# Anytime Fairness Guarantees in Stochastic Combinatorial MABs: A Novel Learning Framework

Subham Pokhriyal
Indian Institute of Technology Ropar
Rupnagar, India
subham.22csz0002@iitrpr.ac.in

Shweta Jain
Indian Institute of Technology Ropar
Rupnagar, India
shwetajain@iitrpr.ac.in

Ganesh Ghalme
Indian Institute of Technology Hyderabad
Hyderabad, India
ganeshghalme@ai.iith.ac.in

Vaneet Aggarwal
Purdue University
West Lafayette, USA
vaneet@purdue.edu

## ABSTRACT

This paper proposes a novel framework for incorporating *anytime fairness* guarantees in a general Stochastic Combinatorial Multi-Armed Bandit (CMAB) problem when the time horizon is unknown. Our framework does not make any assumptions about the reward feedback or structure and provides fairness guarantees as long as a sublinear regret algorithm exists to solve the same problem. The framework essentially operates in the episodes of length $H$, which is a user-defined parameter. The framework divides each episode of length $H$ into fairness rounds and learning rounds. Motivated by preemptive scheduling on uniform machines, we propose `Fair-CMAB` which prioritizes fairness rounds upfront and uses any existing CMAB algorithm for learning rounds. This helps in generalizing the framework significantly. Theoretically, we prove that for a sufficiently large value of $H$, `Fair-CMAB` achieves anytime fairness guarantees after some initial number of rounds and achieves the regret guarantees of the same order as the learning algorithm.

## KEYWORDS

Combinatorial Multi-Arm Bandits; Fairness; Scheduling

**ACM Reference Format:**
Subham Pokhriyal, Shweta Jain, Ganesh Ghalme, and Vaneet Aggarwal. 2025. Anytime Fairness Guarantees in Stochastic Combinatorial MABs: A Novel Learning Framework. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 10 pages.

## 1 INTRODUCTION

The classical Multi-Armed Bandit (MAB) problem, introduced by [53], is an important online decision-making problem where the data arrives in a sequence and the learner must make an irrevocable decision at each time. In a stochastic MAB problem, a learner selects a *single* arm from the set of $N$ arms at each time step and receives a reward sampled from an independent distribution associated with that arm. The objective is to maximize the expected cumulative reward or, equivalently, to minimize the regret.



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

The Combinatorial Multi-Armed Bandit (CMAB) problem generalizes this framework by allowing the learner to select a subset of arms at each time. The expected reward from a subset of arms is determined by a deterministic mapping from a set of arms pulled to a real-valued reward as a function of the expected rewards from the pulled arms. Upon selecting a subset of arms, the learner receives feedback in terms of a random reward. CMAB is studied under two feedback types: semi-bandit feedback [5, 15, 22, 23, 31, 34], where individual reward samples from each selected arm are observed; and full-bandit feedback [1, 2, 19, 21, 41–43], where only the combined reward for the selected subset is revealed to the learner. CMAB is further studied under two settings for the reward function: linear [18, 50, 52], or non-linear [19, 21, 40, 41, 43], with the reward function being linear or non-linear, respectively, based on the individual rewards of the arms in the selected subset. CMAB plays an important role in applications such as sponsored search auctions [44], crowdsourcing [51], influence maximization [14, 61], recommender systems [49], etc. The feedback type and the structure of the reward function directly influence the arm-pulling strategy and learning efficiency. Note that the semi-bandit feedback reveals more information to the learner than a full-bandit feedback.

While a long line of work (see [1–3, 19, 21, 41, 43] for an overview) focuses on identifying the optimal set of arms, incorporating fairness requirements in the stochastic CMAB problem remains largely unexplored. An optimal MAB algorithm tends to allocate resources or opportunities in a skewed manner, and such *winner-takes-all* allocations may lead to societal issues such as loss of trust. Fairness is important in many real-world applications in combinatorial bandits, where equitable distribution of resources or opportunities among arms is crucial: in sponsored ads, it ensures visibility for all advertisers; in crowdsourcing, equitable task distribution; in wireless scheduling, minimum service quality [37]; and in recommender systems, diverse content and cross-selling prevention [2, 62].

This paper focuses on fairness constraints induced by a *minimum guaranteed number of pulls* for each individual arm. Though this problem has interesting connections with the scheduling literature (see [39]), the design of optimal online learning with a minimum-pull guarantee for each arm is relatively new. Patil et al. [46] introduces the problem of satisfying the minimum-pull guarantee as a fraction of the total number of pulls for each round $t$ (aka, anytime guarantee) in the classical stochastic MAB setting when only a single arm needs to be pulled at a time.

Anytime fairness guarantee is required in many applications that use CMAB. One such example is an online crowdsourcing platform where organizations would like to choose the best $K$ workers (arms) at each time. However, selecting only a few workers at each time might lead to workers leaving the platforms. Therefore, one might ask to maintain a minimum exposure guarantee for each worker. Another example is in sponsored search advertisements. As per the example provided in [46], Facebook was recently sued by the US Department of Housing and Urban Development for targeting ads based on attributes such as gender, race, and religion. Anytime fairness guarantee can demand the minimum fraction of exposure on such attributes thus preventing these discriminations while selecting $K$ advertisements at each time.

Existing regret guarantees in [46] hold only when the learning algorithm is based on the Upper Confidence Bound (UCB) [6, 7] approach. Therefore, the algorithm in [46] cannot be extended to the CMAB setting, because, firstly, in the CMAB setting, the most rewarding arm may not be a part of the optimal subset of arms. Thus, allocating the remaining rounds after satisfying the fairness requirement to a set of optimal arms may not result in an optimal schedule. And secondly, in a full-bandit feedback setting, where reward feedback is available for the subset of pulled arms, but the fairness requirements are specified individually for each arm. In this context, it is unclear how to identify the best schedule from all feasible schedules. More importantly, the theoretical analysis for regret changes is based on the base learning algorithm. Our main aim is to provide a unified framework for fairness in CMAB problems which can be integrated with any existing algorithm while providing anytime fairness guarantees.

Our unified approach not only advances the theoretical understanding of fairness in CMAB problems but also facilitates the practical application of these algorithms across diverse, real-world scenarios. Further, our framework can also be used to provide anytime fairness guarantees for existing algorithms in classical bandit settings since CMAB generalizes classical bandit settings (where only one arm is selected). In summary, our contributions include:

(1) We introduce an anytime `Fair-CMAB` framework for a broad class of feedback settings (semi-bandit and bandit), applicable to different classes of reward functions (linear, non -linear, and submodular) along with cardinality constraints.
(2) Anytime `Fair-CMAB` framework uses a simple algorithm that incorporates a user-defined parameter $H$. It guarantees anytime fairness for all rounds $t \geq H\eta_H$, with $\eta_H$ denoting the minimum fraction of the $H$ rounds required to ensure that each arm is pulled the necessary fraction of times up to $H$.
(3) We show that the extra number of rounds that the `Fair-CMAB` framework uses for fairness converges to that of the number of fairness rounds used by any optimal fair algorithm for sufficiently small value of $H$.
(4) We prove that the anytime `Fair-CMAB` framework achieves similar order bounds on regret as the learning algorithm used for the framework.

## 2 RELATED WORK

We first discuss existing CMAB algorithms with semi-bandit and full-bandit feedback followed by literature on fairness in stochastic bandits. A comprehensive coverage of the theoretical insights and application of MAB can be found in [10, 35, 55].

### 2.1 Stochastic Combinatorial Bandits

CMAB was first studied for the shortest path problem by György et al. [28] and later extensively analyzed by Lattimore and Szepesvári [35]. Subsequent works on combinatorial bandits with semi-bandit feedback include [12, 13, 16, 17, 23, 24, 32, 59, 61]. Gai et al. [23] introduced approximation oracles under linear rewards, while Chen et al. [15] generalized this to $\alpha$-approximation oracles covering a much larger class of linear and nonlinear rewards. Furthermore, Chen et al. [16] extended these results considering arm dependencies and probabilistically trigger arms, establishing online influence maximization as a subclass. Wang and Chen [61] refined regret bounds from exponential to sublinear for this subclass. The work of Zimmert et al. [64] further bridged adversarial and stochastic settings to accommodate nonlinear reward functions. Although these studies focus on optimizing reward or minimizing regret, none considers fairness in CMAB. While semi-bandit feedback is a simplified setting, there are numerous applications for the bandit feedback setting [21, 40, 41, 43, 47, 57, 58]. Earlier works on full-bandit feedback include [18, 52] for the linear reward setting and Agarwal et al. [1, 2] for the non-linear reward setting. It must be noted that different approaches are required to solve the problem depending on the structure of the rewards. For example, monotone submodular rewards result in a cardinality-constrained best K-arms selection problem, whereas non-monotone submodular rewards does not require cardinality constraint. Similarly, while the approach in [1] works by dividing the arms into groups and merging them to obtain the best $K$ arms, [20, 21, 41, 43] proposed an explore-then-commit based algorithm in a more general reward model.

