new algorithm, called *Learning with Fairness Guarantee (LFG)*, for the CSMAB-F problem. Further, we rigorously prove that not only LFG is *feasibility-optimal*, but it also has a time-average *regret* (i.e., the reward difference between an optimal algorithm that has *a priori* knowledge of the mean rewards and the considered algorithm) upper bounded by $\frac{N}{2\eta} + \frac{\beta_1\sqrt{mNT\log T}+\beta_2 N}{T}$, where β_1 and β_2 are constants and η is a design parameter that we can tune. Note that our regret analysis is more challenging as the traditional regret analysis becomes non-applicable here due to the integration of virtual queues for handling the fairness constraints.

Finally, we conduct extensive simulations to elucidate the effectiveness of the proposed algorithm. From the simulation results, we observe that LFG can effectively meet the fairness requirement while achieving a good regret performance. Interestingly, the simulation results also reveal a critical tradeoff between the regret and the speed of convergence to a point satisfying the fairness constraints. We can control and optimize this tradeoff by tuning the value of parameter η.

The rest of the paper is organized as follows. We first discuss related work and describe the proposed CSMAB-F framework in Sections II and III, respectively. Then, we develop the LFG algorithm for the CSMAB-F problem in Section IV, followed by the performance analysis in Section V. Detailed discussions about several real-world applications are provided in Section VI. Finally, we present simulation results in Section VII and make concluding remarks in Section VIII.

II. Related Work

Starting with the seminal work of [6], the MAB problems have been extensively studied in a large body of work (see, e.g., [1], [7]). In the basic MAB setting, the authors of [6] establish a fundamental logarithmic lower bound on the regret of a class of "uniformly good policies" and propose UCB policies that asymptotically achieve the lower bound. Further, the work of [8] shows that logarithmic regret can be achieved uniformly over time rather than asymptotically by simpler sample-mean-based UCB policies and an ϵ_t-greedy policy.

Following this line of research, different MAB variants have been proposed to model several important factors of the system in real-world applications. The ones that are relevant to ours include combinatorial MAB (CMAB) where multiple arms form a super arm and can be simultaneously played [9]–[14] and sleeping MAB (SMAB) where an arm could sometimes be "sleeping" (i.e., unavailable) [15]–[18]. Being the first to study the CMAB problem, the work of [9] considers combinations of a fixed number of simultaneous plays. This simple combinatorial structure has been generalized to permutations [10] and matroids [11]. The work of [12], [13] generalizes linear reward functions considered in [9]–[11] to include a large class of linear and nonlinear rewards. In [14], the authors prove a tight problem-specific lower bound for stochastic CMAB (where the reward of each played arm rather than the combinatorial reward is revealed) and propose an efficient sampling algorithm with an improved multiplicative factor. The work of [15] is among the first to study the SMAB problem. This work provides a computationally efficient algorithm for the setting of stochastic rewards while allowing both stochastic and adversarial availability. Follow-up work of [16], [17] studies the setting of adversarial rewards while the availability of arms is either stochastic or adversarial. Very recently, the authors of [18] analyze the performance of Thompson Sampling for the SMAB problem and show that it empirically performs better than other algorithms.

MAB settings with constraints have also been considered in prior studies. Most of them focus on bandits with budgets (see, e.g., [19]) or bandits with knapsacks (see, e.g., [20]), where no more plays can be made if the budget/knapsack constraints are violated. Hence, these types of constraints are very different from the *long-term* fairness constraints we consider in this paper. Some very recent work considers multi-type rewards [21] and multi-level rewards [22], [23]. They introduce a minimum guarantee requirement that the total reward of some type/level must be no smaller than a given threshold. However, these studies differ significantly from ours in the following key aspects. First, and most importantly, their constraints do not model fairness among arms. The required minimum guarantee is for the total rewards (of some type/level) rather than for each individual arm. Second, no learning algorithm is proposed in [21]; the proposed learning algorithms in [22], [23] may violate the constraints, although they show provable violation bounds. Third, they assume that all the arms are available at all times. Last but not least, the proof techniques for regret analysis in [22], [23] are very different from ours.

Fairness in online learning has been studied in [24], [25]. A key idea of their proposed fair algorithm is that two arms should be played with equal probability until they can be distinguished with a high confidence. Another work [26] studies how to learn proportionally fair allocations by considering the maximization of a logarithmic utility function. These studies are less relevant to our work, although they share some high-level similarities with ours in modeling fairness.

At a technical level, the work of [27] that integrates learning and queueing is most related to ours. We follow a similar line of regret analysis in [27] for deriving the upper bound. However, they do not explicitly model fairness constraints, nor do they consider the availability of arms.

III. System model and problem formulation

In this section, we describe the detailed setting of our proposed CSMAB-F framework. Let $\mathcal{N} = \{1, 2, \ldots, N\}$ denote the set of N arms. Each arm $i \in \mathcal{N}$ is associated with a reward $X_i(t)$ in round t, where $t = 0, 1, 2, \ldots$. The reward is a random variable on $[0, 1]$ and follows a certain distribution with mean μ_i. We assume that the reward for each arm is *i.i.d.* over time. The mean reward vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_N)$ is unknown *a priori*. In our setting, an arm could sometimes be "sleeping" (i.e., unavailable). Let $A(t) \in \mathcal{P}(\mathcal{N})$ denote the set of available arms in round t, where $\mathcal{P}(\mathcal{N})$ is the power set of $\mathcal{N}$. We use $P_{\mathbf{A}}(Z) \triangleq P(A(t) = Z)$, where $Z \in \mathcal{P}(\mathcal{N})$, to denote the distribution of available arms, which is assumed to be *i.i.d* over time. This distribution is unknown *a priori*, but the set of available arms $A(t)$ will be revealed to the player at the beginning of each round t.

In each round, the player is allowed to play multiple, but no more than m, available arms (i.e., arms belonging to $A(t)$). Each subset of available arms is also called a *super arm* [12]. We restrict the size of a chosen super arm to be no larger than m so as to account for resource constraints (see discussions on applications in Section VI). Let $\mathcal{S}(Z)$ represent the set of all feasible super arms when the set of available arms Z is observed, i.e., $\mathcal{S}(Z) \triangleq \{S \subseteq Z : |S| \leq m\}$, where $|S|$ denotes the cardinality of set S. In round t, a player selects a super arm $S(t) \in \mathcal{S}(A(t))$ and receives a compound reward $R(t)$, which is a weighted sum of the rewards of the played arms, i.e., $R(t) \triangleq \sum_{i \in S(t)} w_i X_i(t)$, where w_i is the weight of arm i. We assume that the weights w_i are fixed positive numbers known *a priori* and are upper bounded by a finite constant $w_{\max} > 0$. The goal of the player is to maximize the expected time-average reward for a given time horizon of T rounds, i.e., $\mathbb{E}[\frac{1}{T}\sum_{t=0}^{T-1} R(t)]$.

To describe the action for each individual arm, we use a binary vector $\mathbf{d}(t) = (d_1(t), \ldots, d_N(t))$ to indicate whether each arm is played or not in round t, where $d_i(t) = 1$ if arm i is played, i.e., $i \in S(t)$; otherwise, $d_i(t) = 0$. Then, the action vector $\mathbf{d}(t)$ must satisfy $\sum_{i=1}^{N} d_i(t) \leq m$ for all $t \geq 0$.

As we discussed in the introduction, in addition to maximize the reward, ensuring fairness among the arms is also a key design concern for many real-world applications. To model the fairness requirement, we introduce the following constraints on a minimum selection fraction for each individual arm:

$$\liminf_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[d_i(t)] \geq r_i \ \forall i \in \mathcal{N}, \tag{1}$$

where $r_i \in (0, 1)$ is the required minimum fraction of rounds in which arm i is played. The minimum selection fraction vector $\mathbf{r} = (r_1, \ldots, r_N)$ is said to be *feasible* if there exists a policy that makes a sequence of decisions $S(t)$ for $t \geq 0$ such that (1) is satisfied. Then, the *maximal feasibility region* $\mathcal{C}$ is defined as the set of all such feasible vectors $\mathbf{r} \in (0, 1)^N$. A policy is said to be *feasibility-optimal* if it can support any vector $\mathbf{r}$ (i.e., (1) is satisfied) strictly inside the maximal feasibility region $\mathcal{C}$.

We now consider the special class of stationary and randomized policies called *A-only policies*. An A-only policy observes the set of available arms $A(t)$ for each round t and independently chooses a super arm $S(t) \in \mathcal{S}(A(t))$ as a (possibly randomized) function of the observed $A(t)$ only. An A-only policy α is characterized by a group of probability distributions, denoted by $\mathbf{q} = [q_S(Z), \forall S \in \mathcal{S}(Z), \forall Z \in \mathcal{P}(\mathcal{N})]$, where $q_S(Z)$ is the probability that policy α chooses super arm $S \in \mathcal{S}(Z)$ when observing the set of available arms $Z \in \mathcal{P}(\mathcal{N})$, and $\sum_{S \in \mathcal{S}(Z)} q_S(Z) = 1$ for all $Z \in \mathcal{P}(\mathcal{N})$. Then, under policy α, the action $d_i^{\alpha}(t)$ is *i.i.d.* over time with the following mean:

$$\mathbb{E}[d_i^{\alpha}(t)] = \sum_{Z \in \mathcal{P}(\mathcal{N})} P_{\mathbf{A}}(Z) \sum_{S \in \mathcal{S}(Z): i \in S} q_S(Z), \tag{2}$$

for every arm $i \in \mathcal{N}$ and for all $t \geq 0$, and thus, constraint (1) is equivalent to $\mathbb{E}[d_i^{\alpha}(t)] \geq r_i$ for every arm $i \in \mathcal{N}$. Further, we have the following lemma.

