Bandit Learning for Proportionally Fair Allocations, Simply Solved

Tianyu Wang[*] Cynthia Rudin[†]

Abstract

A proportionally fair allocation aims to distribute resources, where each individual is given a share according to that individual's utility for that resource (e.g., need, request, yield, etc.) so that the overall utility is proportionally maximized. In this paper, we seek to find a proportionally fair allocation under noisy bandit feedback. We introduce two new algorithms for this problem: Algorithm Fairly-Greedy, which uses a simple greedy allocation, and Algorithm Proportional Catch-Up, which allocates resources whose arms have not been pulled enough relative to their utilities. We show that the allocation learned by our algorithms is close to the optimal allocation with high probability. We apply our algorithms to synthetic data and real data, and show that they outperform existing methods in tasks of proportionally-fair allocation.

1 Introduction

Resource allocation (of goods, of services) is a central problem in many areas of society: social services need to be allocated to those who need them, computer servers need to schedule jobs efficiently, and budgets of organizations need to be allocated effectively. In this paper, we study an important bandit resource allocation problem that occurs often in practice. In this new setting, *new resources become available over time, and our goal is to allocate them proportionally fairly*, where proportional fairness means that no alternative allocation can increase the overall proportionally-weighted utility (Kelly et al., 1998). The learning environment consists of K individuals and T units of resource that become available one at a time. The task of an agent is to distribute the resource fairly to the K individuals, where the utility of the resource to each of the individuals is unknown (to be learned). At each time, the agent needs to give one unit of resource to one of the K individuals. After giving out a unit of resource to individual i, the agent can observe the response from i, which reflects the utility of the resource to the individual. Once a unit of resource is given to an individual, it can never be taken back.

This model is a viable abstraction for many real-world scenarios, which are highly related to recommender systems. Examples include:

1) Advertising: Under budget constraints, a college training program wants to allocate its available ads among a distribution of individuals with different age demographics, based on how valuable it estimates the ad to be to these groups. The college training program estimates that the ad might appeal most to 17-25 year-olds, less so to 26-40 year-olds, and other groups even less so. One demographic group i would initially be allocated some ads, and from there, the college receives feedback at time t consisting of successes (e.g., training program enrollment increases) or indicators (e.g., click rate) of the strength of the ad $Y_{t,i}$. An algorithm then adjusts the allocation

[*] wangtianyu@fudan.edu.cn

[†] cynthia@duke.edu

of ads for the next time period, and continues allocating proportional to the estimated values of the ads for the various age groups.

2) Community resource distribution: A city is allocating resources to its various community training programs. μ is how many job training resources each community training program needs in order to successfully provide their community with jobs, which is not known in advance. Each community i is allocated some job training resources at time t, and from there, we receive feedback $Y_{t,i}$ which could be determiners or indicators of job success, such as the counts of individuals who are now employed after training, or their wage increase, or test scores. After this feedback, we continue allocating resources proportionally to the needs of the community training programs.

3) Server job scheduling. A server is serving requests from K clients, and can serve only one client at a time. Each client has a different resource consumption rate that is not known in advance. The server need to allocate the computing resources to the K clients so that the total throughput is maximized.

The general problem discussed above seems to naturally fit into the framework of multi-armed bandits because it has information revealed over time; however, typical bandit algorithms aim to minimize regret, which means they typically aim to balance exploration and exploitation to find a single best arm that yields the maximum reward. In this proportional fairness setting, we do not aim to find a single arm. The goal instead is to create an allocation over all K arms that maximizes proportional fairness. This distinction is important, as it means that no classical bandit algorithms apply to this setting. In this work, we propose novel bandit algorithms for learning these allocations. The algorithms directly allocates resources according to an estimate of each individual's utility, which is gathered over time.

An interesting aspect of our analysis involves the reward function. In particular, the objective of proportional fairness ensures a "diminishing returns" property, namely that the marginal gain of one individual repeatedly receiving resources is decreasing. Intuitively, this submodular objective is aligned with fairness: it ensures that resources will be more evenly distributed. In the analysis, we use this property of the proportional fairness objective to obtain an approximation-rate type result. Specifically, we show that with high probability, the utility of the allocation learned by *simple algorithms* at step T is within $\mathcal{O}(\frac{1}{\sqrt{T}})$ of the optimal proportionally-fair utility.

Related Works

Our paper is related to previous works on proportional fairness. One major motivation for proportional fairness originated from job scheduling in computer clusters (Kelly et al., 1998). Since then, a line of work has studied the properties of proportional fairness (Kushner and Whiting, 2004; Bonald et al., 2006). Specifically, Bonald et al. (2006) studied game theoretical properties of proportional fairness. Kushner and Whiting (2004) also proposed an algorithm for finding proportionally fair allocations for divisible resources. Their paper uses a discretized ODE algorithm for learning the proportional fair allocations, whereas we use a greedy approach or a proportional catch-up method for proportionally fair allocation learning under bandit feedback. Additionally, the result in our work is stronger in the sense that a finite time analysis is given, whereas the analysis of Kushner and Whiting (2004) gives only an asymptotic guarantee. In the machine learning community, many authors have studied fairness from different angles (Kearns and Roth, 2019; Tsamados et al., 2021). For instance, Agarwal et al. (2019) studied fair regression, Krishnaswamy et al. (2021) studied fair classification and Chen et al. (2019) studied proportionally fair clustering. However, none of the above works solve the problem of learning proportionally fair allocations under bandit feedback.

Another line of related work comes from the multi-armed bandit problem. Bandit problems date back to at least Thompson (1933), and have been studied by many authors. One of the the most popular approaches to the stochastic bandit problem is the Upper Confidence Bound (UCB) algorithms (Robbins, 1952; Lai and Robbins, 1985; Auer, 2002), which has various extensions (Srinivas et al., 2010; Abbasi-Yadkori et al., 2011; Agrawal and Goyal, 2012; Bubeck and Slivkins, 2012; Seldin and Slivkins, 2014). Specifically, some work uses KL-divergence to construct the confidence bound (Lai and Robbins, 1985; Garivier and Cappé, 2011; Maillard et al., 2011), or include variance estimates within the confidence bound (Audibert et al., 2009; Auer and Ortner, 2010). The UCB algorithm and its variations are also used in other settings, including the contextual bandit setting (e.g., Li et al., 2010; Krause and Ong, 2011; Slivkins, 2014), and the stochastic combinatorial bandit setting (Chen et al., 2013, 2016). Parallel to the stochastic setting, studies on the adversarial bandit problem form another line of literature. Since randomized weighted majorities (Littlestone and Warmuth, 1994), exponential weights remains a top strategy for adversarial bandits (Auer et al., 1995; Cesa-Bianchi et al., 1997; Auer et al., 2002). The exponential weights method is a special of the Follow-The-Regularized Leader (FTRL) or mirror descent with the Shannon's entropy as its regularizer (Shalev-Shwartz et al., 2011). Recently, Zimmert and Seldin (2021); Zimmert et al. (2019) show that the FTRL framework solves both the stochastic and adversarial setting optimally.