### 2.2 Fairness in Stochastic and CMAB with Semi-Bandit Feedback

For the classical bandit setting, similar to ours, [46] guarantees anytime fairness of exposure for each arm, and [60] introduces merit-based exposure to the arms. Motivated by the group fairness notion in the machine learning community [8, 27, 29, 30, 45, 54], Grazzi et al. [26] considers exposure fairness within the group, while Pokhriyal et al. [48] proposed bi-level fairness, which ensures minimum exposure (anytime guarantee) to groups and merit-based exposure (asymptotic guarantee) to arms within the group.

In the context of CMAB settings with semi-bandit feedback, Li et al. [37] propose a UCB-based approach using a virtual queue technique to ensure fairness in expectation for each arm, asymptotically. Extending this, Xu et al. [63] employ online convex optimization to achieve sublinear regret, generalizing to concave rewards and knapsack constraints. Liu et al. [38] unify bandits with knapsack and fairness constraints using an LP-style algorithm, achieving problem-independent regret bounds with zero fairness violations. For real-time systems, Steiger et al. [56] focus on short-term fairness in combinatorial semi-bandit problems with sleeping arms and delayed feedback, providing instance-independent regret bounds. All these fairness notions provide only asymptotic guarantees. Furthermore, each work provides fairness guarantees in a specific setting that too under the semi-bandit feedback setting. On the other hand,

our proposed framework works for a large variety of settings as long as an algorithm exists for the corresponding unfair setting. This allows our framework to function even for bandit feedback settings, where no prior work has specifically addressed fairness.

In another growing set of works in the equitability over time fairness setting in online algorithms, such as ones in [4, 11, 36], the setup is different from ours, as these works consider arms arrival over time and the goal is to decide whether to pull the arm or not. Whereas, in our setting, stochastic arms are available and the goal is to learn the arm with optimal reward.

## 3 PROBLEM FORMULATION

In a combinatorial MAB setting, the learner pulls a subset $S_t$ of arms from a finite set $[N] = \{1, 2, \cdots, N\}$ at each time $t$ and receives a stochastic reward $f_t(S_t)$ until an a priori unknown time horizon $T$. We define an arm $i \in [N]$ as a base arm and refer to $S \subseteq [N]$ as a super arm. There are two possible settings in CMAB. The first is a fixed budget setting [1, 2, 41, 52], where the learner pulls exactly $K$ arms at each time. The second allows the learner to pull any number of arms in each round (e.g., non-monotone submodular bandits [19]). The set of all possible actions is represented by $\mathcal{A}$. In the budgeted MAB setting, $\mathcal{A} = \{S \subseteq [N] : |S| = K\}$, and for the non-monotone setting, $\mathcal{A} = \{S \subseteq [N]\}$.

In the stochastic setting, the expected reward for each action $S$ is derived from a reward function $f : \mathcal{A} \to \mathbb{R}$. The realized reward for $S \in \mathcal{A}$ comes from a fixed distribution $D_S$ with a mean $f(S)$. Let the realized reward at time $t$ when pulling the super arm $S_t$ be denoted by $f_t(S_t)$. Then, the reward for action $S_t$ is $f(S_t) = \mathbb{E}[f_t(S_t)]$, where $\{f_1, f_2, \ldots, f_t\}$ are independent and identically distributed (IID) rewards for $S_t \in \mathcal{A}$ generated from distribution $D_{S_t}$. We further assume that the rewards $f_t$ are bounded[1]. For simplicity of notation, we omit the time index $t$ when evident from context. The learner's goal is to identify and play the optimal action $S^\star \in \mathcal{A}$ defined as $S^\star \in \arg\max_{S \in \mathcal{A}} f(S)$.

Without loss of generality, we assume that the optimal super-arm is unique, that is for all $S \neq S^\star$, $f(S^\star) > f(S)$. In the stochastic setting, based on how the feedback is received, we have two settings.

(1) **Full-bandit Feedback:** In a full-bandit feedback setting [3, 19, 21, 43, 52], the learner only observes the reward $f_t(S_t)$ for the selected action $S_t$.
(2) **Semi-Bandit Feedback:** In the case of semi-bandit feedback [15, 61, 64], the learner additionally observes rewards for each base arm in a selected super arm $S_t$.

Note that `Fair-CMAB` utilizes existing CMAB algorithms for learning, which allows it to operate for both the feedback settings.

### 3.1 Fairness Requirement:

This paper studies the fairness requirement that guarantees that each base arm is pulled for at least a specified fraction of times. We define the fairness constraint through a vector $r = (r_1, r_2, \ldots, r_N)$, where $r_i \in \left(0, \frac{1}{2\kappa}\right)$ for each arm $i \in [N]$, with $\kappa = \lceil N/K \rceil$. This parameter $\kappa$ restricts the fairness quota for each arm, ensuring that each arm $i$ is pulled at least a fraction $r_i$ of times. The goal of the learner is to satisfy fairness constraints at every time step $t$ by ensuring that the number of times arm $i$ has been pulled by time $t$, denoted by $n_{i,t}$, satisfies: $n_{i,t} \geq \lfloor r_i t \rfloor \quad \forall i \in [N]$.

[1]We assume, without loss of generality, that $f_t(S_t) \in [0, 1]$.

Definition 1 ($(r, \gamma)$-fair algorithm). *Let $T > 1$ be an arbitrary stopping time. We call a combinatorial multi-armed bandit (CMAB) algorithm $(r, \gamma)$-fair if, for a given fairness vector $r = (r_1, r_2, \ldots, r_N)$ and a given $\gamma > 0$, it satisfies the following condition for $t \geq \gamma$ i.e.,*

$$n_{i,t} \geq \lfloor r_i \cdot t \rfloor \; \forall t \geq \gamma.$$

Note that we recover *aytime* fairness guarantee when $\gamma = 1$ [46].

### 3.2 Regret

The $\alpha$-regret of the algorithm used in this paper captures the gap between $\alpha$ times sum of the rewards when optimal subset is chosen and the rewards obtained by the algorithm [43]. More formally,

Definition 2 ($\alpha$-Regret). *Let $(S_t)_{t \geq 1}$ denote the sequence of pulls made by CMAB algorithm* Alg, *$T > 1$ be an arbitrary stopping time and $\alpha > 0$. The $\alpha$-regret of Alg over $T$ rounds is defined as*

$$\mathcal{R}^{Alg}(T) = \alpha \cdot T \cdot f(S^\star) - \mathbb{E}\left[\sum_{t=1}^{T} f_t(S_t)\right],$$

*where $f(S^\star)$ is the optimal expected reward.*

The goal of the learner is to maximize the expected cumulative reward $\mathbb{E}\left[\sum_{t=1}^{T} f_t(S_t)\right]$, to obtain sublinear $\alpha-$regret while optimizing for approximation parameter $\alpha$. Note that the parameter $\alpha$ absorbs the approximation factor for combinatorial bandits and the fraction of rounds spent by the algorithm towards fairness. With $K = 1$, the definition coincides with the fairness-aware regret by Patil et al. [46] when the algorithm spends an optimal number of rounds towards fairness. We now provide a framework that works for a broad class of stochastic combinatorial bandit settings.

## 4 `FAIR-CMAB`: A FRAMEWORK TOWARDS FAIR COMBINATORIAL MAB ALGORITHM

Our proposed framework works in episodes each of fixed length $H$. For each episode $\mathcal{E}$, the algorithm computes the fraction of time required to satisfy the fairness guarantees in a given episode. Our fairness framework essentially creates a sequence of intervals $I_0, I_1, I_2, \ldots$, with predefined lengths that separate pulls for ensuring fairness and learning, respectively. Let us further define

$$\eta_d = \frac{1}{d} \max\left(\left\lceil \frac{\sum_i \lceil r_i d \rceil}{K} \right\rceil, \max_i \lceil r_i d \rceil\right). \tag{1}$$

Thus, the terms $\eta_d$ and $d\eta_d$ denote the minimum fraction and the minimum number of pulls required to satisfy fairness guarantees for $d$ rounds, respectively, which we explain later in connection to Makespan scheduling [39].