Lemma 1. *If a vector* $\mathbf{r}$ *is strictly inside the maximal feasibility region* $\mathcal{C}$*, then there exists an* A*-only policy that can support vector* $\mathbf{r}$.

Proof. The proof is omitted as it is quite standard and follows a similar line of analysis in the proof of Theorem 4.5 in [28] (see [28, pp. 92-95]). □

Lemma 1 implies that there exists an optimal A-only policy. Hence, assuming that the mean reward vector $\boldsymbol{\mu}$ is known in advance, one can formulate the reward maximization problem with minimum selection fraction constraint as the following linear program (LP):

$$\underset{\mathbf{q}}{\text{maximize}} \quad \sum_{Z \in \mathcal{P}(\mathcal{N})} P_{\mathbf{A}}(Z) \sum_{S \in \mathcal{S}(Z)} q_S(Z) \sum_{i \in S} w_i \mu_i \tag{3a}$$

$$\text{subject to} \quad \sum_{Z \in \mathcal{P}(\mathcal{N})} P_{\mathbf{A}}(Z) \sum_{S \in \mathcal{S}(Z): i \in S} q_S(Z) \geq r_i, \forall i \in \mathcal{N}, \tag{3b}$$

$$\sum_{S \in \mathcal{S}(Z)} q_S(Z) = 1, \forall Z \in \mathcal{P}(\mathcal{N}), \tag{3c}$$

$$q_S(Z) \in [0, 1], \forall S \in \mathcal{S}(Z), \forall Z \in \mathcal{P}(\mathcal{N}). \tag{3d}$$

Suppose that an optimal solution to the above LP is $\mathbf{q}^* = [q_S^*(Z), \forall S \in \mathcal{S}(Z), \forall Z \in \mathcal{P}(\mathcal{N})]$. Then an optimal A-only policy α^* characterized by $\mathbf{q}^*$ obtains the maximum reward:

$$R^* \triangleq \sum_{Z \in \mathcal{P}(\mathcal{N})} P_{\mathbf{A}}(Z) \sum_{S \in \mathcal{S}(Z)} q_S^*(Z) \sum_{i \in S} w_i \mu_i. \tag{4}$$

However, the mean reward vector $\boldsymbol{\mu}$ is unknown to the player in advance. Hence, the player not only needs to maximize the reward based on the estimated mean rewards (i.e., exploitation), but she also has to simultaneously learn to obtain a more accurate estimate of the mean rewards (i.e., exploration). Such a learning process typically incurs a loss in the obtained reward, which is called the *regret*. Formally, the time-average regret of a policy π for a time horizon of T rounds, denoted by $R_\pi(T)$, is defined as the difference between the maximum reward R^* and the expected time-average reward obtained under policy π that chooses super arm $S(t)$ in round t, i.e.,

$$R_\pi(T) \triangleq R^* - \mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1}\sum_{i\in S(t)} w_i X_i(t)\right]. \tag{5}$$

Note that minimizing the regret is equivalent to maximizing the reward. Hence, the regret is a commonly used metric in the MAB literature for measuring the performance of learning algorithms. In this paper, we will adopt the time-average regret defined in (5) as the main performance metric.

The key notations used in this paper are listed in Table I.

IV. The LFG Algorithm

In this section, by carefully integrating the key ideas of UCB [6], [8] and the virtual queue technique [28], we develop a new algorithm, called *Learning with Fairness Guarantee (LFG)*, to tackle the CSMAB-F problem. While UCB is extended to deal with the exploitation-exploration tradeoff, the virtual queue technique is employed to handle the fairness constraints.

There are two main challenges in designing an efficient algorithm for the CSMAB-F problem: (i) how to maximize the reward in the face of unknown mean rewards and (ii) how to satisfy the fairness constraints. Note that these two challenges cannot be addressed separately as they are tightly coupled together. Therefore, we need a holistic approach to manage the balance between maximizing the reward and satisfying the fairness constraints. In what follows, we will first discuss the key ideas for addressing each individual challenge and then propose the LFG algorithm by carefully integrating them.

The key of maximizing the reward with uncertainty is to strike a balance between exploitation (i.e., choosing the option that gave highest rewards in the past) and exploration (i.e., seeking new options that might give higher rewards in the future). We extend a simple UCB policy based on the concept of optimism in the face of uncertainty to address this challenge and describe the details as follows.

Let $h_i(t)$ be the number of times arm i has been played by the end of round t, i.e., $h_i(t) \triangleq \sum_{k=0}^{t} d_i(k)$. We set $h_i(-1) = 0$ as the system begins at $t = 0$. Also, let $\hat{\mu}_i(t)$ be the sample mean of the observed rewards of arm i by the end of round t, i.e., $\hat{\mu}_i(t) \triangleq \frac{\sum_{k=0}^{t} X_i(k)d_i(k)}{h_i(t)}$. We set $\hat{\mu}_i(t) = 1$ if arm i has not been played yet by the end of round t (i.e., if $h_i(t) = 0$). We use $\bar{\mu}_i(t)$ to denote the UCB estimate of arm i in round t, which is given as follows:

$$\bar{\mu}_i(t) \triangleq \min\left\{\hat{\mu}_i(t-1) + \sqrt{\frac{3\log t}{2h_i(t-1)}}, 1\right\}, \tag{6}$$

where $\hat{\mu}_i(t-1)$ and $\sqrt{\frac{3\log t}{2h_i(t-1)}}$ correspond to exploitation and exploration, respectively. We use the above truncated version of the UCB estimate (i.e., capped at 1) as the actual reward must be in $[0, 1]$. Similarly, we set $\bar{\mu}_i(t) = 1$ if $h_i(t-1) = 0$.

TABLE I
Summary of Key Notations

Notations	Meaning
$\mathcal{N}$; N	Set of arms; number of arms
$\mathcal{P}(\mathcal{N})$	Power set of $\mathcal{N}$
T	Time horizon
m	Maximum number of simultaneously played arms
μ_i	Mean reward of arm i
w_i	Weight of arm i
r_i	Required minimum selection fraction for arm i
$X_i(t)$	Reward of arm i in round t
$\hat{\mu}_i(t)$	Sample mean of the observed reward of arm i up to round t
$\bar{\mu}_i(t)$	UCB estimate of arm i in round t
$h_i(t)$	Number of times arm i has been played up to round t
$d_i(t)$	Indicator of whether arm i is played or not in round t
$Q_i(t)$	Virtual queue length for arm i in round t
$A(t)$	Set of available arms in round t
$S(t)$	Super arm played in round t
$P_{\mathbf{A}}(Z)$	Probability that the set of available arms is Z
$\mathcal{S}(Z)$	Set of feasible super arms when observing available arms Z
$q_S(Z)$	Probability that an A-only policy α chooses super arm S when observing available arms Z
$\mathcal{C}$	Maximal feasibility region
R^*	Maximum reward with *a priori* knowledge of $\boldsymbol{\mu}$
$R_\pi(T)$	Time-average regret of policy π

In the basic MAB setting, the classic UCB policy simply selects the arm that has the largest UCB estimate in each round [6], [8]. However, in the CSMAB-F setting we are faced with several new challenges introduced by combinatorial arms, availability of arms, and fairness constraints. In particular, integrating fairness constraints adds a new layer of difficulty to the combinatorial sleeping MAB problem that is already quite challenging. This is because not only the player is faced with the exploitation-exploration dilemma when attempting to maximize the reward, but she also encounters a new tradeoff between maximizing the reward and satisfying the fairness requirement. Therefore, directly applying the UCB policy will not work as it was designed without fairness constraints in mind. Next, we will explain how to use the virtual queue technique to properly handle the fairness constraints, as well as how to cohesively integrate it with UCB to address the overall challenge of the CSMAB-F problem.

Following the framework developed in [28], we create a virtual queue Q_i for each arm i to handle the fairness constraints in (1). By slightly abusing the notation, we also use $Q_i(t)$ to denote the queue length of Q_i at the beginning of round t, which is a counter that keeps track of the "debt" to arm i up to round t. Specifically, the virtual queue length $Q_i(t)$ evolves according to the following dynamics:

$$Q_i(t) = [Q_i(t-1) + r_i - d_i(t-1)]^+, \tag{7}$$

where $[x]^+ \triangleq \max\{x, 0\}$. We set $Q_i(0) = 0$ as the system begins at $t = 0$. As can be seen in the above queue-length evolution, the "debt" to arm i increases by r_i in each round as r_i is the minimum selection fraction, and it decreases by one if arm i is selected in round $t-1$ (i.e., $d_i(t-1) = 1$).

Having introduced the UCB estimate and the virtual queues, we are now ready to describe the proposed LFG algorithm, which is presented in Algorithm 1. At the very beginning, we initialize $h_i(-1) = 0$ and $Q_i(0) = 0$ for all arms $i \in \mathcal{N}$ (lines 1-3). In each round t, we first update the UCB estimates $\bar{\mu}_i(t)$ and the virtual queue lengths $Q_i(t)$ according to (6) and (7) for all arms $i \in \mathcal{N}$, respectively, based on the decision and the feedback from the previous rounds (lines 4-11); we set $\bar{\mu}_i(t) = 1$ if $h_i(t-1) = 0$. Then, we observe the set of available arms $A(t)$ (line 12) and select a super arm $S(t) \in \mathcal{S}(A(t))$ that maximizes the compound value of the updated $\bar{\mu}_i(t)$ and $Q_i(t)$ as follows (line 13):

$$S(t) \in \operatorname*{argmax}_{S \in \mathcal{S}(A(t))} \sum_{i \in S} \left(Q_i(t) + \eta w_i \bar{\mu}_i(t) \right), \tag{8}$$

where η is a positive parameter we can tune to manage the balance between the reward and the virtual queue lengths. Note that the size of $\mathcal{S}(A(t))$ is exponential in m. Hence, the complexity of selecting a super arm $S(t)$ according to (8) could be prohibitively high in general. However, thanks to the special structure of linear compound reward, we can efficiently solve (8) and find a best super arm $S(t)$ by iteratively selecting best individual arms. Specifically, we select a super arm $S(t)$ consisting of the top-m^* arms in $A(t)$, where $m^* \triangleq \min\{m, |A(t)|\}$. That is, starting with an empty $S(t)$, we iteratively select arm i^* such that

$$i^* \in \operatorname*{argmax}_{i \in A(t) \setminus S(t)} Q_i(t) + \eta w_i \bar{\mu}_i, \tag{9}$$

and after each iteration, we update super arm $S(t)$ by adding arm i^* to it, i.e., $S(t) = S(t) \cup \{i^*\}$. Repeating the above procedure for m^* iterations solves (8) and finds a best super arm $S(t)$. After we play arms in $S(t)$ and set vector $\mathbf{d}(t)$ accordingly (line 14), we observe the reward $X_i(t)$ for all played arms $i \in S(t)$ (lines 15-17) and update $h_i(t)$ and $\hat{\mu}_i(t)$ accordingly for all arms $i \in \mathcal{N}$ (lines 18-20).