Recently, there has been an increasing trend in studying multi-armed bandit learning problems with fairness considerations. Joseph et al. (2016) was among the first who considered fairness in multi-armed bandits and introduced the Knows What It Knows model. Several authors have also studied multi-armed bandits with fairness considerations, including Patil et al. (2020), where a notion of fairness tolerance is used as a fairness metric, and (Li et al., 2019), where fairness is included into consideration for combinatorial bandits.

Perhaps the most related works are the ones by Talebi and Proutiere (2018) and Wang et al. (2021), where proportional fairness has been considered. In (Talebi and Proutiere, 2018; Wang et al., 2021), exploration methods such as Upper Confidence Bound and/or Thompson Sampling are used to solve such problems. In this paper, unlike the previous work by Wang et al. (2021), we show that, in the vanilla proportional fairness setting with unknown utilities Kelly et al. (1998), UCB-type exploration-exploitation balancing is not necessary. More specifically, neither Fairly-Greedy or Proportional Catch-Up performs exploration-exploration balancing. Yet they both achieve the optimal $\mathcal{O}(\sqrt{T})$ regret rate (Singh and Joachims, 2018). In addition, as shown by empirical studies, both Fairly-Greedy and Proportional Catch-Up can outperform UCB-type algorithms (Singh and Joachims, 2018; Wang et al., 2021), in the task of learning vanilla proportional fair allocations.

2 Background and Setting

The goal is to learn an allocation, which is a distribution of resources among individuals. Let there be T unit of a (divisible) resource and K individuals. An allocation is a vector $(x_1, x_2, \cdots, x_K)$ so that $x_i \geq 0$ and $\sum_{i=1}^{K} x_i = T$. In the past, variance allocation schemes have been studied (e.g., Brandt et al., 2016). Among them, the concept of proportionally fair allocation comes from game theory for job scheduling, where the original motivation is to efficiently allocate computing resources according to the priority of jobs, and at the same time, ensure that no job is starving (e.g., Kushner and Whiting, 2004). The formal definition is stated in Definition 1.

Definition 1. *Let $\mu_i \in (0, 1]$ be the utility for individual i. An allocation $x^* = (x_1^*, x_2^*, \cdots, x_K^*)$ (with $\sum_i x_i^* = T$) is called proportionally fair if, for any $x' = (x_1', x_2', \cdots, x_K')$ (with $\sum_i x_i' = T$),*

$$\sum_i \mu_i \frac{x_i' - x_i^*}{x_i^*} \leq 0. \tag{1}$$

Intuitively, an allocation is proportionally fair if no alternative allocation can increase the proportionally weighted overall utility. If individual i is given many resources in the optimal allocation (i.e., x_i^* is large), changing the amount of resources given to i would not affect the weighted sum in (1) very much. On the contrary, if individual i is given few resources in the optimal allocation (i.e., x_i^* is small), changing the amount of resources given to i would drastically affect the weighted sum in (1).

An allocation that satisfies the fairness property in (1) is the one where each individual gets a share proportional to their utility ($x_i^* \propto \mu_i$); such allocation is *proportionally fair*. This phenomenon is found in Proposition 1. Proposition 1 is an established result from the classic study of allocation theory, and a proof is provided for completeness.

Proposition 1. *The allocation where* $x_i^* = \frac{T\mu_i}{\sum_j \mu_j}$ *solves*

$$\max_{x_i, i\in[K]} \sum_i \mu_i \log x_i, \qquad \textit{subject to} \qquad \sum_i x_i = T, \tag{2}$$

and this allocation is proportionally fair, meaning that it satisfies (1)*. In particular, this optimal allocation satisfies*

$$\frac{x_i^*}{x_j^*} = \frac{\mu_i}{\mu_j}. \tag{3}$$

Proof of Proposition 1. Let $\lambda \geq 0$ be the Lagrangian multiplier for the optimization problem in (2). The resulting convex problem is

$$\min_{\lambda, x_i, i\in[K]} - \sum_i \mu_i \log x_i + \lambda \left(\sum_i x_i - T \right).$$

First order stationary condition gives $\frac{\mu_i}{x_i^*} = \lambda$ and $\sum_i x_i^* = T$. This gives $x_i^* = \frac{T\mu_i}{\sum_j \mu_j}$, which is a solution that satisfies all constraints and KKT conditions. In addition, we can verify that for any other allocation $(x_1', x_2', \cdots, x_K')$ with $\sum_i x_i' = T$,

$$\sum_i \mu_i \left(\frac{x_i'}{x_i^*} - 1 \right) = \sum_i \mu_i \left(\frac{x_i' \sum_j \mu_j}{T\mu_i} - 1 \right) = 0,$$

which means this allocation is proportionally fair. □

In other words, the solution to (2) satisfies (1). This proposition allows us to transform the constraint satisfaction problem in (1) to the optimization problem in (2). Due to concavity of (2), the problem is usually simplified after this transformation.

From the expression (2), it holds that giving more resources to rich individuals generates less marginal gain, which justifies the fairness from another perspective. In addition to justifying that an allocation satisfying (1) is indeed fair in a proportional sense, this proposition also suggests a concise and concrete objective for learning proportionally fair allocation, that is, to maximize the objective in Eq. (2). For indivisible resources, x_i needs to take integer values. In such cases, we would use the best integer-valued solution for (2) to define a proportionally fair allocation for indivisible goods.

Bandit Learning for Proportionally Fair Allocations Problem Setting

With the concepts of proportional fairness in mind, we now describe a bandit learning protocol for proportionally-fair resource distribution. Henceforth, we will use terminology from the multi-armed bandit literature unless otherwise noted. As an example, when we say “pull arm i,” this is equivalent to “allocate a unit of resource to individual i.”

There are K arms (individuals) in the environment, each having their own reward (utility) distribution that is unknown and supported on $(0,1]$. The learning process repeats for T rounds, where at each round, the agent needs to pull an arm (allocate a unit of resource). After arm i is played, a reward sample of arm i is revealed, and the agent proceeds to the next round. Formally, in round t, the agent plays arm $j_t \in [K]$, and observes $y_{j_t,t}$, where $y_{j_t,t}$ is a sample of $Y_{j_t,t} \in (0,1]$ and $\mathbb{E}[Y_{i,t}] = \mu_i$ for all $i \in [K]$ and $t \in [T]$. This feedback structure is typical for multi-armed bandit problems. Unlike traditional multi-armed bandit tasks, the goal here is to learn a proportionally fair allocation (Definition 1), instead of minimizing regret.