Further, we create the lengths of the intervals as follows. The initial interval $I_0$ is considered of length $H\eta_H$. Apart from $I_0$, every odd interval $I_{2\ell-1}$, $\ell = 1, \ldots, m$, is of length $(\ell+1)H\eta_{(\ell+1)H} - \ell H\eta_{\ell H}$, and is called a fairness-aware interval where the framework primarily selects the arms to satisfy the fairness requirement for the next episode that follows. Here, index $\ell$ represents the index of the episode, which starts after the initial pulls for $H\eta_H$ rounds, and $m = \left\lceil \frac{T - H\eta_H}{H} \right\rceil$ represent the last episode. It should be further

noted that to compute the length of the intervals for Fair-CMAB we require $K$ as an input which is not fixed for non-monotone and non-budgeted CMAB setting. For such a setting, we provide $K = N$ as an input to the algorithm (for the fairness intervals).

On the other hand, the even intervals $I_{2\ell}, \ell = 1, \ldots, m$, are called the learning intervals, and are of length $H - |I_{2\ell-1}| = H - (\ell + 1)H\eta_{(\ell+1)H} + \ell H\eta_{\ell H}$. In the even intervals, the framework primarily pulls the arms that are suggested by any learning algorithm Learn$(\cdot)$ (without worrying about the fairness constraints). These intervals can be better understood by Figure 1. Note that substituting $K = N$ in fairness (odd) rounds does not hamper the learning of base algorithm as we are calling the base algorithm only in even intervals.

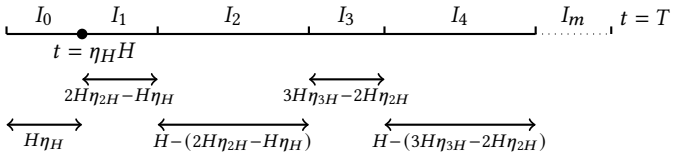



**Figure 1: Intervals with fairness and learning phases along the timeline.**

We note that if $T$ is known, then we could divide the entire time horizon into two intervals namely $I_1$ and $I_2$ of length $T\eta_T$ and $T(1-\eta_T)$ rounds such that pulls done in interval $I_1$ would have been sufficient for all times in $I_2$ to satisfy anytime fairness guarantees (albeit after $T\eta_T$ time). However, unknown $T$ requires us to solve the problem separately for each episode until the stopping time. Doubling trick [7, 9], where the estimate of time doubles every time when the algorithm exceeds the estimated time from previous rounds might adversely affect the fairness requirement as it will lead to exponential growth of fairness rounds. This would lead to high regret in last round if the stopping time is before the end of the episode. For instance, we would incur linear regret if the stopping time is just at the end of the fairness time-slot in the last epoch. Therefore, we propose a fixed length episode strategy for achieving anytime fairness over an unknown time horizon $T$. Next, connect $\eta_d$ in equation (1) to the makespan of the job scheduling problem on identical machines with preemption [39].

*Connection to Makespan Scheduling:* The makespan problem minimizes the time to complete $N$ jobs on $K$ machines with preemption with each job $i$ having a processing time $p_i$. Analogously, jobs correspond to arms, machines to the number of arms pulled, and as we enforce fairness for $H$ rounds, $p_i = \lceil r_i H \rceil$ represents the time required to meet fairness guarantees for the $H$ rounds.

Definition 3. *Then Makespan on identical machines with preemption is given by [39]:*

$$\texttt{Makespan}(N, K, p) = \max\left\{\frac{\sum_i p_i}{K}, \max_i p_i\right\}.$$

This definition directly leads to the value of $d\eta_d$ in Equation (1). Therefore, any preemptive scheduling algorithm that achieves the optimal makespan will guarantee that after running the algorithm for $H\eta_H$ number of rounds, each arm will be pulled at least $\lceil r_i H \rceil$ times which forms the foundation of our framework.

## 5 THE ALGORITHM : FAIR-CMAB

As mentioned earlier, the algorithm runs in episodes of equal size, $H$ until the time horizon $T$. The algorithm divides the entire duration into three sets of intervals, namely, the initial interval (fairness), odd intervals (fairness), and even intervals (learning). The length of the initial interval is given by $H\eta_H$. Our episodes of length $H$ start after this initial interval and the indexing of the episode is denoted by $\ell$. For each $\ell = 1, \cdots, m$, $\ell^{th}$ fairness interval is of the length $(\ell + 1)H\eta_{(\ell+1)H} - \ell H\eta_{\ell H}$ which forms a decreasing sequence. Similarly, length of the $\ell^{th}$ learning interval is given as $(1 - (\ell + 1)\eta_{(\ell+1)H} + \ell\eta_{\ell H})H$. Thus, each pair of fairness and learning interval forms an episode of length $H$. During the fairness intervals, the algorithm uses Longest Remaining Processing Time (LRPT) algorithm to pull arms [25] whereas in the learning intervals the algorithm invokes the existing combinatorial MAB algorithm based on the problem instance [2, 3, 40, 41, 43, 57]. In LRPT, the algorithm pulls the arms that have the highest remaining pulls to satisfy the fairness constraint and this is specified in Algorithm 1. Here, $p_i$ denotes the fairness requirement of arm $i$, $|I_\ell|$ denotes the duration for which LRPT is called for, and $T$ is used to indicate that the algorithm should stop as soon as the algorithm reaches the last round. For each LRPT call, $p_i's$ are carefully crafted such that anytime fairness guarantees are maintained and the duration of LRPT is minimized to ensure the optimal value of $\alpha$ in the regret with respect to fairness rounds.

**Algorithm 1** LRPT$(p, |I_\ell|)$

**Require:** Processing time vector $(p = \{p_i\}_{i=1}^N)$, number of time steps for which LRPT to run: $|I_\ell|$.

1: Initialize $n_i^f = 0$ for each arm $i \in [N]$, $t_f = 1$
2: **while** $t_f \leq |I_\ell|$ *and* $t \leq T$ **do**
3: $S_t \leftarrow$ Pull $K$ arms with highest value of $p_i - n_i^f$
4: $t_f = t_f + 1$
5: Update $n_i^f = n_i^f + 1, \forall i \in S_t$
6: **end while**
7: $E = \{i | p_i - n_i^f < 0\}$
8: Return $E$

The Fair-CMAB Framework presented in Algorithm 2 begins with the initialization of key parameters, including the fairness vector $r = \{r_i \in (0, \frac{1}{2K})\}_{i=1}^N$, episode length $H$, the number of arms $N$, and the cardinality constraint $K$. In the next step, the initial interval of length $H\eta_H$ is computed to determine the fairness time slots in the initial interval (Line 2). The algorithm then calls the Longest Remaining Processing Time (LRPT) scheduling with $p_i = \lceil r_i H \rceil$ such that all arms satisfy fairness requirement untill $H$ rounds (Line 3). This ensures anytime fairness guarantees during interval $I_1$ if $\eta_H \leq \frac{1}{2}$.

The algorithm then alternates between fairness and learner slots (Lines 4–16). In the fairness slots, each arm receives a minimum number of pulls as per its fairness quota, ensuring the algorithm maintains anytime fairness. In the learner slots, the algorithm calls a learning procedure. The sets $L, U$, and $E$ are created to distinguish between the arms that might have received extra pulls because of the ceiling factor in the previous rounds. The $p_i's$ of the arms in

**Algorithm 2** Fair-CMAB Framework

**Require:** Fairness vector($r$): $r_i \in \left(0, \frac{1}{2\kappa}\right) \forall i \in [N]$, episode length $H$, number of arms $N$, and cardinality constraint $K$.