Remark: As we mentioned earlier, we introduce a design parameter η to manage the balance between the reward and virtual queue lengths. When η is large, the LFG algorithm gives a higher priority to maximizing the reward compared to meeting the fairness constraints. This is because an arm with a large estimated reward (i.e., UCB estimate) will be favored, compared to another arm that has a small estimated reward but a large "debt" (i.e., virtual queue length). In contrast, when η is small, the LFG algorithm gives a higher priority to meeting the fairness constraints because an arm with a large virtual queue length will be favored even if it has a small estimated reward. Indeed, our simulation results presented in Section VII reveal an interesting tradeoff between the regret and the speed of convergence to a point satisfying the fairness constraints.

V. Main Results

In this section, we analyze the performance of our proposed LFG algorithm and present our main results. Specifically, we show that the LFG algorithm is feasibility-optimal (i.e., it can satisfy any feasible requirement of minimum selection fraction for each individual arm) in Section V-A and derive an upper bound on the time-average regret in Section V-B.

Algorithm 1 Learning with Fairness Guarantee (LFG)

```
 1: for $i \in \mathcal{N}$ do
 2:     Initialize $h_i(-1) = 0$ and $Q_i(0) = 0$;
 3: end for
    In each round $t$:
 4: for $i \in \mathcal{N}$ do
 5:     if $h_i(t-1) > 0$ then
 6:         Update $\bar{\mu}_i(t)$ according to (6);
 7:     else
 8:         Set $\bar{\mu}_i(t) = 1$;
 9:     end if
10:     Update $Q_i(t)$ according to (7);
11: end for
12: Observe the set of available arms $A(t)$;
13: Select super arm $S(t)$ according to (8);
14: Play arms in $S(t)$ and set vector $\mathbf{d}(t)$ accordingly;
15: for $i \in S(t)$ do
16:     Observe the reward $X_i(t)$;
17: end for
18: for $i \in \mathcal{N}$ do
19:     Update $h_i(t)$ and $\hat{\mu}_i(t)$ according to $d_i(t)$ and $X_i(t)$.
20: end for
```

A. Feasibility Optimality

We first present the feasibility-optimality result. That is, the LFG algorithm can satisfy the fairness constraints in (1) for any minimum selection fraction vector $\mathbf{r}$ strictly inside the maximal feasibility region $\mathcal{C}$.

Note that the constraints in (1) are satisfied as long as the virtual queue system defined in (7) is *mean rate stable* [28, pp. 56-57], i.e., $\lim_{T \to \infty} \frac{\mathbb{E}[\sum_{i=1}^{N} Q_i(T)]}{T} = 0$. In our virtual queue system, mean rate stability is implied by a stronger notion called *strong stability*, i.e., $\limsup_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\sum_{i=1}^{N} Q_i(t)] < \infty$. Therefore, in order to prove feasibility-optimality, it is sufficient to show that the virtual queue system is strongly stable whenever the minimum selection fraction vector $\mathbf{r}$ is strictly inside $\mathcal{C}$. We state this result in Theorem 1.

Theorem 1. *The LFG algorithm is feasibility-optimal. Specifically, for any minimum selection fraction* $\mathbf{r}$ *strictly inside the maximal feasibility region* $\mathcal{C}$*, the virtual queue system defined in* (7) *is strongly stable under LFG. That is,*

$$\limsup_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left[\sum_{i=1}^{N} Q_i(t) \right] \le \frac{B}{\epsilon} < \infty, \tag{10}$$

where $B \triangleq \frac{N}{2} + \eta m w_{\max}$ *and* ϵ *is some positive constant satisfying that* $\mathbf{r} + \epsilon \mathbf{1}$ *is still strictly inside* $\mathcal{C}$*, with* $\mathbf{1}$ *being the* N*-dimensional vector of all ones.*

We prove Theorem 1 by using standard Lyapunov-drift analysis [28]. The detailed proof is provided in Appendix A.

Remark: Note that the work of [22] also studies an MAB problem with minimum-guarantee constraints. However, their work differs significantly from ours because their considered minimum guarantee is for the total rewards (of some type/level) rather than for each individual arm, i.e., fairness among arms is not modeled. More importantly, the proposed learning algorithm in [22] may violate the constraints. Although they show that the violations are upper bounded by $O(T^{5/6})$, this upper bound implies that the constraints may not be satisfied even after a long enough time. In stark contrast, Theorem 1 states that our proposed LFG algorithm can satisfy the (long-term) fairness constraints as long as the requirement is feasible. Another difference is that they do not consider sleeping bandits, which can further complicate the problem.

B. Upper Bound on Regret

In this subsection, we prove an upper bound on the time-average regret (as defined in (5)) under the LFG algorithm. This upper bound is achieved uniformly over time (i.e., for any finite time horizon T) rather than asymptotically when T goes to infinity. We state this result in Theorem 2.

Theorem 2. *Under the LFG algorithm, the time-average regret defined in* (5) *has the following upper bound:*

$$R_{LFG}(T) \leq \frac{N}{2\eta} + \frac{\beta_1\sqrt{mNT\log T} + \beta_2 N}{T}, \tag{11}$$

where $\beta_1 \triangleq 2\sqrt{6}w_{\max}$, *and* $\beta_2 \triangleq 4w_{\max}$.

We prove Theorem 2 by using a similar line of regret analysis in [27]. The detailed proof is provided in Appendix B.

Remark: The derived regret upper bound in (11) is quite appealing as it separately captures the impact of the fairness constraints and the impact of the uncertainty in the mean rewards for any finite time horizon T. Note that the regret upper bound in (11) has two terms. The first term $\frac{N}{2\eta}$ is inversely proportional to η and is attributed to the impact of the fairness constraints. Specifically, when η is small, the LFG algorithm gives a higher priority to meeting the fairness requirement by favoring an arm with a larger "debt" (i.e., virtual queue length) as in (9), even if this arm has a small estimated reward. This results in a larger regret captured in the first term. Similarly, a larger η leads to a smaller regret captured in the first term, but it will take longer for the LFG algorithm to converge to a point satisfying the fairness constraints. This interesting tradeoff can also be observed from our simulation results in Section VII. The second term $\frac{\beta_1\sqrt{mNT\log T}+\beta_2 N}{T}$ is of the order $O(\sqrt{\log T/T})$. This part of the regret corresponds to the notion of regret in typical MAB problems and is attributed to the cost that needs to be paid in the learning/exploration process. Note that the second term is an *instance-independent* upper bound that does not depend on the problem-specific parameter $\boldsymbol{\mu}$. Our derived bound on the time-average regret is consistent with the instance-independent result for basic MAB problems [1, Ch. 2.4.3][1].

VI. Applications

In this section, we provide more detailed discussions about real-world applications of our proposed CSMAB-F framework. Specifically, we will discuss the following three applications as examples: scheduling of real-time traffic in wireless networks [4], ad placement in online advertising systems [29], and task assignment in crowdsourcing platforms [30].

A. Scheduling of Real-time Traffic in Wireless Networks

Consider the problem of scheduling real-time traffic with QoS constraints in a single-hop wireless network. Assume that there are N clients competing for a shared wireless channel to transmit packets to a common AP (see, e.g., [4]). Time is slotted. The AP decides which client(s) can transmit at what times. Consider a scheduling cycle, called a frame, that consists of m consecutive time slots. Every client generates one data packet at the beginning of each frame. To avoid interference, we assume that at most one client can transmit in each time slot. Note that some clients may sometimes be unable to transmit when experiencing poor channel conditions (due to fading or mobility). Assume that the channel conditions remain unchanged during a frame but may vary over frames and that the AP obtains the exact knowledge about the channel conditions via probing. At the beginning of each frame, the AP makes scheduling decisions by selecting an available client to transmit in each of the m time slots; at the beginning of each time slot, the AP broadcasts a control packet that announces the scheduling decision, and then, the selected client transmits a packet to the AP in that time slot. We model real-time traffic by assuming that packets have a lifetime of m time slots and expire at the end of the frame. The above framework is illustrated in Fig. 1. While a successfully delivered packet will generate a utility, which could represent the value of the information contained in the packet, an expired packet will be dropped at the end of the frame. We assume that the utility corresponding to each client is a random variable, and its mean is unknown *a priori*. There is a weight associated with each client, indicating the importance of the information provided by the client.

The goal of the AP is to maximize the cumulative utilities by scheduling packet transmissions in the face of unknown mean utilities. In addition, each client has a QoS requirement that a minimum delivery ratio must be guaranteed. Clearly, the scheduling problem with minimum delivery ratio guarantee can naturally be formulated as a CSMAB-F problem.