3 Algorithms

By Proposition 1, we can maximize $\sum_j \mu_i \log x_i$ for the purpose of learning a proportionally fair allocation. In this bandit learning environment, we do not know μ_i and need to play an arm at each round. Therefore, the first step is to estimate μ_i. We use $n_{t,i} = \sum_{s=1}^{t} \mathbb{I}_{[j_s=i]}$ to denote the number of times i is played up to time t. The estimator for μ_i (at time t) is defined, as usual, as $\widehat{\mu}_{t,i} = \frac{\sum_{s=1}^{t} y_{j_s,s} \mathbb{I}_{[j_s=i]}}{n_{t,i}}$.

At time t, the current allocation is $(n_{t,1}, n_{t,2}, \cdots, n_{t,K})$, and the corresponding estimated objective value that we are maximizing is

$$\sum_i \widehat{\mu}_{t,i} \log n_{t,i}. \tag{4}$$

A natural strategy is to play an arm to greedily increment (4). More specifically, we play j_t such that

$$j_t \in \arg\max_j \sum_i \widehat{\mu}_{t-1,i} \log \left(\frac{n_{t-1,i} + \mathbb{I}_{[j=i]}}{n_{t-1,i}} \right). \tag{5}$$

This natural and simple strategy is summarized in Algorithm 1, which we call the Fairly-Greedy Algorithm.

Algorithm 1 Fairly-Greedy

```
Input: Time horizon (total unit of resources): $T$.
Initialization: Play each arm once and initialize $n_{0,i}$ and $\widehat{\mu}_{0,i}$.
for $t = 1, 2, \ldots, T$ do
    Play $j_t$ as defined in (5).
    Observe $y_{j_t,t}$ and update $n_{t,i}$ and $\widehat{\mu}_{t,i}$ accordingly.
end for
```

Another strategy is to ensure that the allocation received by an individual is always approximately proportional to its estimated utility. This strategy follows the intuition from Proposition 1. If we find that an arm has not been pulled enough relative to its estimated reward, we pull it. This algorithm is called Proportional Catch-Up, since it "plays catch-up" with arms that have fallen behind in their pulls, so that $\frac{n_{s,i}}{n_{s,j}} \approx \frac{\widehat{\mu}_{s,i}}{\widehat{\mu}_{s,j}}$ for all s, i, j. Here we use $\frac{n_{s,i}}{n_{s,j}} \approx \frac{\widehat{\mu}_{s,i}}{\widehat{\mu}_{s,j}}$ to denote that $\left(\frac{n_{s,i}}{n_{s,j}} - \frac{\widehat{\mu}_{s,i}}{\widehat{\mu}_{s,j}} \right) \left(\frac{n_{s,i}+1}{n_{s,j}} - \frac{\widehat{\mu}_{s,i}}{\widehat{\mu}_{s,j}} \right) \le 0$ or $\left(\frac{n_{s,i}}{n_{s,j}} - \frac{\widehat{\mu}_{s,i}}{\widehat{\mu}_{s,j}} \right) \left(\frac{n_{s,i}-1}{n_{s,j}} - \frac{\widehat{\mu}_{s,i}}{\widehat{\mu}_{s,j}} \right) \le 0$.

4 Analysis

In this section, we provide analysis for Algorithms 1 and 2. The analysis for Algorithms 1 uses the submodular property of the objective in (2), and the analysis for Algorithm 2 uses the observation

Algorithm 2 Proportional Catch-Up

```
1: Input: Time horizon (total unit of resources): $T$.
2: Initialization: Play each arm once and initialize $n_{0,i}$ and $\widehat{\mu}_{0,i}$.
3: Initialize loop counter $s = 1$.
4: repeat
5:     With probability $\epsilon_s = s^{-1/4}$, play an arm uniformly at random, and skip line 6. Otherwise, execute line 6.
6:     Play arms so that $\frac{n_{s,i}}{n_{s,j}} \approx \frac{\widehat{\mu}_{s,i}}{\widehat{\mu}_{s,j}}$, for all $i, j \in [K]$.
   /* For implementation, one can use the for-loop in Algorithm 3 to approximate this step. */
7:     Increase counter $s = s + 1$, and update the estimators for $\mu_i$ for all $i$.
8: until All $T$ units of resources are exhausted.
```

Algorithm 3

```
for $(i, j) \in [K] \times [K]$ and $i \neq j$ do
    If $\frac{n_{t,i}}{n_{t,j}} > \frac{\widehat{\mu}_{t,i}}{\widehat{\mu}_{t,j}}$, play $j_t = j$. Otherwise, play $j_t = i$.
end for
```

that the optimal proportionally-fair allocation satisfies (3).

4.1 Analysis for Fairly-Greedy

To state results in the language of greedy algorithms, we first discuss functions defined over multi-sets. For items (arms) $[K]$, let $\mathbb{N}^{[K]}$ denote all multi-sets defined over $[K]$. Any $A \in \mathbb{N}^{[K]}$ can be represented by a tuple $A = (n_1^A, n_2^A, \cdots, n_K^A)$, where $n_i^A \in \mathbb{N}$ denotes the number of repetitions of item i in A. For any $A, B \in \mathbb{N}^{[K]}$, we write $A \subseteq B$ (A belongs to B) if $n_i^A \leq n_i^B$ for all $i \in [K]$. For any $A \in \mathbb{N}^{[K]}$, we define the union operation between multi-set A and a singleton as $A \cup \{i\} := (n_1^A, n_2^A, \cdots, n_i^A + 1, \cdots, n_K^A)$. The cardinality of a multi-set A is defined as $|A| = \sum_{i=1}^K n_i^A$. For any multi-set A and B, the multi-set complement is defined as $A \setminus B = \big(\max\{0, n_1^A - n_1^B\}, \max\{0, n_2^A - n_2^B\}, \cdots, \max\{0, n_K^A - n_K^B\}\big)$, and the multi-set union is defined as $A \cup B = \big(\max\{n_1^A, n_1^B\}, \max\{n_2^A, n_2^B\}, \cdots, \max\{n_K^A, n_K^B\}\big)$. Note that the union between two multi-sets and the union between a multi-set and a singleton are different, and the $\cup$ notation is overloaded. For a multi-set $A = (n_1^A, \cdots, n_K^A)$, we use $\sum_{i \in A}$ to denote the summation over all $j \in [K]$ such that each j is repeated for n_j^A times. In other words, we define $\sum_{i \in A} f(i) = \sum_{k=1}^K \sum_{i_k=1}^{n_k^A} f(i_k)$ for any function f defined over $[K]$. With these multi-set notations, we say function $f : \mathbb{N}^{[K]} \to \mathbb{R}$ is:

- increasing if $f(S \cup \{i\}) \geq f(S)$ for any $S \in \mathbb{N}^{[K]}$ and $i \in [K]$,
- and submodular if $f(X \cup \{x\}) - f(X) \geq f(Z \cup \{x\}) - f(Z)$ for any $X \subseteq Z \in \mathbb{N}^{[K]}$ and $x \in [K]$.