Compute $\eta_H$ from equation 1, $\ell = 0$, $|I_\ell| = H\eta_H$,
$E_l = \text{LRPT}(\{\lceil r_i H \rceil\}_{i=1}^N, |I_\ell|)$
$t = H\eta_H + 1$, $\ell = \ell + 1$
**while** $t \leq T$ **do**
  $L = \{i | \lceil r_i \ell H \rceil < \lceil r_i(\ell-1)H \rceil + \lceil r_i H \rceil\}$
  $U = \{i | \lceil r_i \ell H \rceil = \lceil r_i(\ell-1)H \rceil + \lceil r_i H \rceil\}$
  $p_i^\ell = \lfloor r_i H \rfloor \ \forall i \in L$ and $p_i^\ell = \lceil r_i H \rceil \ \forall i \in U$
  $p_i^\ell = p_i^\ell - 1 \ \forall i \in E_\ell$
  $|I_\ell| = (\ell+1)H\eta_{(\ell+1)H} - \ell H \eta_{\ell H}$
  $E_l = \text{LRPT}(\{p_i^\ell\}_{i=1}^N, |I_\ell|)$
  $t = t + |I_\ell|$
  $t_i = 0$
  **while** $t_i \leq H - |I_\ell|$ and $t \leq T$ **do**
    Call Learn($\cdot$)
    $t_i = t_i + 1$
    $t = t + 1$
  **end while**
  $\ell = \ell + 1$
**end while**

their respective sets are updated accordingly for the next fair slot. This alternating process continues until the time limit $t \leq T$ while ensuring that both fairness and learning objectives are achieved. It should be noted that while our algorithm employs $T$, it is only used as a stopping criterion. No parameters within the algorithm explicitly depend on $T$.

## 6 THEORETICAL ANALYSIS

We now provide the theoretical analysis of our algorithm by first proving anytime fairness guarantees for $t > H\eta_H$.

Theorem 1. Fair-CMAB *is r-fair* $\forall\ t > H\eta_H$ *for any CMAB algorithm if* $r_i \in \left[0, \frac{1}{2\kappa} - \epsilon\right]$ *and* $H \geq \frac{1}{\epsilon}$.

We first prove the following lemmas:

Lemma 1. *When* $r_i \in \left[0, \frac{1}{2\kappa} - \epsilon\right]$ *and* $H > \frac{1}{\epsilon}$, *then* $\eta_H \leq \frac{1}{2}$.

Proof. We have $\lceil r_i H \rceil \leq r_i H + 1$. Therefore:

$$\eta_H \leq \max\left\{\frac{\sum r_i}{K} + \frac{N}{KH}, \max_i\{r_i + \frac{1}{H}\}\right\}$$
$$\leq \max\left\{\frac{N}{2\kappa K} - \frac{N\epsilon}{K} + \frac{N}{KH}, \frac{1}{2\kappa} - \epsilon + \frac{1}{H}\right\} \leq \frac{1}{2} \qquad \square$$

Lemma 2. *All fair slots are bounded by* $H\eta_H$, *i.e.*

$$\ell H\eta_{\ell H} - (\ell-1)H\eta_{(\ell-1)H} \leq H\eta_H,\ \forall \ell > 1.$$

Proof. This is immediate from the definition. $\square$

For any $\ell$, let $p_i^\ell$ denote the processing time for arm $i$ when $\ell^{th}$ call to fairness is made and $n_i^\ell$ denote the total number of times arm $i$ is pulled in that interval, then consider the following sets defined in Algorithm 2: $U = \{i | \lceil r_i \ell H \rceil = \lceil r_i(\ell-1)H \rceil + \lceil r_i H \rceil\}$, $L = \{i | \lceil r_i \ell H \rceil < \lceil r_i(\ell-1)H \rceil + \lceil r_i H \rceil\}$, and $E_\ell = \{i | p_i^\ell - n_i^\ell < 0\}$. Note that $E_\ell$ denotes the set of arms that are pulled extra despite their processing time is finished. This might happen because in the last round of LRPT, less than $K$ arms had their processing time left or we are calling LRPT more number of times than required. Further, let $s_j = |E_j|$, then let us define $\widehat{\eta}^\ell$ to be the number of rounds that are required to complete processing times $p_i^\ell$ and is given by $\widehat{\eta}^\ell = \max\left\{\left\lceil \frac{\sum_i p_i^\ell}{K} \right\rceil, \max_i\{p_i^\ell\}\right\} =$

$$\max\left\{\left\lceil \frac{\sum_{i\in U}\lceil r_i H\rceil + \sum_{i\in L}\lfloor r_i H\rfloor - s_{\ell-1}}{K} \right\rceil, \{\lceil r_i H\rceil\}_{i\in U},\right.$$
$$\left.\{\lfloor r_i H\rfloor\}_{i\in L}, \{\lceil r_i H\rceil - 1\}_{i\in U\cap E_\ell}, \{\lfloor r_i H\rfloor - 1\}_{i\in L\cap E_\ell}\right\}$$

Our next important lemma proves the fact that the length of each $\ell^{th}$ fair interval is sufficient to complete $p_i^\ell$ requirement of each arm.

Lemma 3. $\ell H\eta_{\ell H} - (\ell-1)H\eta_{(\ell-1)H} \geq H\widehat{\eta}$.

Proof. $\ell H\eta_{\ell H} = \max\left\{\left\lceil \frac{\sum\lceil r_i \ell H\rceil}{K} \right\rceil, \lceil r_i \ell H\rceil\right\}$

$$= \max\left\{\left\lceil \frac{\sum_{i\in U}\lceil r_i \ell H\rceil + \sum_{i\in L}\lceil r_i \ell H\rceil}{K} \right\rceil, \{\lceil r_i \ell H\rceil\}_{i\in U}, \{\lceil r_i \ell H\rceil\}_{i\in L}\right\}$$
$$= \max\left\{\left\lceil \frac{\sum_{i\in U}\lceil r_i(\ell-1)H\rceil + \lceil r_i H\rceil + \sum_{i\in L}\lceil r_i(\ell-1)H\rceil + \lfloor r_i H\rfloor}{K} \right\rceil,\right.$$
$$\left.\{\lceil r_i(\ell-1)H\rceil + \lceil r_i H\rceil\}_{i\in U}, \{\lceil r_i(\ell-1)H\rceil + \lfloor r_i H\rfloor\}_{i\in L}\right\}$$
$$= \max\left\{\left\lceil \frac{\sum\lceil r_i(\ell-1)H\rceil + \sum_{i\in U}\lceil r_i H\rceil + \sum_{i\in L}\lfloor r_i H\rfloor}{K} \right\rceil,\right.$$
$$\left.\{\lceil r_i(\ell-1)H\rceil + \lceil r_i H\rceil\}_{i\in U}, \{\lceil r_i(\ell-1)H\rceil + \lfloor r_i H\rfloor\}_{i\in L}\right\}$$

Let $\sum\lceil r_i(\ell-1)H\rceil = K\gamma_{\ell-1} + \delta_{\ell-1}$ for some $\delta_{\ell-1} < K, \gamma_{\ell-1}, \delta_{\ell-1} \in \mathbb{Z}$, then:

$$\ell H\eta_{\ell H} = \max\left\{\left\lceil \frac{K\gamma_{\ell-1} + \delta_{\ell-1} + \sum p_i^\ell + s_{\ell-1}}{K} \right\rceil,\right.$$
$$\left.\{\lceil r_i(\ell-1)H\rceil + \lceil r_i H\rceil\}_{i\in U}, \{\lceil r_i(\ell-1)H\rceil + \lfloor r_i H\rfloor\}_{i\in L}\right\}$$
$$\geq \max\left\{\gamma_{\ell-1} + 1 + \left\lceil \frac{\sum p_i^\ell}{K} \right\rceil, \{\lceil r_i(\ell-1)H\rceil + \lceil r_i H\rceil\}_{i\in U},\right.$$
$$\left.\{\lceil r_i(\ell-1)H\rceil + \lfloor r_i H\rfloor\}_{i\in L}\right\} \geq (\ell-1)H\eta_{(\ell-1)H} + H\widehat{\eta}^\ell$$

Here, the second last equality follows from the fact that $\delta_{\ell-1} + s_{\ell-1} \geq K$. This is because $s_{\ell-1}$ represent the number of extra pulls in $\ell - 1^{th}$ fairness round, whereas $\delta_{\ell-1}$ represent the number of arms that were pulled in last slot of $(\ell-1)^{th}$ fairness round, thus $s_{\ell-1} \geq K - \delta_{\ell-1}$. The above lemma implies that LRPT is run at least $H\widehat{\eta}^\ell$ times in every $\ell^{th}$ fair round. $\square$

**Proof of Theorem 1:** We prove this by induction. We know that after LRPT at $I_0$, we have: $n_{i,t} \geq \lceil r_i H\rceil + 1 \forall i \in E_0$ and $n_{i,t} \geq \lceil r_i H\rceil \forall i \notin E_0$. Since $|I_0| + |I_1| \leq H\eta_H + H\eta_H \leq H$ (from Lemma

2), we get anytime fairness guarantee for time in $I_1$ i.e. $n_{i,t} \geq \lceil r_i t \rceil \forall t \in I_1$. Now consider LRPT call at $I_1$, from Lemma 3, we have: $n_{i,t} \geq \lceil r_i H \rceil + p_i^1 + 1 \; \forall i \in E_1, n_{i,t} \geq \lceil r_i H \rceil + \lceil r_i H \rceil \geq \lceil r_i 2H \rceil + 1 \forall i \in E_1 \; \forall i$, and $n_{i,t} \geq \lceil r_i H \rceil + \lceil r_i H \rceil \geq \lceil r_i 2H \rceil \forall i \notin E_1 \; \forall i$.