B. Ad Placement in Online Advertising Systems

Online advertising has emerged as a very popular Internet application [29]. Take a page of *Weather.com* website shown in Fig. 2 for example. When an Internet user visits the webpage, the publisher dynamically chooses multiple ads from the ads pool to display in the ad-mix areas (highlighted by red circles in Fig. 2). We assume that the ads pool consists of N ads, and the ad-mix area has a limited capacity, which allows displaying no more than m ads simultaneously. Note that some ads are irrelevant to certain users, depending on the context

[1] Time-average regret $O(\sqrt{\log T/T})$ vs. cumulative regret $O(\sqrt{T\log T})$.

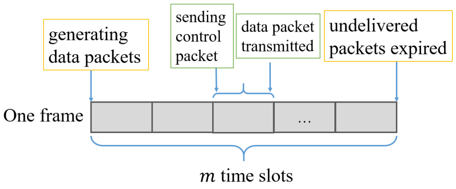


Fig. 1. Scheduling of real-time traffic

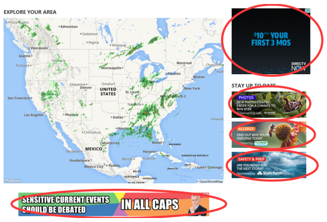


Fig. 2. Ad placement

amazon FLEX
amazon mechanical turk
testlio
crowdsourcedtesting
Tasks
Online platform
Workers

Fig. 3. Task assignment in crowdsourcing

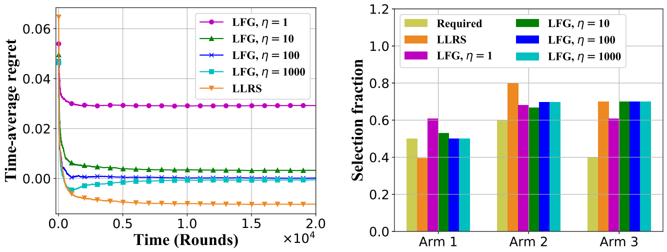
including users' characteristics (gender, interest, location, etc.) and content of the webpage. Hence, such irrelevant ads can be viewed as unavailable to those users, and the availability of ads depends on the distribution of the context. After seeing a displayed ad, the user may or may not click it. The click-through rate (i.e., the rate at which the ad is clicked) of each ad is unknown *a priori*. Each click of an ad will potentially generate a revenue for the advertiser, which can be viewed as the weight of the ad.

The goal of the ad publisher is to maximize the cumulative revenues by determining a best subset of ads to display in the face of unknown click-through rates. In addition, the publisher must guarantee a minimum display frequency for advertisers who pay a fixed cost over a specified period, regardless of users' responses to the displayed ads. Obviously, the ad placement problem with minimum display frequency guarantee fits perfectly into our proposed CSMAB-F framework.

C. Task Assignment in Crowdsourcing Platforms

The increasing application of crowdsourcing is significantly changing the way people conduct business and many other activities [30]. Consider a crowdsourcing platform such as Amazon Mechanical Turk, Amazon Flex (for package delivery), and Testlio (for software testing), as shown in Fig. 3. Tasks arriving to the crowdsourcing platform will be assigned to a group of workers with different unknown skill levels. Specifically, when a task arrives, the platform may divide the task into multiple sub-tasks; then the sub-tasks will be assigned to no more than m workers from a pool of N workers, due to the number of sub-tasks or a limited budget. Note that some workers could be unavailable to take certain tasks due to various reasons (time conflicts, location constraints, limited skills, preferences, etc.). Each completed task will generate a payoff that depends on the quality or efficiency of the workers. The payoff is a random variable, and its mean is unknown *a priori* due to unknown skill levels of workers.

The goal of the crowdsourcing platform is to maximize the cumulative payoffs by determining an optimal task allocation in the face of unknown mean payoffs. In addition, the platform has to take fairness towards workers into account through a minimum assignment ratio guarantee for each worker. This fairness guarantee helps maintain a healthy and sustainable platform with improved worker satisfaction and higher worker participation. Apparently, our proposed CSMAB-F framework can be applied to address the task assignment problem with minimum assignment ratio guarantee.



(a) Time-average Regret

(b) Selection fraction

Fig. 4. Performance comparisons of different algorithms

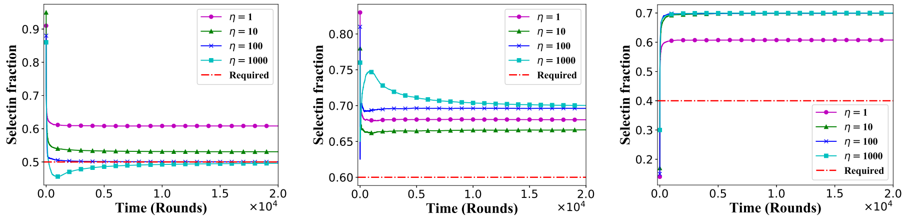
VII. Numerical Results

In this section, we conduct simulations to evaluate the performance of our proposed LFG algorithm and discuss several interesting observations based on the simulation results.

We consider two scenarios for the simulations: (i) $N = 3$ and $m = 2$; (ii) $N = 10$ and $m = 6$. Since the observations are similar for these two scenarios, we will focus on the discussion about the first scenario due to space limitations. We assume that the availability of arm i is a binary random variable that is *i.i.d.* over time with mean p_i. Then, the distribution of available arms can be computed as $P_{\mathbf{A}}(Z) = \prod_{i \in Z} p_i \prod_{i \notin Z} (1 - p_i)$ for all $Z \in \mathcal{P}(\mathcal{N})$. We also assume binary rewards with the same unit weight (i.e., $w_i = 1$) for all the arms. The detailed setting of other parameters is as follows: $\boldsymbol{\mu} = (0.4, 0.5, 0.7)$, $\mathbf{r} = (0.5, 0.6, 0.4)$, and $\mathbf{p} = (p_1, p_2, p_3) = (0.9, 0.8, 0.7)$.

First, in order to demonstrate that LFG can effectively meet the fairness requirement, we compare LFG with a fairness-oblivious combinatorial MAB algorithm, called *Learning with Linear Rewards (LLR)* [10]. We modify the LLR algorithm to accommodate sleeping bandits; the modified version is called *LLR for Sleeping bandits (LLRS)*. In each round t, observing the set of available arms $A(t)$, LLRS selects a super arm $S(t)$ that has the largest weighted sum of the UCB estimates among all the feasible super arms in $\mathcal{S}(A(t))$, i.e., $S(t) \in \text{argmax}_{S \in \mathcal{S}(A(t))} \sum_{i \in S} w_i \bar{\mu}_i(t)$. Note that LLRS is oblivious of the fairness constraints in (1).

We simulate LFG with $\eta \in \{1, 10, 100, 1000\}$ and LLRS for $T = 2 \times 10^4$ rounds (at which all the considered algorithms are observed to converge) and present the results in Fig. 4. Fig. 4(a) shows the time-average regret over time for the considered algorithms; Fig. 4(b) shows the selection fraction of each arm at the end of the simulation (i.e., at $T = 2 \times 10^4$).



(a) Arm 1

(b) Arm 2

(c) Arm 3

Fig. 5. Selection fraction over time under LFG with different values of η

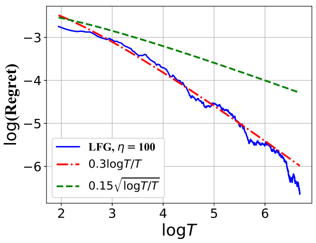


Fig. 6. Regret vs. T

From Fig. 4(a), we can make the following observations: (i) LFG with a larger η results in a smaller regret, and LFG with $\eta \geq 100$ approaches a zero regret; (ii) LLRS achieves the smallest regret, which is even negative (i.e., it achieves a reward larger than the optimal R^*). Observation (i) is expected, as we explained in Section V-B: the upper bound on regret in (11) approaches zero when both η and T become large. Observation (ii) is not surprising because LLRS is fairness-oblivious and may produce an infeasible solution. Indeed, Fig. 4(b) shows that Arm 1's selection fraction under LLRS is smaller than the required value (0.4 vs. 0.5). This is because Arm 1 has the smallest mean reward and is not favored under LLRS, which is unaware of the fairness contraints. On the other hand, Fig. 4(b) also shows that with different values of η, LFG consistently satisfies the required minimum selection fraction, which verifies our theoretical result on feasibility-optimality of LFG (Theorem 1).

At first glance, the above observations seem to suggest that LFG with a large η is desirable because that leads to a vanishing regret while still providing fairness guarantee. However, what is missing here is the speed of convergence to a point satisfying the fairness requirement, which is another critical design concern in practice. To understand the convergence speed of LFG with different values of η, in Fig. 5 we plot the selection fraction over time for each arm. Taking Fig. 5(a) for example, we can observe that the convergence slows down as η increases. In addition, before LFG with $\eta = 1000$ converges (e.g., when $T \leq 10^4$), the actual selection fraction of Arm 1 does not meet the required minimum value of 0.5. Since the constraints may be temporarily violated, the regret could even be negative before LFG converges (see $\eta = 1000$ in Fig. 4(a)). Therefore, the simulation results reveal an interesting tradeoff between the regret and the convergence speed. We can control and optimize this tradeoff by tuning η. For example, for the considered scenario, LFG with $\eta = 100$ seems to achieve a good balance between the regret and the convergence speed.

Finally, we want to investigate the tightness of the upper bound derived in (11). Consider the average of 100 independent simulation runs for LFG with $\eta = 100$. Fig. 6 shows the time-average regret vs. the time horizon T in a log-log plot. Recall that the upper bound in (11) has two terms. The impact of T appears in the second term that is of the order $\sqrt{\log T/T}$. When T becomes large, it becomes difficult to see the impact of T on the regret as the first term $\frac{N}{2\eta}$ becomes dominant. Therefore, we consider the region with $T \leq 1000$ (i.e., $\log T \leq 6.9$). Fig. 6 seems to suggest that the time-average regret follows the order $\log T/T$ rather than $\sqrt{\log T/T}$. This implies that the upper bound in (11) is not tight. One reason could be that the $\sqrt{\log T/T}$ bound is instance-independent. It remains open whether one can come up with novel analytical techniques to derive a better bound of $\log T/T$.