With these notions of increasing and submodular functions for multi-sets, we proceed to present the theoretical guarantees.

Proposition 2. *If the rewards are supported on $[c, 1]$ ($c \in (0, 1)$), then it holds that $n_{t,i} \geq \frac{ct}{4(K-1)+c}$ for all $i \in [K]$ and all $t \in \mathbb{N}_{>0}$.*

That is, every arm has a guarantee on how often it is pulled, which relates directly to our goal of fair allocation of arm pulls.

Proof. Suppose $\frac{n_{t,j}}{n_{t,i}} > \frac{4}{c}$ for some t and i, j. Then:

$$
\begin{aligned}
& c\left[\log(n_{t,i}+1) - \log n_{t,i}\right] \overset{\textcircled{1}}{\geq} \frac{c}{n_{t,i}} - \frac{c}{2n_{t,i}^2} \\
& \overset{\textcircled{2}}{\geq} \frac{c}{2n_{t,i}} \overset{\textcircled{3}}{>} \frac{1}{n_{t,j}} \overset{\textcircled{4}}{\geq} \log(n_{t,j}+1) - \log n_{t,j}
\end{aligned}
\tag{6}
$$

which ① uses $\log(1+x) \geq x - \frac{x^2}{2}$ for $x \geq 0$, ② uses that $\frac{c}{2n_{t,i}^2} \leq \frac{c}{2n_{t,i}}$ for all $n_{t,i} \geq 1$, ③ uses that $\frac{n_{t,j}}{n_{t,i}} > \frac{2}{c}$, ④ uses $x \geq \log(1+x)$ for $x \in \mathbb{R}$. Since the rewards are supported on $[c, 1]$, the empirical means $\widehat{\mu}_{t,i}$ are also supported on $[c, 1]$. Thus from (6), it holds that

$$
\begin{aligned}
& \widehat{\mu}_{t,i}\left[\log(n_{t,i}+1) - \log n_{t,i}\right] \\
\geq\ & c\left[\log(n_{t,i}+1) - \log n_{t,i}\right] && \text{(Since } \widehat{\mu}_{t,i} \geq c\text{)} \\
\geq\ & \log(n_{t,j}+1) - \log n_{t,j} && \text{(by Eq. 6)} \\
\geq\ & \widehat{\mu}_{t,j}\left[\log(n_{t,j}+1) - \log n_{t,j}\right]. && \text{(Since } \widehat{\mu}_{t,j} \leq 1\text{)}
\end{aligned}
$$

This implies that j will be played instead of i whenever $\frac{n_{t,i}}{n_{t,j}} > \frac{2}{c}$, and thus $\frac{n_{t,i}}{n_{t,j}}$ will decrease (since $n_{t,j}$ is in the denominator) until $\frac{n_{t,i}}{n_{t,j}} \leq \frac{2}{c}$, and thus can never exceed $\frac{4}{c}$ (since the ratio $\frac{n_{t,i}}{n_{t,j}}$ can never double when the numerator is incremented by 1). Therefore $\frac{n_{t,i}}{n_{t,j}} \leq \frac{4}{c}$ for all t, i, j. Since $\sum_{j=1}^{K} n_{t,j} = t$ for any t, we have, for any i':

$$
\begin{aligned}
t = \sum_{i=1}^{K} n_{t,i} &\leq n_{t,i'} + \sum_{j \neq i'} \frac{4}{c} n_{t,i'} \\
&= \left(\frac{4(K-1)}{c} + 1\right) n_{t,i'}, \quad \forall i' \in [K],
\end{aligned}
$$

which means $n_{t,i'} \geq \frac{ct}{4(K-1)+c}$ for all $i' \in [K]$. Since we chose i' arbitrarily, we have that $n_{t,i} \geq \frac{ct}{4(K-1)+c}$ for all i. □

From now on, we use the following notation. For a multi-set S, let $n_i(S) = \sum_{x \in S} \mathbb{I}_{[x=i]}$ be the number of occurrences of i in S. The function f is defined as, unless otherwise noted, $f(S) = \sum_{i \in [K]} \mu_i \log(n_i(S))$. Also, define $\widehat{f}_t(S) = \sum_{i \in [K]} \widehat{\mu}_{t,i} \log(n_i(S))$. With this f and $\widehat{f}_t$, define $f_S(i) = f(S \cup \{i\}) - f(S)$ and $\widehat{f}_{S,t}(i) = \widehat{f}_t(S \cup \{i\}) - \widehat{f}_t(S)$. Also, we use S_t to denote the multi-set of allocation learned at time t. In other words, $S_t = (n_{t,1}, n_{t,2}, \cdots, n_{t,K})$. We start with the following proposition (e.g., Bach et al., 2013), which is a standard property for submodularity.

Proposition 3. *If f is monotone (increasing) and submodular,*

$$
f(V) \leq f(S) + \sum_{x \in V \setminus S} f_S(x) \textit{ for all } S, V \in \mathbb{N}^{[K]}.
$$

This proposition is a multi-set version of the basic property for submodular set functions. The proof is that if f is monotone increasing, then $f(V) \leq f(S \cup V)$. If f is also submodular, then $f(V) \leq f(S \cup V) \leq f(S) + \sum_{x \in V \setminus S} f_S(x)$.

Recall that in this setting, even if we knew the true values of the μ's, the best possible solution would not choose the same arm each time because its rewards would decay eventually, and another arm would be better at that point.

Below in Lemma 1, we show that the utility of the learned allocation $f(S_t)$ is close to that of the optimal allocation.