Further, since the combined length of intervals $I_0, I_1, I_2, I_3$ is less than or equal to $2H$, we get $n_{i,t} \geq \lfloor r_i t \rfloor \; \forall i \in I_1 \cup I_2 \cup I_3$.

By induction hypothesis assume that after $(\ell-1)^{th}$ fairness interval, we have $n_{i,t} \geq \lceil r_i(\ell-1)H \rceil + 1 \; \forall i \in E_{\ell-1}$ and $n_{i,t} \geq \lceil r_i(\ell-1)H \rceil \; \forall i \notin E_{\ell-1}$. Then, after $\ell^{th}$ call to fairness round,

$$\begin{aligned} n_{i,t} &\geq \lceil r_i(\ell-1)H+ \rceil + \lfloor r_i H \rfloor + 1 \; \forall i \in L \cup E_\ell \\ n_{i,t} &\geq \lceil r_i(\ell-1)H+ \rceil + \lfloor r_i H \rfloor \; \forall i \in L \setminus E_\ell \\ n_{i,t} &\geq \lceil r_i(\ell-1)H+ \rceil + \lceil r_i H \rceil + 1 \; \forall i \in U \cup E_\ell \\ n_{i,t} &\geq \lceil r_i(\ell-1)H+ \rceil + \lceil r_i H \rceil + 1 \; \forall i \in U \setminus E_\ell \end{aligned}$$

This immediately give us the result $n_{i,t} \geq \lceil r_i(\ell) H \rceil + 1 \; \forall i \in E_\ell$ and $n_{i,t} \geq \lceil r_i(\ell) H \rceil \; \forall i \notin E_\ell$.

Thus, any fairness interval $I_j$ will guarantee anytime fairness for intervals $I_{j+1}$ and $I_{j+2}$ rounds with $j \geq 1$. □

We note that the sum of the time spent in fairness slots is the sum of times in the initial interval $I_0$ and all odd intervals. This time is at most $H(\lceil T/H+1 \rceil)\eta_{\lceil T/H+1 \rceil H}$. We define $\eta$ as $\lim_{d\to\infty} \eta_d$, which is given as $\eta = \max\left(\frac{\sum_i r_i}{K}, \max_i r_i\right)$.

*Discussion on the choice of $H$.* For our guarantees to hold, we require $H \geq 1/\epsilon$ with $\epsilon = \frac{1}{2\kappa} - \max_i r_i$, which can be precomputed as $\kappa$ and $r_i$ are known. To minimize the gap between algorithmic and optimal fairness slots (Lemma 4), $H$ should be as small as possible. A larger $H$ prolongs the initial phase for which anytime fairness guarantees do not satisfy. Thus, the optimal choice is $H = 1/\epsilon$.

Lemma 4. *The gap between the sum of fairness slots and $\eta T$ is at most $O(H)$. More formally,*

$$H(\lceil T/H+1 \rceil)\eta_{\lceil T/H+1 \rceil H} - \eta T = O(H). \tag{2}$$

Proof. We first note that the gap between $\eta_H$ and $\eta$ is mainly due to the ceilings involved in the expression of $\eta_H$, and since the ceiling is at most 1 away from the expression without ceiling, we have $\eta_H - \eta \leq \frac{N+K}{HK}$. Further, $H\eta_H$ increases with $H$. We have

$$\begin{aligned} & H(\lceil T/H+1 \rceil)\eta_{\lceil T/H+1 \rceil H} - \eta T \\ \leq \; & (T+2H)\eta_{T+2H} - \eta T \\ \leq \; & T(\eta_{T+2H} - \eta) + 2H\eta_{T+2H} \\ \leq \; & \frac{N+K}{K} + 2H\eta_{T+2H} = O(H) \quad \square \end{aligned}$$

With the bound on the number of fairness slots, we now provide the regret of the proposed algorithm in the following.

Theorem 2. *Suppose there is an online algorithm for CMAB with a cardinality constraint of $K$ that achieves $\beta$-regret of $O(L(T))$ for any $T$, where $L(T)$ is monotonically non-decreasing with $T$. Then, the proposed framework* `Fair-CMAB` *at any time $T$ achieves $(1-\eta)\beta$-regret of $O(\max(\beta H, L(T(1-\eta))))$ for any unknown $T$.*

Proof. The $(1-\eta)\beta$-regret for `Fair-CMAB` at time $T$ is given by:

$$\mathcal{R}^{\texttt{Fair-CMAB}}(T) = \beta(1-\eta)Tf(S^\star) - \sum_{t=1}^{T} f_t(S_t).$$

We decompose the sum of the rewards as:

$$\sum_{t=1}^{T} f_t(S_t) = \sum_{\text{Learner}} f_t(S_t) + \sum_{\text{Fairness}} f_t(S_t)$$

Given that the total time spent in the fairness times is at most $\eta T + O(H)$ (from Lemma 4), the time spent by the learner is at least

$$T(1-\eta) - O(H).$$

Given that the learner has any-time $\beta$-regret of $L(H)$ for any $H > 0$, we have

$$\sum_{\text{Learner}} f_t(S_t) \geq \beta((1-\eta)T - O(H))f(S^\star) - O(L(T(1-\eta) - O(H)))$$

Since, $\sum_{\text{Fairness}} f_t(S_t) \geq 0$,

$$\mathcal{R}^{\texttt{Fair-CMAB}}(T) \leq \beta O(H) f(S^\star) + \mathbb{O}(L(T(1-\eta) - O(H))).$$

Since $L(\cdot)$ is a non-decreasing function, we have $L(T(1-\eta) - O(H)) \leq L(T(1-\eta))$, thus, having the regret upper bounded as $\mathcal{R}^{\texttt{Fair-CMAB}}(T) = O(\max(\beta H, L(T(1-\eta))))$. □

**Remark 1.** *We note that we can further upper regret bound above to write $\mathcal{R}^{\texttt{Fair-CMAB}}(T) = O(\max(H, L(T)))$ for simplicity.*

*Regret Comparison with Existing Works.* It must be noted that most of the existing work provides weak fairness guarantees such as [37] which only guarantee asymptotic fairness, [56] which satisfy approximate fairness, [63] which guarantee fairness only at the end of $T$ rounds, and [38] ensures fairness only in expectation as opposed to our anytime fairness guarantee. Thus, their regret bounds are incomparable to ours. While [38] claims $O(1)$ regret, it includes an instance-dependent term $1/\Delta_{\min}^2$ in regret. Our algorithm enforces the strongest fairness notion with instance-independent regret. As far as the tightness of regret bounds is concerned because Lemma 4 establishes only a constant number of extra pulls for satisfying fairness, tightness on regret bounds naturally follows from the base learning algorithm.

*On the approximation ratio of fairness-aware algorithms.* The term $\eta T$ denotes the minimum time that any algorithm needs to satisfy fairness, derived from the makespan calculation in job scheduling. The offline algorithm incurs an additional approximation of $1-\eta$, so we have to make a comparison with the offline fairness-aware algorithm. Relative to the optimal fairness-aware algorithm, our regret follows that of the base learning algorithm, with only an additional constant regret $O(H)$ in fairness rounds (Lemma 4).

Since regret is measured against an optimal offline algorithm that maximizes reward, it is no longer linear, since the offline fairness-aware algorithm has an approximation guarantee of $\beta(1-\eta)$. This aligns with [46, Section 5. Cost of Fairness], where the approximation loss accounts for the fairness time. We establish sub-linear fairness-aware regret for combinatorial bandits by reducing it to the regret for fairness-unaware algorithm.