VIII. Conclusion

In this paper, we proposed a unified CSMAB-F framework that integrates several critical factors (i.e., combinatorial actions, availability of actions, and fairness) of the system in many real-world applications. In particular, no prior work has studied MAB problems with fairness constraints on a minimum selection fraction for each individual arm. To address the new challenges introduced by modeling these factors, we developed a new LFG algorithm that achieves a provable regret upper bound while effectively providing fairness guarantee.

We leave the following interesting questions to our future work: Can one prove a tighter upper bound on regret? How to develop efficient algorithms for a more general model that potentially accounts for nonlinear reward functions, more sophisticated combinatorial structures (e.g., matroids), and more general fairness criteria other than temporal fairness that we consider in this paper?

References

[1] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Foundations and Trends® in Machine Learning*, vol. 5, no. 1, pp. 1–122, 2012.

[2] T. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.

[3] X. Liu, E. K. Chong, and N. B. Shroff, "A framework for opportunistic scheduling in wireless networks," *Computer networks*, vol. 41, no. 4, pp. 451–474, 2003.

[4] I. H. Hou, V. Borkar, and P. R. Kumar, "A theory of qos for wireless," in *Proceedings of IEEE INFOCOM*, April 2009, pp. 486–494.

[5] T. Lan, D. Kao, M. Chiang, and A. Sabharwal, "An axiomatic theory of fairness in network resource allocation," in *2010 Proceedings IEEE INFOCOM*, March 2010, pp. 1–9.

[6] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in applied mathematics*, vol. 6, no. 1, pp. 4–22, 1985.

[7] J. Gittins, K. Glazebrook, and R. Weber, *Multi-armed bandit allocation indices*. John Wiley & Sons, 2011.

[8] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, no. 2-3, pp. 235–256, 2002.

[9] V. Anantharam, P. Varaiya, and J. Walrand, "Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays-part i: Iid rewards," *IEEE Transactions on Automatic Control*, vol. 32, no. 11, pp. 968–976, 1987.

[10] Y. Gai, B. Krishnamachari, and R. Jain, "Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations," *IEEE/ACM Transactions on Networking (TON)*, vol. 20, no. 5, pp. 1466–1478, 2012.

[11] B. Kveton, Z. Wen, A. Ashkan, H. Eydgahi, and B. Eriksson, "Matroid bandits: Fast combinatorial optimization with learning," in *Proceedings of UAI*, 2014.

[12] W. Chen, Y. Wang, and Y. Yuan, "Combinatorial multi-armed bandit: General framework and applications," in *International Conference on Machine Learning*, 2013, pp. 151–159.

[13] W. Chen, W. Hu, F. Li, J. Li, Y. Liu, and P. Lu, "Combinatorial multi-armed bandit with general reward functions," in *Advances in Neural Information Processing Systems*, 2016, pp. 1659–1667.

[14] R. Combes, M. S. T. M. Shahi, A. Proutiere *et al.*, "Combinatorial bandits revisited," in *Advances in Neural Information Processing Systems*, 2015, pp. 2116–2124.

[15] R. Kleinberg, A. Niculescu-Mizil, and Y. Sharma, "Regret bounds for sleeping experts and bandits," *Machine learning*, vol. 80, no. 2-3, pp. 245–272, 2010.

[16] V. Kanade, H. B. McMahan, and B. Bryan, "Sleeping experts and bandits with stochastic action availability and adversarial rewards," in *Artificial Intelligence and Statistics*, 2009, pp. 272–279.

[17] V. Kanade and T. Steinke, "Learning hurdles for sleeping experts," *ACM Transactions on Computation Theory (TOCT)*, vol. 6, no. 3, p. 11, 2014.

[18] A. Chatterjee, G. Ghalme, S. Jain, R. Vaish, and Y. Narahari, "Analysis of thompson sampling for stochastic sleeping bandits." in *UAI*, 2017.

[19] R. Combes, C. Jiang, and R. Srikant, "Bandits with budgets: Regret lower bounds and optimal algorithms," *ACM SIGMETRICS Performance Evaluation Review*, vol. 43, no. 1, pp. 245–257, 2015.

[20] A. Badanidiyuru, R. Kleinberg, and A. Slivkins, "Bandits with knapsacks," *Journal of the ACM (JACM)*, vol. 65, no. 3, pp. 13:1–13:55, 2018.

[21] E. V. Denardo, E. A. Feinberg, and U. G. Rothblum, "The multi-armed bandit, with constraints," *Annals of Operations Research*, vol. 208, no. 1, pp. 37–62, 2013.

[22] K. Cai, X. Liu, Y. J. Chen, and J. C. S. Lui, "An online learning approach to network application optimization with guarantee," in *Proceedings of IEEE INFOCOM*, 2018, in press.

[23] K. Chen, K. Cai, L. Huang, and J. Lui, "Beyond the click-through rate: Web link selection with multi-level feedback," *arXiv preprint arXiv:1805.01702*, 2018.

[24] M. Joseph, M. Kearns, J. H. Morgenstern, and A. Roth, "Fairness in learning: Classic and contextual bandits," in *Advances in Neural Information Processing Systems*, 2016, pp. 325–333.

[25] M. Joseph, M. Kearns, J. Morgenstern, S. Neel, and A. Roth, "Fair algorithms for infinite and contextual bandits," *arXiv preprint arXiv:1610.09559*, 2016.

[26] M. S. Talebi and A. Proutiere, "Learning proportionally fair allocations with low regret," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 2, no. 2, pp. 36:1–36:31, 2018.

[27] W.-K. Hsu, J. Xu, X. Lin, and M. R. Bell, "Integrate learning and control in queueing systems with uncertain payoffs," Purdue University, available at https://engineering.purdue.edu/%7elinx/papers.html, Tech. Rep., 2018.

[28] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.

[29] "Adspeed ad server," https://www.adspeed.com/, 2018, [Accessed: 2018-06-30].

[30] F. Basık, B. Gedik, H. Ferhatosmanoglu, and K.-L. Wu, "Fair task allocation in crowdsourced delivery," *IEEE Transactions on Services Computing*, 2018.

Appendix

A. Proof of Theorem 1

Proof. Consider the LFG algorithm. To prove feasibility optimality, we want to show that for any vector $\mathbf{r}$ strictly inside the maximal feasibility region $\mathcal{C}$, the minimum selection fraction requirements (i.e., Eq. (1)) are satisfied. Note that the requirements of (1) are satisfied as long as the virtual queue system defined in (7) is *mean rate stable* [28, pp. 56-57], i.e., $\lim_{T\to\infty} \frac{\mathbb{E}[\sum_{i=1}^{N} Q_i(T)]}{T} = 0$. In our virtual queue system, mean rate stability is implied by a stronger notion called *strong stability*, i.e., $\limsup_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}[\sum_{i=1}^{N} Q_i(t)] < \infty$. Therefore, it is sufficient to show that the virtual queue system is strongly stable for any vector $\mathbf{r}$ strictly inside $\mathcal{C}$.

We proceed the proof using the Lyapunov-drift analysis [28]. Let $\mathbf{Q}(t) = (Q_1(t), \ldots, Q_N(t))$ be the queue-length vector in round t. Consider the following Lyapunov function:

$$L(\mathbf{Q}(t)) \triangleq \frac{1}{2}\sum_{i=1}^{N} Q_i^2(t). \tag{12}$$

The drift of the Lyapunov function is given by

$$\begin{aligned}
&L(\mathbf{Q}(t+1)) - L(\mathbf{Q}(t)) \\
&= \frac{1}{2}\sum_{i=1}^{N} Q_i^2(t+1) - \frac{1}{2}\sum_{i=1}^{N} Q_i^2(t) \\
&\overset{(a)}{\leq} \frac{1}{2}\sum_{i=1}^{N}(Q_i(t) + r_i - d_i(t))^2 - \frac{1}{2}\sum_{i=1}^{N} Q_i^2(t) \\
&= \frac{1}{2}\sum_{i=1}^{N}(r_i - d_i(t))^2 + \sum_{i=1}^{N}(r_i - d_i(t))Q_i(t) \\
&\overset{(b)}{\leq} \frac{N}{2} + \sum_{i=1}^{N} r_i Q_i(t) - \sum_{i=1}^{N} d_i(t) Q_i(t),
\end{aligned} \tag{13}$$

where (a) is from the queue-length evolution (7) and (b) holds because both r_i and $d_i(t)$ are within $[0, 1]$. Taking conditional expectation of both sides of the above gives

$$\begin{aligned}
&\mathbb{E}[L(\mathbf{Q}(t+1)) - L(\mathbf{Q}(t)) | \mathbf{Q}(t)] \\
&\leq \frac{N}{2} + \sum_{i=1}^{N} r_i Q_i(t) - \mathbb{E}\left[\sum_{i=1}^{N} d_i(t) Q_i(t) | \mathbf{Q}(t)\right] \\
&= \frac{N}{2} + \sum_{i=1}^{N} r_i Q_i(t) - \mathbb{E}\left[\sum_{i\in S(t)} d_i(t) Q_i(t) | \mathbf{Q}(t)\right] \\
&= \frac{N}{2} + \sum_{i=1}^{N} r_i Q_i(t) + \mathbb{E}\left[\sum_{i\in S(t)} \eta w_i \bar{\mu}_i(t) | \mathbf{Q}(t)\right] \\
&\quad - \mathbb{E}\left[\sum_{i\in S(t)} (Q_i(t) + \eta w_i \bar{\mu}_i(t)) | \mathbf{Q}(t)\right] \\
&\leq \frac{N}{2} + \sum_{i=1}^{N} r_i Q_i(t) + \eta m w_{\max} \\
&\quad - \mathbb{E}\left[\sum_{i\in S(t)} (Q_i(t) + \eta w_i \bar{\mu}_i(t)) | \mathbf{Q}(t)\right] \\
&= B + \sum_{i=1}^{N} r_i Q_i(t) \\
&\quad - \mathbb{E}\left[\sum_{i\in S(t)} (Q_i(t) + \eta w_i \bar{\mu}_i(t)) | \mathbf{Q}(t)\right],
\end{aligned} \tag{14}$$

where the last inequality holds because $w_i \leq w_{\max}$, $\bar{\mu}_i(t) \leq 1$, and $|S(t)| \leq m$, and $B \triangleq \frac{N}{2} + \eta m w_{\max}$ is a constant.