Lemma 1. *For any $\delta \in (0,1)$ and any $T > 0$, with probability at least $1-\delta$, Algorithm 1 satisfies, for all $t \in [T]$,*

$$f(S^*) \leq f(S_t) + |S^* \setminus S_t| \left(\mu_{j_t} + \sqrt{\frac{\log(2TK/\delta)}{n_{t,j_t}}} \right) \log \left(1 + \frac{1}{n_{t,j_t}} \right) + \sum_{i \in S^* \setminus S_t} \sqrt{\frac{2 \log(2KT/\delta)}{n_{t,i}}} \log \left(1 + \frac{1}{n_i(S_t)} \right),$$

where S_t is the multi-set of the allocation at time t, and j_t is the arm played at time t.

Proof. By Hoeffding's inequality and a union bound over K and T, we have, for any $\delta \in (0,1)$,

$$\mathbb{P} \left(|\mu_i - \widehat{\mu}_{t,i}| \geq \sqrt{\frac{\log(2KT/\delta)}{n_{t,i}}},\ \forall i \in [K], t \in [T] \right) \leq \delta. \tag{7}$$

Consequently, since $f(S_t) = \sum_{i=1}^K \mu_i \log n_{t,i}$ and $\widehat{f}(S_t) = \sum_{i=1}^K \widehat{\mu}_{t,i} \log n_{t,i}$ and $\left| \widehat{f}(S_t) - f(S_t) \right| \leq \sum_{i=1}^K |\mu_i - \widehat{\mu}_{t,i}| \log n_{t,i}$, it holds that with probability exceeding $1-\delta$,

$$\left| f(S_t) - \widehat{f}(S_t) \right| \leq \sum_{i=1}^K \sqrt{\frac{\log(2KT/\delta)}{n_{t,i}}} \log n_{t,i}, \forall t \in [T]. \tag{8}$$

Let j_t be the t-th arm played, and let $S_t = \{j_1, j_2, \cdots, j_t\}$ be the first t arms played. We have, by setting $V = S^*$ and $S = S_t$ in Proposition 3,

$$f(S^*) \leq f(S_t) + \sum_{i \in S^* \setminus S_t} f_{S_t}(i). \tag{9}$$

From the definition of f_{S_t} and (7), with probability at least $1-\delta$, it holds that

$$\begin{aligned} f_{S_t}(i) &= \mu_i \log(n_i(S_t) + 1) - \mu_i \log(n_i(S_t)) \\ &= \mu_i \log \left(1 + \frac{1}{n_i(S_t)} \right) \\ &\leq \left(\widehat{\mu}_{t,i} + \sqrt{\frac{\log(2KT/\delta)}{n_{t,i}}} \right) \log \left(1 + \frac{1}{n_i(S_t)} \right) && \text{(by Eq. 7)} \\ &\leq \widehat{\mu}_{t,i} \log \left(1 + \frac{1}{n_i(S_t)} \right) + \sqrt{\frac{\log(2KT/\delta)}{n_{t,i}}} \log \left(1 + \frac{1}{n_i(S_t)} \right) \\ &= \widehat{f}_{S_t,t}(i) + \sqrt{\frac{\log(2KT/\delta)}{n_{t,i}}} \log \left(1 + \frac{1}{n_i(S_t)} \right). && (10) \end{aligned}$$

Combining (9) and (10), we have, with probability at least $1-\delta$,

$$\begin{aligned} f(S^*) &\leq f(S_t) + \sum_{i \in S^* \setminus S_t} f_{S_t}(i) && \text{(by Eq. 9)} \\ &\leq f(S_t) + \sum_{i \in S^* \setminus S_t} \widehat{f}_{S_t,t}(i) + \sum_{i \in S^* \setminus S_t} \sqrt{\frac{\log(2KT/\delta)}{n_{t,i}}} \log \left(1 + \frac{1}{n_i(S_t)} \right). && (11) \end{aligned}$$

Since $j_t \in \arg\max_{i\in[K]} \left\{ \widehat{f}_t\left(S_t \cup \{i\}\right) - \widehat{f}_t\left(S_t\right) \right\}$, it holds that with probability exceeding $1-\delta$, for any $i \in [K]$ and t,

$$\begin{aligned} \widehat{f}_{S_t,t}\left(i\right) &\le \widehat{f}_t\left(S_t \cup \{j_t\}\right) - \widehat{f}_t\left(S_t\right) \qquad (12) \\ &= \widehat{\mu}_{t,j_t}\left(\log\left(n_{t,j_t}+1\right) - \log(n_{t,j_t})\right) \\ &\le \left(\mu_{j_t} + \sqrt{\frac{\log(2TK/\delta)}{n_{t,j_t}}}\right) \log\left(1+\frac{1}{n_{t,j_t}}\right), \qquad (13) \end{aligned}$$

where (12) uses the greedy nature of the algorithm. We have now bounded each term by an upper bound on its largest value. Plugging (13) into (11) gives that

$$\begin{aligned} f(S^*) &\le f(S_t) + \sum_{i\in S^*\setminus S_t} \left(\mu_{j_t} + \sqrt{\frac{\log(2TK/\delta)}{n_{t,j_t}}}\right) \log\left(1+\frac{1}{n_{t,j_t}}\right) + \sum_{i\in S^*\setminus S_t} \sqrt{\frac{\log\left(2KT/\delta\right)}{n_{t,i}}} \log\left(1+\frac{1}{n_i(S_t)}\right) \\ &\le f(S_t) + |S^*\setminus S_t| \left(\mu_{j_t} + \sqrt{\frac{\log(2TK/\delta)}{n_{t,j_t}}}\right) \log\left(1+\frac{1}{n_{t,j_t}}\right) + \sum_{i\in S^*\setminus S_t} \sqrt{\frac{\log\left(2KT/\delta\right)}{n_{t,i}}} \log\left(1+\frac{1}{n_i(S_t)}\right), \end{aligned}$$

which concludes the proof. □

Next we give a bound on $|S^*\setminus S_T|$ in Proposition 4.