## 7 FAIR-CMAB: A FRAMEWORK FROM CMAB TO FAIR-CMAB ALGORITHM DISCUSSION AND APPLICATION

Since our proposed algorithm works on top of any online algorithm for stochastic CMAB, we provide an overview of different settings to

which the proposed algorithm can be applied. For semi-bandit feedback, Chen et al. [15] introduced a generalized framework, ComUCB1, a UCB1-like algorithm for any general monotone reward function. Kveton et al. [33] improved the guarantees by providing tight upper and lower bounds of $\sqrt{KNT\log(T)}$ and $K^{\frac{1}{2}}N^{\frac{1}{2}}T^{\frac{1}{2}}$, respectively. We focus here on bandit feedback settings, also summarized in Table 1.

*Linear Reward Functions:* Rejwan and Mansour [52] studied the problem of selecting $K$ out of $N$ arms with linear rewards and full-bandit feedback, proposing the CSAR (Combinatorial Successive Accepts and Rejects) algorithm.

*Restricted Non-linear Reward Functions:* Agarwal et al. [1, 3] studied the scenario where the reward function over the chosen $K$ arms is non-linear, and the individual arm rewards are unknown. Under the assumptions that the reward function is an element-wise, strictly increasing function of the individual rewards obtained by the constituent arms and the first-order stochastic dominance assumption, their proposed CMAB-SM algorithm achieves a regret bound of $\tilde{O}(K^{\frac{1}{2}}N^{\frac{1}{3}}T^{\frac{2}{3}})$. Under the assumption that good arms generate good actions, Agarwal et al. [2] develop DART, an elimination-based algorithm that deals with non-linear rewards. This algorithm is shown to achieve a regret bound of $\tilde{O}(K^{\frac{3}{2}}N^{\frac{1}{2}}T^{\frac{1}{2}})$.

*General Combinatorial Reward Functions:* Nie et al. [41], Fourati et al. [19] proposed algorithms for CMAB with monotone submodular reward functions with cardinality constraints, and non-monotone submodular reward functions, respectively, with full-bandit feedback. Nie et al. [43] extended these works to provide a framework in which discrete offline approximation algorithms for combinatorial optimization can be converted into sublinear $\alpha$-regret methods that only require bandit feedback. The framework only requires the offline algorithm to be robust to small errors in function evaluation. This framework assumes that the offline algorithm has an approximation ratio of $\alpha$, which was further extended in [20], to allow for an offline algorithm with $\alpha - \epsilon$ approximation ratio for any $\epsilon > 0$. This framework allows for improved regret results for monotone submodular reward functions with cardinality constraints, also studied in its special case in [21]. As another special case, this framework includes the regret bound of non-monotone submodular reward functions with cardinality constraints. These cases are also summarized in Table 1.

## 8 EXPERIMENTAL ANALYSIS

In this section, we provide an experimental evaluation of our proposed Fair-CMAB framework and compare it with the extension of Patil et al. [46] to combinatorial bandits which we termed as Fair-CMAB-delayed.

The algorithm in [46] ensures fairness by prioritizing arms that would otherwise violate anytime fairness, and provides guarantees for the case where at most one arm per round is pulled to ensure fairness. However, its regret analysis relies on learning during fairness pulls, which makes its extension to CMAB with bandit feedback and non-linear rewards challenging. In contrast, Fair-CMAB separates the fairness and learning phases, ensuring fairness in generalized CMAB while maintaining strong regret guarantees. To enable a fair comparison, Fair-CMAB-delayed integrates the approach of [46] by selecting up to $K$ arms with the highest fairness deficit (i.e., the lowest $n_{i,t} - r_i t$ at round $t+1$ if there is any arm $i$ with $n_{i,t} - r_i t < 0$ at round $t$) in fairness rounds, without using their reward updates. For $K = 1$, Fair-CMAB-delayed aligns with [46] but without incorporating learning (reward updates) of the arms during fairness pulls.

**Table 1: Table presents an application of Fair-CMAB for different online stochastic combinatorial optimization problems with full-bandit feedback. We use L=linear reward, NL=non-linear reward with certain assumptions as in the reference, S=sub-modular rewards; M=monotone set rewards, NM= non-monotone set rewards; CC=cardinality constraint of $K$. Column Ref gives the reference for the learner algorithm. Column $\alpha$ gives the approximation ratio for Fair-CMAB. For the regret, we skip the maximization with $H$ for simplicity.**

| CMAB Setting | Ref | $\alpha$ | Fair-CMAB Regret |
|---|---|---|---|
| L + CC | [52] | $1-\eta$ | $KN^{\frac{1}{2}}T^{\frac{1}{2}}$ |
| NL + CC | [1, 3] | $1-\eta$ | $K^{\frac{1}{2}}N^{\frac{1}{3}}T^{\frac{2}{3}}$ |
| NL + CC | [2] | $1-\eta$ | $K(KNT)^{\frac{1}{2}}$ |
| S + NM | [19] | $\frac{1}{2}(1-\eta)$ | $NT^{\frac{2}{3}}$ |
| S + M + CC | [21] | $1-\eta-\frac{1-\eta}{e}$ | $K^{\frac{2}{3}}N^{\frac{1}{3}}T^{\frac{2}{3}}$ |
| S + NM + CC | [20] | $\frac{1-\eta}{e}$ | $K^{\frac{2}{5}}N^{\frac{1}{5}}T^{\frac{4}{5}}$ |

We evaluated the Fair-CMAB for a general learner algorithm so that any setup of CMAB can be analyzed. In order to do that, the learner slots are not accounted for the updates of the pulled arms, while only fairness slots are accounted to capture how many times each arm has been pulled. This allows for general results of the proposed algorithm for any learner. However, we note that since the arms are also selected in the learner times, the fairness violations will be even lower. We compare Fair-CMAB and Fair-CMAB-delayed for $K > 1$ on the following two metrics:

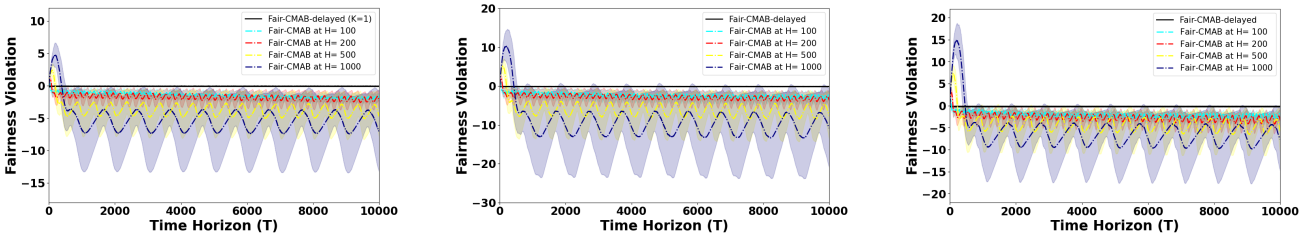
(1) **Fairness Violation:** We show the anytime fairness guarantees for both algorithms validating Theorem 1.
(2) **Fairness Pulls:** We compare the number of times each algorithm pulls the arms just to maintain the fairness constraints. For Fair-CMAB, these rounds correspond to the rounds in the initial intervals and the odd intervals. Whereas for Fair-CMAB-delayed, these rounds correspond to the rounds when any arm $i$ satisfies $r_i t - n_{i,t} < 1$. Note that the number of fairness pulls control the regret guarantees (approximation ratio and regret bound). Therefore we need lower number of fairness pulls while achieving zero fairness violations.

We further note that since the number of fairness pulls inherently captures the regret term, we do not necessarily need to compare the regret. The subsequent subsections outline our experimental setup, detailing the configurations and parameters used.

### 8.1 Experimental Setup

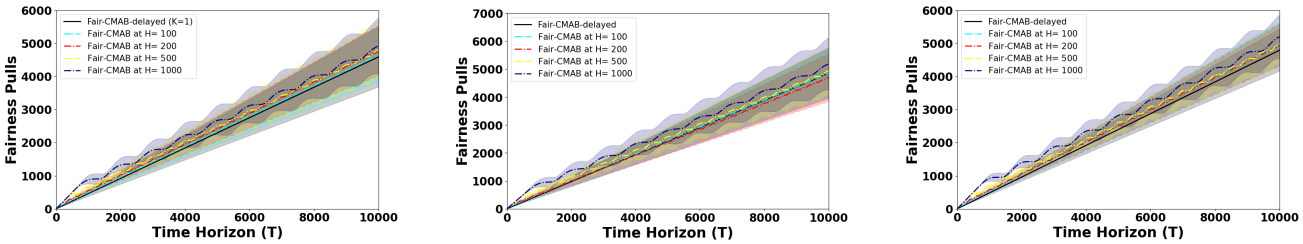
We evaluate the Fair-CMAB framework [2] over 10,000 episodes, on 100 instances, under three configurations: 1). $N = 10$, $K = 1$; 2).