Recall that $\mathbf{r}$ is strictly inside $\mathcal{C}$. Then, there must exist some $\epsilon > 0$ such that $\mathbf{r}+\epsilon\mathbf{1}$ is also strictly inside $\mathcal{C}$, where $\mathbf{1}$ denotes the N-dimensional all-ones vector. By Lemma 1, there exists an A-only policy α that can support vector $\mathbf{r} + \epsilon\mathbf{1}$. That is,

$$\sum_{Z\in\mathcal{P}(\mathcal{N})} P_{\mathbf{A}}(Z) \sum_{S\in\mathcal{S}(Z):i\in S} q_S^\alpha(Z) \geq r_i + \epsilon,\ \forall i \in \mathcal{N}, \quad (15)$$

where $\mathbf{q}^\alpha = [q_S^\alpha(Z), \forall S \in \mathcal{S}(Z), \forall Z \in \mathcal{P}(\mathcal{N})]$ is the group of probability distributions associated with policy α. Recall that in each round t, policy α observes available arms $A(t)$ and chooses a super arm $S^\alpha(t) \in \mathcal{S}(A(t))$ independent of $\mathbf{Q}(t)$. Then, the last term of the right-hand side of (14) satisfies

$$\begin{aligned}
&\mathbb{E}\left[\sum_{i\in S(t)} (Q_i(t) + \eta w_i \bar{\mu}_i(t))|\mathbf{Q}(t)\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\sum_{i\in S(t)} (Q_i(t) + \eta w_i \bar{\mu}_i(t))|\mathbf{Q}(t), A(t)\right]\right] \\
&\overset{(a)}{\geq} \mathbb{E}\left[\mathbb{E}\left[\sum_{i\in S^\alpha(t)} (Q_i(t) + \eta w_i \bar{\mu}_i(t))|\mathbf{Q}(t), A(t)\right]\right] \\
&\geq \mathbb{E}\left[\mathbb{E}\left[\sum_{i\in S^\alpha(t)} Q_i(t)|\mathbf{Q}(t), A(t)\right]\right] \\
&\overset{(b)}{=} \mathbb{E}\left[\mathbb{E}\left[\sum_{i\in S^\alpha(t)} Q_i(t)|A(t)\right]\right] \\
&\overset{(c)}{=} \mathbb{E}\left[\sum_{S\in\mathcal{S}(A(t))} q_S^\alpha(A(t)) \sum_{i\in S} Q_i(t)\right] \\
&= \sum_{Z\in\mathcal{P}(\mathcal{N})} P_{\mathbf{A}}(Z) \sum_{S\in\mathcal{S}(Z)} q_S^\alpha(Z) \sum_{i\in S} Q_i(t) \\
&= \sum_{i=1}^N Q_i(t) \sum_{Z\in\mathcal{P}(\mathcal{N})} P_{\mathbf{A}}(Z) \sum_{S\in\mathcal{S}(Z):i\in S} q_S^\alpha(Z),
\end{aligned} \quad (16)$$

where (a) is due to the operations of LFG (specifically, (8)), (b) holds because policy α's decision is independent of $\mathbf{Q}(t)$, and (c) is due to the operations of policy α.

Substituting (16) into (14) and applying (15) give

$$\begin{aligned}
&\mathbb{E}[L(\mathbf{Q}(t+1)) - L(\mathbf{Q}(t))|\mathbf{Q}(t)] \\
&\leq B + \sum_{i=1}^N r_i Q_i(t) \\
&\quad - \sum_{i=1}^N Q_i(t) \sum_{Z\in\mathcal{P}(\mathcal{N})} P_{\mathbf{A}}(Z) \sum_{S\in\mathcal{S}(Z):i\in S} q_S^\alpha(Z) \\
&\leq B + \sum_{i=1}^N r_i Q_i(t) - \sum_{i=1}^N Q_i(t)(r_i + \epsilon) \\
&= B - \epsilon \sum_{i=1}^N Q_i(t).
\end{aligned} \quad (17)$$

Finally, invoking the Lyapunov Drift Theorem [28, Theorem 4.1] gives (10), which completes the proof. □

B. Proof of Theorem 2

Proof. Consider an optimal A-only policy α^* and its associated probability distributions $\mathbf{q}^* = [q_S^*(Z), \forall S \in \mathcal{S}(Z), \forall Z \in \mathcal{P}(\mathcal{N})]$. Let $S^*(t)$ be the super arm selected by policy α^* in round t. Vector $\mathbf{d}^*(t) = (d_1^*(t), \ldots, d_N^*(t))$ is the corresponding action vector. Due to (4), we have

$$\begin{aligned}
R^* &= \sum_{Z\in\mathcal{P}(\mathcal{N})} P_{\mathbf{A}}(Z) \sum_{S\in\mathcal{S}(Z)} q_S^*(Z) \sum_{i\in S} w_i \mu_i \\
&= \mathbb{E}\left[\sum_{i\in S^*(t)} w_i \mu_i\right].
\end{aligned} \quad (18)$$

Plugging (18) into (5), we can rewrite the regret of LFG as

$$\begin{aligned}
R_{\text{LFG}}(T) &= R^* - \frac{1}{T}\mathbb{E}\left[\sum_{t=0}^{T-1} \sum_{i\in S(t)} w_i X_i(t)\right] \\
&= \frac{1}{T}\sum_{t=0}^{T-1} \left\{R^* - \mathbb{E}\left[\sum_{i\in S(t)} w_i \mu_i\right]\right\} \\
&= \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[\underbrace{\sum_{i\in S^*(t)} w_i \mu_i - \sum_{i\in S(t)} w_i \mu_i}_{\Delta R(t)}\right].
\end{aligned} \quad (19)$$

We define the following quantity:

$$\begin{aligned}
\Delta R(t) &\triangleq \sum_{i\in S^*(t)} w_i \mu_i - \sum_{i\in S(t)} w_i \mu_i \\
&= \sum_{i=1}^N w_i \mu_i d_i^*(t) - \sum_{i=1}^N w_i \mu_i d_i(t),
\end{aligned} \quad (20)$$

which captures the gap between the expected rewards achieved by policy α^* and LFG in round t. Adding $\Delta R(t)$ scaled by η to the drift of the Lyapunov function (i.e., (13)) gives the drift-plus-regret:

$$\begin{aligned}
&L(\mathbf{Q}(t+1)) - L(\mathbf{Q}(t)) + \eta\Delta R(t) \\
&\leq \frac{N}{2} + \sum_{i=1}^N r_i Q_i(t) - \sum_{i=1}^N d_i(t) Q_i(t) \\
&\quad + \eta \sum_{i=1}^N w_i \mu_i d_i^*(t) - \eta \sum_{i=1}^N w_i \mu_i d_i(t) \\
&= \frac{N}{2} + \sum_{i=1}^N (Q_i(t) + \eta w_i \mu_i)(d_i^*(t) - d_i(t)) \\
&\quad + \sum_{i=1}^N Q_i(t)(r_i - d_i^*(t)).
\end{aligned} \quad (21)$$

We can bound the expected drift-plus-regret as

$$\begin{aligned}
&\mathbb{E}[L(\mathbf{Q}(t+1)) - L(\mathbf{Q}(t)) + \eta\Delta R(t)] \\
&\le \frac{N}{2} + \sum_{i=1}^{N} \mathbb{E}[(Q_i(t) + \eta w_i \mu_i)(d_i^*(t) - d_i(t))] \\
&\quad + \sum_{i=1}^{N} \mathbb{E}[Q_i(t)(r_i - d_i^*(t))] \\
&\le \frac{N}{2} + \mathbb{E}\left[\underbrace{\sum_{i=1}^{N} (Q_i(t) + \eta w_i \mu_i)(d_i^*(t) - d_i(t))}_{C_1(t)} \right],
\end{aligned} \tag{22}$$

where the last step follows from $\mathbb{E}[Q_i(t)d_i^*(t)] = \mathbb{E}[Q_i(t)]\mathbb{E}[d_i^*(t)]$ (due to the decision of policy α^* being independent of the queue length $Q_i(t)$) and $\mathbb{E}[d_i^*(t)] \ge r_i$ (because policy α^* is stationary and feasible). Define $C_1(t) \triangleq \sum_{i=1}^{N} (Q_i(t) + \eta w_i \mu_i)(d_i^*(t) - d_i(t))$. Summing (22) for all $t \in \{0, \dots, T-1\}$, using the trick of telescoping sum, and dividing both sides of the inequality by $T\eta$, we obtain

$$\begin{aligned}
&\frac{1}{T\eta}\mathbb{E}[L(\mathbf{Q}(T)) - L(\mathbf{Q}(0))] + \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\Delta R(t)] \\
&\le \frac{N}{2\eta} + \frac{1}{T\eta}\sum_{t=0}^{T-1}\mathbb{E}[C_1(t)].
\end{aligned} \tag{23}$$

Since $L(\mathbf{Q}(T)) \ge 0$ and $L(\mathbf{Q}(0)) = 0$, we have

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\Delta R(t)] \le \frac{N}{2\eta} + \frac{1}{T\eta}\sum_{t=0}^{T-1}\mathbb{E}\left[C_1(t)\right]. \tag{24}$$

In Appendix C, we will show the following bound:

$$\frac{1}{T\eta}\sum_{t=0}^{T-1}\mathbb{E}[C_1(t)] \le \frac{w_{\max}}{T}\left(2\sqrt{6mNT\log T} + 4N\right). \tag{25}$$

Finally, plugging (25) into (24) and combining it with (19) yield (11). This completes the proof of Theorem 2. □

C. Bounding $C_1(t)$

In this section, we want to show (25).