Proposition 4. *With probability exceeding $1-\delta$, it holds that*

$$|S^*\setminus S_T| \le K\sqrt{\frac{(4(K-1)+c)T\log(2TK/\delta)}{c}} + O(1).$$

Proof. When t is large, we have $n_{t,i}^{-1} = O(t^{-1})$ for all i (Proposition 2). Thus the increment ratio $\frac{\widehat{f}_{S_t}(i)}{\widehat{f}_{S_t}(j)}$ can be written out as

$$\frac{\widehat{f}_{S_t}(i)}{\widehat{f}_{S_t}(j)} = \frac{\widehat{\mu}_{t,i}\log\left(1+\frac{1}{n_{t,i}}\right)}{\widehat{\mu}_{t,j}\log\left(1+\frac{1}{n_{t,j}}\right)} = \frac{\widehat{\mu}_{t,i}\left(\frac{1}{n_{t,i}}+O(t^{-2})\right)}{\widehat{\mu}_{t,j}\left(\frac{1}{n_{t,j}}+O(t^{-2})\right)} = \frac{\widehat{\mu}_{t,i}\left(\frac{1}{n_{t,i}}\right)}{\widehat{\mu}_{t,j}\left(\frac{1}{n_{t,j}}\right)} + O(t^{-2}) = \frac{\widehat{\mu}_{t,i}}{\widehat{\mu}_{t,j}}\frac{n_{t,j}}{n_{t,i}} + O(t^{-2}). \qquad (14)$$

Suppose there exists (i,j) such that $\frac{n_{t,i}}{n_{t,j}} \not\approx \frac{\widehat{\mu}_{t,i}}{\widehat{\mu}_{t,j}} + \Omega(t^{-1})$ for large t; i.e., (I) $\frac{n_{t,i}}{n_{t,j}} > \frac{\widehat{\mu}_{t,i}}{\widehat{\mu}_{t,j}} + \Omega(t^{-1})$ or (II) $\frac{n_{t,i}}{n_{t,j}} < \frac{\widehat{\mu}_{t,i}}{\widehat{\mu}_{t,j}} - \Omega(t^{-1})$. Since $\Omega(t^{-1}) > O(t^{-2})$, we have $\widehat{f}_{S_t}(i) < \widehat{f}_{S_t}(j)$ if (I) and $\widehat{f}_{S_t}(i) < \widehat{f}_{S_t}(j)$ if (II). This means j will be played before i if (I) and i will be played before j if (II), which implies that

$$\frac{n_{t,i}}{n_{t,j}} = \frac{\widehat{\mu}_{t,i}}{\widehat{\mu}_{t,j}} + O(t^{-1}).$$

Since $n_{T,i} \ge \frac{cT}{4(K-1)+c}$ for all i, with probability exceeding $1-\delta$, we have

$$\frac{\mu_i + \sqrt{\frac{\log(2TK/\delta)}{\frac{cT}{4(K-1)+c}}}}{\sum_j\left(\mu_j + \sqrt{\frac{\log(2TK/\delta)}{\frac{cT}{4(K-1)+c}}}\right)} \le \frac{\mu_i - \sqrt{\frac{\log(2TK/\delta)}{n_{T,i}}}}{\sum_j\left(\mu_j + \sqrt{\frac{\log(2TK/\delta)}{n_{T,j}}}\right)} \le \frac{\widehat{\mu}_{t,i}}{\sum_j \widehat{\mu}_{t,j}} \le \frac{\mu_i + \sqrt{\frac{\log(2TK/\delta)}{n_{T,i}}}}{\sum_j\left(\mu_j - \sqrt{\frac{\log(2TK/\delta)}{n_{T,j}}}\right)} \le \frac{\mu_i + \sqrt{\frac{\log(2TK/\delta)}{\frac{cT}{4(K-1)+c}}}}{\sum_j\left(\mu_j - \sqrt{\frac{\log(2TK/\delta)}{\frac{cT}{4(K-1)+c}}}\right)$$

By Taylor's theorem, it holds that

$$\frac{\mu_i}{\sum_j \mu_j} - K\sqrt{\frac{\log(2TK/\delta)}{\frac{cT}{4(K-1)+c}}} + O(T^{-1}) \leq \frac{\widehat{\mu}_{t,i}}{\sum_j \widehat{\mu}_{t,j}} \leq \frac{\mu_i}{\sum_j \mu_j} + K\sqrt{\frac{\log(2TK/\delta)}{\frac{cT}{4(K-1)+c}}} + O(T^{-1}). \tag{15}$$

Thus it holds that

$$|S^* \setminus S_T| \leq \sum_j |n_j(S^*) - n_{T,j}| \leq \sum_j \left| \frac{\mu_j}{\sum_i \mu_i} - \frac{\widehat{\mu}_{t,j}}{\sum_i \widehat{\mu}_{t,i}} \right| T + O(1) \leq K\sqrt{\frac{(4(K-1)+c)T\log(2TK/\delta)}{c}} + O(1),$$

where the last inequality uses (15).

□

Next in Theorem 1, we present a theoretical guarantee for Algorithm 1, which shows that the allocation learned at time T is close to the optimal allocation at time T.

Theorem 1. *Assume that the rewards are supported on $[c, 1]$. Let S^* be the multi-set corresponding to the optimal allocation for T pulls. For any $\delta \in (0, 1)$, with probability at least $1 - \delta$, the allocation found by Algorithm 1 satisfies*

$$f(S_T) \geq f(S^*) - O\left(\sqrt{\frac{K^5}{c^3 T}} \log(KT/\delta)\right).$$

Proof of Theorem 1. This Theorem follows from Propositions 2 and 4, and Lemma 1. Combining the above results gives, with probability exceeding $1 - \delta$,

$$\begin{aligned} f(S^*) &\leq f(S_T) + |S^* \setminus S_T| \left(\mu_{j_T} + \sqrt{\frac{\log(2TK/\delta)}{n_{T,j_T}}}\right) \log\left(1 + \frac{1}{n_{T,j_T}}\right) + \sum_{i \in S^* \setminus S_T} \sqrt{\frac{2\log\left(2KT/\delta\right)}{n_{T,i}}} \log\left(1 + \frac{1}{n_i(S_T)}\right) \\ &\overset{(i)}{\leq} f(S_T) + |S^* \setminus S_T| \left(1 + \sqrt{\frac{\log(2TK/\delta)}{n_{T,j_T}}}\right) \frac{1}{n_{T,j_T}} + \sum_{i \in S^* \setminus S_T} \sqrt{\frac{2\log\left(2KT/\delta\right)}{n_{T,i}}} \frac{1}{n_{T,j_T}} \\ &\overset{(ii)}{\leq} f(S_T) + |S^* \setminus S_T| \left(1 + \sqrt{\frac{\log(2TK/\delta)}{\frac{cT}{4(K-1)+c}}}\right) \frac{1}{\frac{cT}{4(K-1)+c}} + |S^* \setminus S_T| \sqrt{\frac{2\log\left(2KT/\delta\right)}{\frac{cT}{4(K-1)+c}}} \frac{1}{\frac{cT}{4(K-1)+c}} \end{aligned}$$