[2]The code is available at: https://github.com/MultiFair-Bandits/Stochastic_Fair_CMAB/




**(a)** $N = 10, K = 1$ **(b)** $N = 10, K = 2$ **(c)** $N = 20, K = 5$

**Figure 2: Fairness violations over time.**




**(a)** $N = 10, K = 1$ **(b)** $N = 10, K = 2$ **(c)** $N = 20, K = 5$

**Figure 3: Fairness pulls over time.**

$N = 10, K = 2$; and 3). $N = 20, K = 5$. For each instance, the reward means $\mu$ were drawn from a uniform distribution $U(0.5, 1)$ and fairness thresholds $r_i$ were generated uniformly at random between $\left(0, \frac{1}{2K} - \epsilon\right)$. For all the instances, we kept $\epsilon = 0.01$ and $H \geq 100$. The experiments were conducted for $H \in \{100, 200, 500, 1000\}$, measuring fair pulls and fairness violations over time.

## 8.2 Fairness Violation versus Time Horizon

Fairness violations occur when an arm is pulled fewer times than its required fairness quota. For arm $i$ at time $t$, a violation occurs if $\lfloor r_i t \rfloor > n_{i,t}$ or $r_i t - n_{i,t} \geq 1$. We track the worst-performing arm using $\max_i(r_i t - n_{i,t})$. Across all configurations, `Fair-CMAB` maintains violations well below 0 not just 1, except for an initial interval of length $H\eta_H$, confirming Theorem 1. Due to its design, `Fair-CMAB`-delayed keeps fairness violations below 1, with a mean close to 0 (see Figure 2) for the single-arm ($K = 1$) setting.

`Fair-CMAB`-delayed satisfies anytime fairness guarantees (Figure 2b, 2c), ensuring fairness violations remain close to zero and below one. As $H$ increases, fairness violations exhibit greater variance, with consistently advance fair pulls. Conversely, as $H$ decreases, advance fair pulls diminish, approaching zero while maintaining the guarantee of no fairness violations after $H\eta_H$ rounds.

## 8.3 Number of Fair Pulls versus Time Horizon

While it is important that the algorithm never violates the fairness constraints, it is also important that the algorithm does just enough fair pulls to keep the regret lower. We compare the number of fair pulls for both algorithms. We note that as $H$ decreases, it is clear that `Fair-CMAB` has lower fairness pulls (Lemma 4) and thus better regret. This is also evident from our plots shown in Figure 3. We note that while `Fair-CMAB`-delayed seems competitive, it has no guarantees including that (i) at most $K$ arms will reach violation, (ii) no bound on the number of times such violations happen to bound the learner times for regret computation. In contrast, `Fair-CMAB` has provable fairness and regret guarantees.

## 9 CONCLUSION AND FUTURE WORK

In conclusion, our anytime Fair-CMAB framework ensures fairness in combinatorial bandit problems across various feedback models and reward functions. It offers strong theoretical guarantees on regret, with experimental results validating its effectiveness and adaptability. This framework provides a flexible, fair solution for a wide range of applications, demonstrating both fairness and learning efficiency. Since our regret guarantees are motivated by makespan literature defining the minimum fraction of time required to satisfy fairness constraint, the bounds cannot be trivially applied to the problems where arm pulling strategy is constrained by some structures such as matroid, paths, and spanning trees. Therefore, extending this work to handle constraints on action space is an interesting future direction.

## ACKNOWLEDGMENTS

This work is supported by Anusandhan National Research Foundation (ANRF)/ Science and Engineering Research Board (SERB)-Purdue University Overseas Visiting Doctoral Fellowship (Award No. SB/S9/Z-03/2017-XVII (2024)), the U.S. National Science Foundation under grant CCF-2149588, and ANRF under grant MTR/2022/000818.

## REFERENCES


[1] Mridul Agarwal, Vaneet Aggarwal, Christopher J Quinn, and Abhishek K Umrawal. 2021. Stochastic Top-$K$ Subset Bandits with Linear Space and Non-Linear Feedback. In *Algorithmic Learning Theory*. PMLR, 306–339.

[2] Mridul Agarwal, Vaneet Aggarwal, Abhishek Kumar Umrawal, and Chris Quinn. 2021. Dart: Adaptive accept reject algorithm for non-linear combinatorial bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 6557–6565.

[3] Mridul Agarwal, Vaneet Aggarwal, Abhishek K Umrawal, and Christopher J Quinn. 2022. Stochastic top k-subset bandits with linear space and non-linear feedback with applications to social influence maximization. *ACM/IMS Transactions on Data Science (TDS)* 2, 4 (2022), 1–39.

[4] Makis Arsenis and Robert Kleinberg. 2022. Individual Fairness in Prophet Inequalities. In *Proceedings of the 23rd ACM Conference on Economics and Computation*. 245–245.

[5] Jean-Yves Audibert, Sébastien Bubeck, and Gábor Lugosi. 2014. Regret in online combinatorial optimization. *Mathematics of Operations Research* 39, 1 (2014), 31–45.

[6] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning* 47 (2002), 235–256.

[7] Peter Auer and Ronald Ortner. 2010. UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica* 61, 1-2 (2010), 55–65.

[8] Solon Barocas, Moritz Hardt, and Arvind Narayanan. 2023. *Fairness and machine learning: Limitations and opportunities*. MIT press.

[9] Lilian Besson and Emilie Kaufmann. 2018. What Doubling Tricks Can and Can't Do for Multi-Armed Bandits. *HAL* (2018).

[10] Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. 2012. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning* 5, 1 (2012), 1–122.

[11] Niv Buchbinder, Kamal Jain, and Mohit Singh. 2014. Secretary problems via linear programming. *Mathematics of Operations Research* 39, 1 (2014), 190–206.

[12] Nicolo Cesa-Bianchi and Gábor Lugosi. 2006. *Prediction, learning, and games*. Cambridge university press.

[13] Nicolo Cesa-Bianchi and Gábor Lugosi. 2012. Combinatorial bandits. *J. Comput. System Sci.* 78, 5 (2012), 1404–1422.

[14] Wei Chen, Wei Hu, Fu Li, Jian Li, Yu Liu, and Pinyan Lu. 2016. Combinatorial multi-armed bandit with general reward functions. *Advances in Neural Information Processing Systems* 29 (2016).

[15] Wei Chen, Yajun Wang, and Yang Yuan. 2013. Combinatorial multi-armed bandit: General framework, results and applications. In *International conference on machine learning*. PMLR, 151–159.

[16] Wei Chen, Yajun Wang, Yang Yuan, and Qinshi Wang. 2016. Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *Journal of Machine Learning Research* 17, 50 (2016), 1–33.

[17] Richard Combes, Mohammad Sadegh Talebi Mazraeh Shahi, Alexandre Proutiere, et al. 2015. Combinatorial bandits revisited. *Advances in neural information processing systems* 28 (2015).

[18] Varsha Dani, Thomas P Hayes, and Sham M Kakade. 2008. Stochastic Linear Optimization under Bandit Feedback.. In *COLT*, Vol. 2. 3.

[19] Fares Fourati, Vaneet Aggarwal, Christopher Quinn, and Mohamed-Slim Alouini. 2023. Randomized greedy learning for non-monotone stochastic submodular maximization under full-bandit feedback. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 7455–7471.

[20] Fares Fourati, Mohamed-Slim Alouini, and Vaneet Aggarwal. 2024. Federated Combinatorial Multi-Agent Multi-Armed Bandits. In *Forty-first International Conference on Machine Learning*.

[21] Fares Fourati, Christopher John Quinn, Mohamed-Slim Alouini, and Vaneet Aggarwal. 2024. Combinatorial stochastic-greedy bandit. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 12052–12060.

[22] Yi Gai, Bhaskar Krishnamachari, and Rahul Jain. 2010. Learning multiuser channel allocations in cognitive radio networks: A combinatorial multi-armed bandit formulation. In *2010 IEEE Symposium on New Frontiers in Dynamic Spectrum (DySPAN)*. IEEE, 1–9.

[23] Yi Gai, Bhaskar Krishnamachari, and Rahul Jain. 2012. Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations. *IEEE/ACM Transactions on Networking* 20, 5 (2012), 1466–1478.