Consider a policy π', which, in each round t, chooses a super arm $S'(t)$ in the following manner:

$$S'(t) \in \operatorname*{argmax}_{S \in \mathcal{S}(A(t))} \sum_{i \in S}(Q_i(t) + \eta w_i \mu_i(t)). \tag{26}$$

Recall that in each round t, the LFG algorithm chooses a super arm $S(t)$ according to (8). Therefore, we have

$$\sum_{i \in S(t)} (Q_i(t) + \eta w_i \bar{\mu}_i(t)) \ge \sum_{i \in S'(t)} (Q_i(t) + \eta w_i \bar{\mu}_i(t)). \tag{27}$$

Next, we derive an upper bound on $C_1(t)$:

$$\begin{aligned}
C_1(t) &= \sum_{i=1}^{N} (Q_i(t) + \eta w_i \mu_i)(d_i^*(t) - d_i(t)) \\
&= \sum_{i \in S^*(t)} (Q_i(t) + \eta w_i \mu_i) - \sum_{i \in S(t)} (Q_i(t) + \eta w_i \mu_i) \\
&\overset{(a)}{\le} \sum_{i \in S'(t)} (Q_i(t) + \eta w_i \mu_i) - \sum_{i \in S(t)} (Q_i(t) + \eta w_i \mu_i) \\
&\overset{(b)}{\le} \sum_{i \in S'(t)} (Q_i(t) + \eta w_i \mu_i) - \sum_{i \in S(t)} (Q_i(t) + \eta w_i \mu_i) \\
&\quad + \sum_{i \in S(t)} (Q_i(t) + \eta w_i \bar{\mu}_i(t)) \\
&\quad - \sum_{i \in S'(t)} (Q_i(t) + \eta w_i \bar{\mu}_i(t)) \\
&= \eta \left(\underbrace{\sum_{i \in S(t)} w_i(\bar{\mu}_i(t) - \mu_i)}_{C_2(t)} + \underbrace{\sum_{i \in S'(t)} w_i(\mu_i - \bar{\mu}_i(t))}_{C_3(t)} \right),
\end{aligned} \tag{28}$$

where (a) is from (26) and (b) is from (27). Define $C_2(t) \triangleq \sum_{i \in S(t)} w_i(\bar{\mu}_i(t) - \mu_i)$ and $C_3(t) \triangleq \sum_{i \in S'(t)} w_i(\mu_i - \bar{\mu}_i(t))$. In Appendices D and E, we will show the following two bounds, respectively:

$$\sum_{t=0}^{T-1}\mathbb{E}[C_2(t)] \le w_{\max}\left(2\sqrt{6mNT\log T} + \frac{5}{2}N\right), \tag{29}$$

$$\sum_{t=0}^{T-1}\mathbb{E}[C_3(t)] \le \frac{3}{2}w_{\max}N. \tag{30}$$

Finally, summing (28) for all $t \in \{0, \dots, T-1\}$, dividing both sides of the resulting inequality by $T\eta$, and plugging (29) and (30) into it yield (25).

Remark: The bound in (29) consists of two terms: the first term is of the order $O(\sqrt{T\log T})$, which corresponds to the notion of regret in typical MAB problems and is attributed to the cost that needs to be paid in the learning/exploration process; the second term is a constant, which is from applying the Chernoff-Hoeffding bound (see, e.g., [8]) to a "bad" event $\{\hat{\mu}_i(t-1) - \mu_i > \sqrt{\frac{3\log t}{2h_i(t-1)}}\}$. Similarly, the bound in (30) is from applying the Chernoff-Hoeffding bound to another "bad" event $\{\hat{\mu}_i(t-1) - \mu_i < -\sqrt{\frac{3\log t}{2h_i(t-1)}}\}$.

D. Bounding $C_2(t)$

In this section, we want to show (29).

Consider an arbitrary arm i in $\mathcal{N}$ and an arbitrary round $t = 0, 1, \dots, T-1$. Let t_a^i be the round in which arm i is played for the a-th time. Recall that $h_i(t)$ is the number of times arm i has been played by the end of round t. Clearly, we have $d_i(t_a^i) = 1$, $h_i(t_a^i) = a$, and $h_i(t_a^i - 1) = a - 1$ for all $a \in \{1, 2, \dots, h_i(T-1)\}$. In addition, we also have

$$0 \le t_1^i < t_2^i < \cdots < t_{h_i(T-1)}^i < T. \tag{31}$$

Define the following event:

$$U_i(t) \triangleq \{\bar{\mu}_i(t) < \mu_i\}. \tag{32}$$

Let E^c be the complement of an event E, and let $\mathbb{1}_{\{.\}}$ denote the indicator function. We bound the expectation of $C_2(t)$ as

$$\begin{aligned}
\mathbb{E}[C_2(t)] &= \mathbb{E}\left[\sum_{i=1}^{N} w_i(\bar{\mu}_i(t) - \mu_i)d_i(t)\right] \\
&= \mathbb{E}\left[\sum_{i=1}^{N} w_i(\bar{\mu}_i(t) - \mu_i)d_i(t)\mathbb{1}_{\{U_i(t)\}}\right] \\
&\quad + \mathbb{E}\left[\sum_{i=1}^{N} w_i(\bar{\mu}_i(t) - \mu_i)d_i(t)\mathbb{1}_{\{U_i^c(t)\}}\right] \\
&\overset{(a)}{\le} \mathbb{E}\left[\sum_{i=1}^{N} w_i(\bar{\mu}_i(t) - \mu_i)d_i(t)\mathbb{1}_{\{U_i^c(t)\}}\right] \\
&\overset{(b)}{\le} w_{\max}\sum_{i=1}^{N} \mathbb{E}[\underbrace{(\bar{\mu}_i(t) - \mu_i)d_i(t)\mathbb{1}_{\{U_i^c(t)\}}}_{J_1(t)}],
\end{aligned} \tag{33}$$

where (a) is due to $\bar{\mu}_i(t) < \mu_i$ when event $U_i(t)$ happens and (b) is due to $\bar{\mu}_i(t) \ge \mu_i$ when event $U_i^c(t)$ happens. Define $J_1(t) \triangleq (\bar{\mu}_i(t) - \mu_i)d_i(t)\mathbb{1}_{\{U_i^c(t)\}}$. Also, define another event:

$$F_i(t) \triangleq \left\{\hat{\mu}_i(t-1) - \mu_i \le \sqrt{\frac{3\log t}{2h_i(t-1)}}\right\}. \tag{34}$$

Then, summing $J_1(t)$ for all $t \in \{0, \dots, T-1\}$ gives

$$\begin{aligned}
&\sum_{t=0}^{T-1} J_1(t) \\
&\overset{(a)}{=} \sum_{a=1}^{h_i(T-1)} (\bar{\mu}_i(t_a^i) - \mu_i)\mathbb{1}_{\{U_i^c(t_a^i)\}} \\
&\overset{(b)}{\le} 1 + \sum_{a=2}^{h_i(T-1)} (\bar{\mu}_i(t_a^i) - \mu_i)\mathbb{1}_{\{U_i^c(t_a^i)\}} \\
&= 1 + \sum_{a=2}^{h_i(T-1)} (\bar{\mu}_i(t_a^i) - \mu_i)\mathbb{1}_{\{U_i^c(t_a^i)\}}(\mathbb{1}_{\{F_i(t_a^i)\}} + \mathbb{1}_{\{F_i^c(t_a^i)\}}) \\
&\overset{(c)}{\le} 1 + \sum_{a=2}^{h_i(T-1)} (\underbrace{(\bar{\mu}_i(t_a^i) - \mu_i)\mathbb{1}_{\{U_i^c(t_a^i)\cap F_i(t_a^i)\}}}_{J_2(t_a^i)} + \mathbb{1}_{\{F_i^c(t_a^i)\}}),
\end{aligned} \tag{35}$$

where (a) is due to $d_i(t_a^i) = 1$ for all $a \in \{1, 2, \dots, h_i(T-1)\}$ and $d_i(t) = 0$ for all other t, (b) is due to $\bar{\mu}_i(t_1^i) - \mu_i \le 1$, and (c) is due to $(\bar{\mu}_i(t_a^i) - \mu_i)\mathbb{1}_{\{U_i^c(t_a^i)\}} \le 1$. We define $J_2(t_a^i) \triangleq (\bar{\mu}_i(t_a^i) - \mu_i)\mathbb{1}_{\{U_i^c(t_a^i)\cap F_i(t_a^i)\}}$ and want to bound both $\sum_{a=2}^{h_i(T-1)} \mathbb{E}[J_2(t_a^i)]$ and $\sum_{a=2}^{h_i(T-1)} \mathbb{E}[\mathbb{1}_{\{F_i^c(t_a^i)\}}]$.