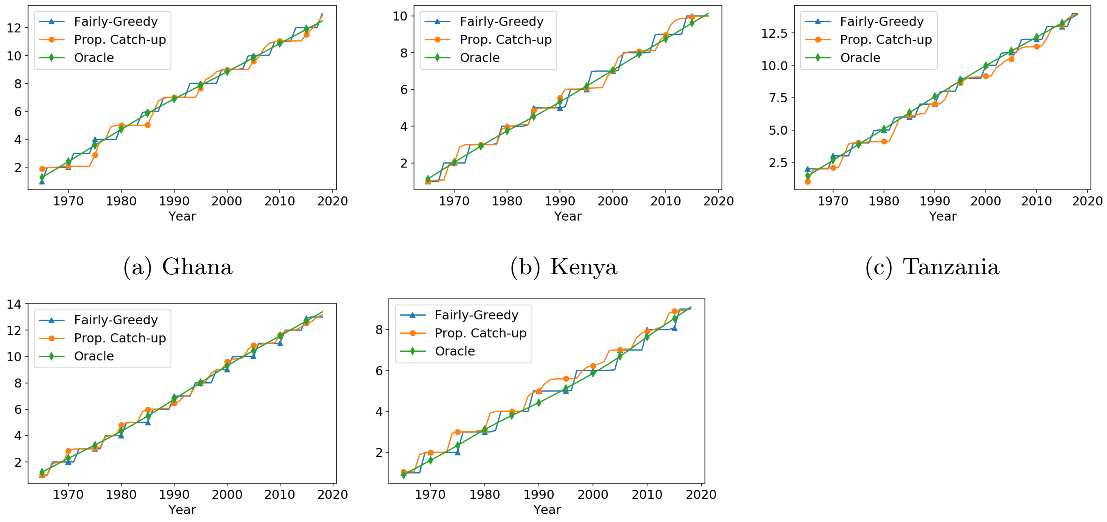
where (i) uses that $\log(1+x) \leq x$ and (ii) uses Proposition 2. Combining the above computations with Proposition 4 gives

$$f(S_T) \geq f(S^*) - O\left(\sqrt{\frac{K^5}{c^3 T}} \log(KT/\delta)\right).$$

□

4.2 Analysis for Proportional Catch-Up

We present a theoretical guarantee for the Proportional Catch-Up algorithm in Theorem 2. The result in Theorem 2 is stronger in the sense that the rewards do not need to be strictly larger than c.



(a) Ghana

(b) Kenya

(c) Tanzania

(d) Zambia

(e) Zimbabwe

Figure 1: Resource allocation learned by different algorithms. Each subfigure plots the resources allocated to one country by the algorithms over the years 1965-2019. The line plot labelled oracle plots the allocation proportional to the average infant mortality rate. The line plots of both Fairly Greedy (Alg1) and Proportional Catch-Up (Alg2) are highly aligned with the oracle line. These plots illustrate how the allocations by Fairly Greedy (Alg1) and Proportional Catch-Up progress over time.

Country	Fairly-Greedy	Prop. Catch-Up	Average Mortality Rate per 10K live newborn
Ghana	12.68	12.94	12.483175
Kenya	10.2	10.19	9.969955
Tanzania	13.98	13.77	13.902460
Zambia	13.18	13.3	13.571910
Zimbabwe	8.96	9.8	9.072500
Cosine Similarity	0.9998	0.9995	

Table 1: The first two columns show the allocations learned by Algorithm 1 (Fairly-Greedy) and Algorithm 2 (Proportional Catch-Up). The last column is the average infants' mortality rate over the years 1966-2019. The bottom row of the table is the cosine similarity between the corresponding allocation and the average infants' mortality rate (the last column). In the experiments, an independent Gaussian noise sampled from $\mathcal{N}(0, 10)$ is added to each observation. Entry table entry (except for the last row and last column) averages over 100 runs. This table shows that Fairly Greedy and Proportional Catch-Up can quickly find an allocation proportional to the average need, which is modelled by the infants' mortality rate in this case.

Theorem 2. *Let δ be an arbitrary number in $(0,1)$, and let t be sufficiently large so that $\frac{t^{3/4}}{K} - \sqrt{2t\log(2Kt/\delta)} \geq 2t^{1/2}\log(2Kt/\delta)$. With probability at least $1 - \frac{\pi^2\delta}{3}$ $(\delta > 0)$, for any $i, j \in [K]$, the allocation learned by Algorithm 2 satisfies, for any $i, j \in [K]$,*

$$\frac{\mu_{t,i} - t^{-1/4}}{\mu_{t,j} + t^{-1/4}} \lesssim \frac{n_{t,i}}{n_{t,j}} \lesssim \frac{\mu_{t,i} + t^{-1/4}}{\mu_{t,j} - t^{-1/4}}, \; i, j \in [K],$$

where the $\lesssim$ sign denotes the $\leq$ relation or the $\approx$ relation.

Proof. Consider the events

$$\mathcal{E}_1 = \left\{ n_{t,i} \geq \frac{t^{3/4}}{K} - \sqrt{2t\log(2Kt/\delta)}, \forall i \in [K], t \in \mathbb{N} \right\},$$

and

$$\mathcal{E}_2 = \left\{ |\widehat{\mu}_{t,i} - \mu_i| \leq \sqrt{\frac{2\log(2Kt/\delta)}{n_{t,i}}}, \forall i \in [K], t \in \mathbb{N} \right\}.$$

By the Azuma-Hoeffding inequality and a union bound, we know that event $\mathcal{E}_1$ and $\mathcal{E}_2$ simultaneously hold with probability exceeding $1 - \frac{\pi^2\delta}{3}$.

We now proceed with the assumption that the high probability event $\mathcal{E}_1 \cap \mathcal{E}_2$ holds true. When t is large enough so that $\frac{t^{3/4}}{K} - \sqrt{2t\log(2Kt/\delta)} \geq 2t^{1/2}\log(2Kt/\delta)$, we have $n_{t,i} \geq 2t^{1/2}\log(2Kt/\delta)$, and thus

$$|\widehat{\mu}_{t,i} - \mu_i| \leq t^{-1/4}. \tag{16}$$

By algorithm design, we have

$$\frac{n_{t,i}}{n_{t,j}} \approx \frac{\widehat{\mu}_{t,i}}{\widehat{\mu}_{t,j}}, \quad \forall i, j \in [K].$$

Thus we have

$$\frac{\mu_{t,i} - t^{-1/4}}{\mu_{t,j} + t^{-1/4}} \lesssim \frac{n_{t,i}}{n_{t,j}} \lesssim \frac{\mu_{t,i} + t^{-1/4}}{\mu_{t,j} - t^{-1/4}}, \; i, j \in [K],$$

where the $\lesssim$ sign denotes $\leq$ or $\approx$.