[24] Aditya Gopalan, Shie Mannor, and Yishay Mansour. 2014. Thompson sampling for complex online problems. In *International conference on machine learning*. PMLR, 100–108.

[25] Ronald L. Graham. 1969. Bounds on multiprocessing timing anomalies. *SIAM journal on Applied Mathematics* 17, 2 (1969), 416–429.

[26] Riccardo Grazzi, Arya Akhavan, John IF Falk, Leonardo Cella, and Massimiliano Pontil. 2022. Group meritocratic fairness in linear contextual bandits. *Advances in Neural Information Processing Systems* 35 (2022), 24392–24404.

[27] Shivam Gupta, Ganesh Ghalme, Narayanan C Krishnan, and Shweta Jain. 2023. Efficient algorithms for fair clustering with a new notion of fairness. *Data Mining and Knowledge Discovery* 37 (2023), 1959–1997.

[28] András György, Tamás Linder, Gábor Lugosi, and György Ottucsák. 2007. The On-Line Shortest Path Problem Under Partial Monitoring. *Journal of Machine Learning Research* 8, 10 (2007).

[29] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. *Advances in neural information processing systems* 29 (2016).

[30] Faisal Kamiran and Toon Calders. 2009. Classifying without discriminating. In *2nd international conference on computer, control and communication*. IEEE, 1–6.

[31] Branislav Kveton, Zheng Wen, Azin Ashkan, Hoda Eydgahi, and Brian Eriksson. 2014. Matroid bandits: fast combinatorial optimization with learning. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*. 420–429.

[32] Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvári. 2015. Combinatorial cascading bandits. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*. 1450–1458.

[33] Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. 2015. Tight regret bounds for stochastic combinatorial semi-bandits. In *Artificial Intelligence and Statistics*. PMLR, 535–543.

[34] Tor Lattimore, Branislav Kveton, Shuai Li, and Csaba Szepesvari. 2018. Toprank: A practical algorithm for online stochastic ranking. *Advances in Neural Information Processing Systems* 31 (2018).

[35] Tor Lattimore and Csaba Szepesvári. 2020. *Bandit algorithms*. Cambridge University Press.

[36] Adam Lechowicz, Rik Sengupta, Bo Sun, Shahin Kamali, and Mohammad Hajiesmaili. 2024. Time Fairness in Online Knapsack Problems. In *The Twelfth International Conference on Learning Representations*.

[37] Fengjiao Li, Jia Liu, and Bo Ji. 2019. Combinatorial sleeping bandits with fairness constraints. *IEEE Transactions on Network Science and Engineering* 7, 3 (2019), 1799–1813.

[38] Qingsong Liu, Weihang Xu, Siwei Wang, and Zhixuan Fang. 2022. Combinatorial bandits with linear constraints: Beyond knapsacks and fairness. *Advances in Neural Information Processing Systems* 35 (2022), 2997–3010.

[39] Robert McNaughton. 1959. Scheduling with deadlines and loss functions. *Management science* 6, 1 (1959), 1–12.

[40] Rad Niazadeh, Negin Golrezaei, Joshua R Wang, Fransisca Susan, and Ashwinkumar Badanidiyuru. 2021. Online learning via offline greedy algorithms: Applications in market design and optimization. In *Proceedings of the 22nd ACM Conference on Economics and Computation*. 737–738.

[41] Guanyu Nie, Mridul Agarwal, Abhishek Kumar Umrawal, Vaneet Aggarwal, and Christopher John Quinn. 2022. An explore-then-commit algorithm for submodular maximization under full-bandit feedback. In *Uncertainty in Artificial Intelligence*. PMLR, 1541–1551.

[42] Guanyu Nie, Vaneet Aggarwal, and Christopher John Quinn. 2024. Stochastic $k$-Submodular Bandits with Full Bandit Feedback. *arXiv e-prints* (2024), arXiv–2412.

[43] Guanyu Nie, Yididiya Y Nadew, Yanhui Zhu, Vaneet Aggarwal, and Christopher John Quinn. 2023. A framework for adapting offline algorithms to solve combinatorial multi-armed bandit problems with bandit feedback. In *International Conference on Machine Learning*. PMLR, 26166–26198.

[44] Alessandro Nuara, Francesco Trovo, Nicola Gatti, and Marcello Restelli. 2018. A combinatorial-bandit algorithm for the online joint bid/budget optimization of pay-per-click advertising campaigns. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[45] Manjish Pal, Subham Pokhriyal, Sandipan Sikdar, and Niloy Ganguly. 2023. Ensuring generalized fairness in batch classification. *Scientific Reports* 13, 1 (2023), 18892.

[46] Vishakha Patil, Ganesh Ghalme, Vineet Nair, and Yadati Narahari. 2021. Achieving fairness in the stochastic multi-armed bandit problem. *The Journal of Machine Learning Research* 22, 1 (2021), 7885–7915.

[47] Mohammad Pedramfar and Vaneet Aggarwal. 2025. Stochastic submodular bandits with delayed composite anonymous bandit feedback. *IEEE Transactions on Artificial Intelligence* (2025).

[48] Subham Pokhriyal, Shweta Jain, Ganesh Ghalme, Swapnil Dhamal, and Sujit Gujar. 2024. Simultaneously Achieving Group Exposure Fairness and Within-Group Meritocracy in Stochastic Bandits. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*. 1576–1584.

[49] Lijing Qin, Shouyuan Chen, and Xiaoyan Zhu. 2014. Contextual combinatorial bandit and its application on diversified online recommendation. In *Proceedings of the 2014 SIAM International Conference on Data Mining*. SIAM, 461–469.

[50] Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. 2008. Learning diverse rankings with multi-armed bandits. In *Proceedings of the 25th international conference on Machine learning*. 784–791.

[51] Anshuka Rangi and Massimo Franceschetti. 2018. Multi-Armed Bandit Algorithms for Crowdsourcing Systems with Online Estimation of Workers' Ability.. In *AAMAS*. 1345–1352.

[52] Idan Rejwan and Yishay Mansour. 2020. Top-$k$ combinatorial bandits with full-bandit feedback. In *Algorithmic Learning Theory*. PMLR, 752–776.

[53] Herbert Robbins. 1952. Some aspects of the sequential design of experiments. *Bull. Amer. Math. Soc.* 58, 5 (1952), 527–535.
[54] Yaniv Romano, Stephen Bates, and Emmanuel Candes. 2020. Achieving equalized odds by resampling sensitive attributes. *Advances in neural information processing systems* 33 (2020), 361–371.
[55] Aleksandrs Slivkins et al. 2019. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning* 12, 1-2 (2019), 1–286.
[56] Juaren Steiger, Bin Li, and Ning Lu. 2022. Learning from delayed semi-bandit feedback under strong fairness guarantees. In *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 1379–1388.
[57] Matthew Streeter and Daniel Golovin. 2008. An online algorithm for maximizing submodular functions. *Advances in Neural Information Processing Systems* 21 (2008).
[58] Matthew Streeter, Daniel Golovin, and Andreas Krause. 2009. Online learning of assignments. In *Proceedings of the 22nd International Conference on Neural Information Processing Systems*. 1794–1802.
[59] Michal Valko. 2016. *Bandits on graphs and structures*. Ph.D. Dissertation. École normale supérieure de Cachan-ENS Cachan.
[60] Lequn Wang, Yiwei Bai, Wen Sun, and Thorsten Joachims. 2021. Fairness of exposure in stochastic bandits. In *International Conference on Machine Learning*. 10686–10696.
[61] Qinshi Wang and Wei Chen. 2017. Improving regret bounds for combinatorial semi-bandits with probabilistically triggered arms and its applications. *Advances in Neural Information Processing Systems* 30 (2017).
[62] Raymond Chi-Wing Wong, Ada Wai-Chee Fu, and Ke Wang. 2003. MPIS: maximal-profit item selection with cross-selling considerations. In *Third IEEE International Conference on Data Mining*. IEEE, 371–378.
[63] Huanle Xu, Yang Liu, Wing Cheong Lau, and Rui Li. 2020. Combinatorial Multi-Armed Bandits with Concave Rewards and Fairness Constraints. In *International Joint Conference on Artificial Intelligence*. IJCAI, 2554–2560.
[64] Julian Zimmert, Haipeng Luo, and Chen-Yu Wei. 2019. Beating stochastic and adversarial semi-bandits optimally and simultaneously. In *International Conference on Machine Learning*. PMLR, 7683–7692.