First, we want to bound $\sum_{a=2}^{h_i(T-1)} \mathbb{E}[J_2(t_a^i)]$. Consider t_a^i for all $a \in \{2, \dots, h_i(T-1)\}$. Suppose event $F_i(t_a^i)$ happens. Then, we have

$$\hat{\mu}_i(t_a^i - 1) - \mu_i \le \sqrt{\frac{3\log t_a^i}{2h_i(t_a^i - 1)}}. \tag{36}$$

From (6), we also have

$$\bar{\mu}_i(t_a^i) \le \hat{\mu}_i(t_a^i - 1) + \sqrt{\frac{3\log t_a^i}{2h_i(t_a^i - 1)}}. \tag{37}$$

Combining (36) and (37) gives

$$\bar{\mu}_i(t_a^i) - \mu_i \le 2\sqrt{\frac{3\log t_a^i}{2h_i(t_a^i - 1)}}, \tag{38}$$

which implies that for all $a \in \{2, \dots, h_i(T-1)\}$, we have

$$\begin{aligned}
J_2(t_a^i) &= (\bar{\mu}_i(t_a^i) - \mu_i)\mathbb{1}_{\{U_i^c(t_a^i)\cap F_i(t_a^i)\}} \\
&\le 2\sqrt{\frac{3\log t_a^i}{2h_i(t_a^i - 1)}}.
\end{aligned} \tag{39}$$

Then, summing $J_2(t_a^i)$ for all $a \in \{2, \dots, h_i(T-1)\}$ gives

$$\begin{aligned}
\sum_{a=2}^{h_i(T-1)} J_2(t_a^i) &\overset{(a)}{\le} \sum_{a=2}^{h_i(T-1)} 2\sqrt{\frac{3\log t_a^i}{2h_i(t_a^i - 1)}} \\
&\overset{(b)}{\le} \sqrt{6\log T}\sum_{a=2}^{h_i(T-1)} \frac{1}{\sqrt{a-1}} \\
&\overset{(c)}{\le} \sqrt{6\log T}\left(1 + \int_{1}^{h_i(T-1)} \frac{1}{\sqrt{x}}\,dx\right) \\
&\le 2\sqrt{6h_i(T-1)\log T},
\end{aligned} \tag{40}$$

where (a) is from (39), (b) is due to $t_a^i \le T$ (from (31)) and $h_i(t_a^i - 1) = a - 1$ for all $a \in \{2, \dots, h_i(T-1)\}$, and (c) is due to a basic relationship between the considered summation and integral. Therefore, we have

$$\sum_{a=2}^{h_i(T-1)} \mathbb{E}[J_2(t_a^i)] \le 2\sqrt{6\log T}\,\mathbb{E}[\sqrt{h_i(T-1)}]. \tag{41}$$

Next, we want to bound $\sum_{a=2}^{h_i(T-1)} \mathbb{E}[\mathbb{1}_{\{F_i^c(t_a^i)\}}]$. By using the Chernoff-Hoeffding bound (see, e.g., [8]), we have

$$\begin{aligned}
\mathbb{E}[\mathbb{1}_{\{F_i^c(t_a^i)\}}] &= \mathbb{P}\left\{F_i^c(t_a^i)\right\} \\
&= \mathbb{P}\left\{\hat{\mu}_i(t_a^i - 1) - \mu_i > \sqrt{\frac{3\log t_a^i}{2h_i(t_a^i - 1)}}\right\} \\
&\le \frac{1}{{t_a^i}^3}.
\end{aligned} \tag{42}$$

Noting that $\{t_2^i, \dots, t_{h_i(T-1)}^i\} \subseteq \{1, 2, \dots\}$, we have

$$\begin{aligned}
\sum_{a=2}^{h_i(T-1)} \mathbb{E}[\mathbb{1}_{\{F_i^c(t_a^i)\}}] &\le \sum_{a=2}^{h_i(T-1)} \frac{1}{{t_a^i}^3} \\
&\le \sum_{t=1}^{\infty} \frac{1}{t^3} \\
&\le \frac{3}{2}.
\end{aligned} \tag{43}$$

Taking expectation of both sides of (35) and plugging (41) and (43) into it yield

$$\sum_{t=0}^{T-1} \mathbb{E}[J_1(t)] \leq 2\sqrt{6\log T}\mathbb{E}[\sqrt{h_i(T-1)}] + \frac{5}{2}. \tag{44}$$

Finally, summing (33) for all $t \in \{0, \ldots, T-1\}$ and plugging (44) into it yield (29):

$$\begin{aligned}\sum_{t=0}^{T-1} \mathbb{E}[C_2(t)] &\leq w_{\max} \sum_{i=1}^{N} \left(2\sqrt{6\log T}\mathbb{E}[\sqrt{h_i(T-1)}] + \frac{5}{2}\right) \\ &\leq w_{\max} \left(2\sqrt{6mNT\log T} + \frac{5}{2}N\right),\end{aligned} \tag{45}$$

where the last step follows from $\frac{1}{N}\sum_{i=1}^{N}\sqrt{h_i(T-1)} \leq \sqrt{\frac{1}{N}\sum_{i=1}^{N} h_i(T-1)}$ (due to Jensen's inequality) and $\sum_{i=1}^{N} h_i(T-1) \leq Tm$ (due to the fact that at most m arms can be selected in each round).

E. Bounding $C_3(t)$

In this section, we want to show (30).

Consider an arbitrary arm i in $\mathcal{N}$ and an arbitrary round $t = 0, 1, \ldots, T-1$. Recall that $C_3(t) = \sum_{i \in S'(t)} w_i(\mu_i - \bar{\mu}_i(t))$. Let $\mathbf{d}'(t) = (d'_1(t), \ldots, d'_N(t))$ be the action vector corresponding to $S'(t)$. Also, recall that $U_i(t) = \{\bar{\mu}_i(t) < \mu_i\}$. Similar to the derivation for $C_2(t)$ in (33), we bound the expectation of $C_3(t)$ as

$$\begin{aligned}\mathbb{E}[C_3(t)] &= \mathbb{E}\left[\sum_{i=1}^{N} w_i(\mu_i - \bar{\mu}_i(t))d'_i(t)\right] \\ &= \mathbb{E}\left[\sum_{i=1}^{N} w_i(\mu_i - \bar{\mu}_i(t))d'_i(t)\mathbb{1}_{\{U_i(t)\}}\right] \\ &\quad + \mathbb{E}\left[\sum_{i=1}^{N} w_i(\mu_i - \bar{\mu}_i(t))d'_i(t)\mathbb{1}_{\{U_i^c(t)\}}\right] \\ &\overset{(a)}{\leq} \mathbb{E}\left[\sum_{i=1}^{N} w_i(\mu_i - \bar{\mu}_i(t))d'_i(t)\mathbb{1}_{\{U_i(t)\}}\right] \\ &\overset{(b)}{\leq} w_{\max}\sum_{i=1}^{N} \mathbb{E}[\underbrace{(\mu_i - \bar{\mu}_i(t))d'_i(t)\mathbb{1}_{\{U_i(t)\}}}_{K_1(t)}],\end{aligned} \tag{46}$$

where (a) is due to $\bar{\mu}_i(t) \geq \mu_i$ when event $U_i^c(t)$ happens, and (b) is due to $\bar{\mu}_i(t) < \mu_i$ when event $U_i(t)$ happens. We define $K_1(t) \triangleq (\mu_i - \bar{\mu}_i(t))d'_i(t)\mathbb{1}_{\{U_i(t)\}}$ and consider two cases for $\mathbb{E}[K_1(t)]$: i) $t \leq t_1^i$ and ii) $t > t_1^i$.

In Case i), event $U_i(t)$ must not happen, i.e., $\bar{\mu}_i(t) \geq \mu_i$ must hold. This is because $\bar{\mu}_i(t) = 1$ (due to $h_i(t-1) = 0$ for $t \leq t_1^i$) and $\mu_i \in [0, 1]$. Hence, for all $t \leq t_1^i$ we have

$$\mathbb{E}[K_1(t)] = 0. \tag{47}$$

In Case ii), suppose event $U_i(t)$ happens. Then we have $\bar{\mu}_i(t) < \mu_i \leq 1$. This, along with (6), implies $\bar{\mu}_i(t) = \hat{\mu}_i(t-1) + \sqrt{\frac{3\log t}{2h_i(t-1)}}$, which further implies

$$U_i(t) = \left\{\hat{\mu}_i(t-1) - \mu_i < -\sqrt{\frac{3\log t}{2h_i(t-1)}}\right\}. \tag{48}$$

This leads to the following bound on $\mathbb{E}[K_1(t)]$ for all $t > t_1^i$:

$$\begin{aligned}\mathbb{E}[K_1(t)] &= \mathbb{E}[(\mu_i - \bar{\mu}_i(t))d'_i(t)\mathbb{1}_{\{U_i(t)\}}] \\ &\overset{(a)}{\leq} \mathbb{E}[\mathbb{1}_{\{U_i(t)\}}] \\ &= \mathbb{P}\left\{\hat{\mu}_i(t-1) - \mu_i < -\sqrt{\frac{3\log t}{2h_i(t-1)}}\right\} \\ &\overset{(b)}{\leq} \frac{1}{t^3},\end{aligned} \tag{49}$$

where (a) is due to $\mu_i - \bar{\mu}_i(t) \leq 1$ and $d'_i(t) \leq 1$, and (b) is from the Chernoff-Hoeffding bound (see, e.g., [8]).

Summing $\mathbb{E}[K_1(t)]$ for all $t \in \{0, \ldots, T-1\}$ and applying (47) and (49) yield

$$\begin{aligned}\sum_{t=0}^{T-1} \mathbb{E}[K_1(t)] &\leq \sum_{t=t_1^i+1}^{T-1} \frac{1}{t^3} \\ &\leq \sum_{t=1}^{\infty} \frac{1}{t^3} \\ &\leq \frac{3}{2}.\end{aligned} \tag{50}$$

Finally, Summing (46) for all $t \in \{0, \ldots, T-1\}$ and plugging (50) into it yield (30).