□

5 Experiments

5.1 Comparison to Other Methods

We compare our methods with the following algorithms: (i) the UCB-prop Singh and Joachims (2018); Wang et al. (2021) algorithm, and (ii) the TS-prop (TS for Thompson Sampling) algorithm (Wang et al., 2021). The UCB-prop (Singh and Joachims, 2018; Wang et al., 2021) algorithm plays a arm j_t at time t according to the following rule

$$j_t \in \arg\max_j \sum_i \left(\widehat{\mu}_{t-1,i} + \sqrt{\frac{\log(4t)}{n_{t-1,i}}} \right) \log \left(1 + \frac{\mathbb{I}_{[j=i]}}{n_{t-1,i}} \right). \tag{17}$$

In other words, it replaces the $\widehat{\mu}_{t,i}$ in the Fairly-Greedy rule (5) with an upper confidence bound. Similarly, the TS-prop algorithm replaces the $\widehat{\mu}_{t,i}$ in (5) with a sample from the posterior distribution for μ_i.

The algorithms (Fairly-Greedy, Proportional Catch-Up, UCB-prop, TS-prop) are all compared with the standard UCB algorithm and the standard Thompson Sampling algorithm for completeness. For both TS-prop and standard Thompson Sampling, the Gaussian model is assumed for each μ_i, and a standard Gaussian prior is used. Via comparing with (i) and (ii), we show that confidence-based exploration is may not be necessary for proportional allocation problems, because arms will be explored anyway based on fairness criteria. Via comparing with (iii) and (iv), the experiments show that, as a sanity check, traditional bandit algorithms cannot solve proportional allocation problems.

The results are summarized in Figure 2.

5.2 Real Data Applications

We apply the algorithm to the World Bank's infant mortality rate dataset (World Bank, 2021), which contains data from 1960 to 2019 for over 200 countries. To use our algorithms on this dataset, we assume that a larger infant mortality rate calls for more medical resources. We select five countries in Africa that are demographically and geographically close. They are Ghana, Kenya, Tanzania, Zambia, and Zimbabwe. The task is to porportionally fairly distribute medical care resources according to infant mortality rate among these countries. The resource distribution protocol proceeds as follows. In each year, we need to distribute one unit of resource to one of the five countries. Once a unit of resource is given to a country, it can never be taken back. In a bandit learning model, we assume that only the feedback from the country that receives resource is revealed. To measure performance, the learned allocation for a country is compared to the country's average mortality rate over years 1961-2019. In the first five years (1961-1965), each country receives one unit of resource as warm-up. The result is summarized in Table 1. As shown in the table, both Fairly-Greedy and Proportional Catch-Up find proportionally fair allocations fairly quickly.

We also visualize the learned allocations and the average need (the moving average of infants' mortality rate) over the years. This result is shown in Figure 1.

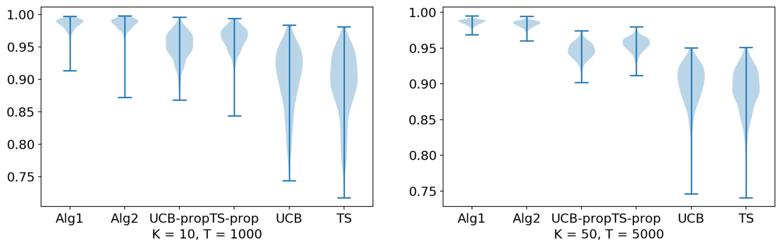


Figure 2: Plots of the cosine similarity between the learned allocations and the true optimal allocation. The left sub-figure shows results for $K = 10$ and $T = 1000$. The right sub-figure shows results for $K = 50$ and $T = 5000$. For each sub-figure, the corresponding parameter setting is repeated 100 times. Each violin plot summarized the distribution of the results from the 100 repetitions, where the horizontal bars enclosing the shaded area are the maximum and minimum values from the runs.

6 Discussion: Continuous-time Stochastic Decision Making

In Proportional Catch-up, each time a unit of resource is allocated to an arm. This decision making process is stochastic, meaning that each time one random arm is selected. The study of the bandit fair allocation algorithms, especially Proportional Catch-up, naturally leads to an important question in stochastic decision making — What if the decision frequency is so high that the a continuous-time model is more appropriate for describing such processes. Naturally, one would expect that the continuous-time process converges to a Wiener process, since Brownian motion is so ubiquitous in the physical world.

The Wiener process can be constructed from symmetric simple random walks. Consider the stochastic process $X_t = \frac{1}{\sqrt{k+1}} \sum_{k=0}^{t} \xi_k$, where ξ_s are *i.i.d.* the Rademacher random variables. Between two times t and s $(t > s)$, the process X_t experiences $(t-s)$ *i.i.d.* normalized increments: $X_t - X_s = \frac{1}{\sqrt{t-s}} \sum_{k=s+1}^{t} \xi_k$. If the process X_t moves more frequently so that there are $m(t-s)$ *i.i.d.* normalized increments between t and s, then we have $X_t - X_s = \frac{1}{m\sqrt{t-s}} \sum_{k=m(s+1)}^{mt} \xi_k$. By the central limit theorem, as m approaches infinity, $X_t - X_s$ converges in distribution to a Gaussian $N(0, t-s)$, for any $t > s \geq 0$. In more modern terms, the Wiener process is governed by the probability law of Wiener measure on continuous functions.

However, the Wiener process, or more precisely multi-dimensional Wiener process, fails to fully capture the stochastic decision making process in bandit learning. The reason is as follows. The decisions in bandit learning are vertices of the $(d-1)$-simplex. Consider the stochastic process $X_t = \frac{1}{\sqrt{m(t+1)}} \sum_{k=0}^{mt} \Delta_k^d$, where Δ_k^d are *i.i.d.* random variables from the vertices of the (centered) simplex. Since the coordinates of Δ_k^d are correlated, $X_t - X_s$ will not converge to a standard multi-dimensional Gaussian distribution as m goes to infinity. A clear understanding of X_t (defined with Δ_k^d) is a profound mathematical problem (Klartag, 2007).

Luckily, we can still study the marginal behavior of X_t. Let $\Delta_{i,j}^d$ be the j-th coordinate of Δ_i^d. The distribution of $\Delta_{i,j}^d$ is

$$\mathbb{P}\left(\Delta_{i,j}^d = 1 - \frac{1}{d}\right) = \frac{1}{d} \quad \text{and} \quad \mathbb{P}\left(\Delta_{i,j}^d = -\frac{1}{d}\right) = \frac{d-1}{d}, \quad \forall i, j.$$

Now, let us fix a coordinate j and define $\xi_t = \frac{d}{\sqrt{m(d-1)(t+1)}} \sum_{k=0}^{mt} \Delta_{k,j}^d$, and let m go to infinity, then the process $\{\xi_t\}_{t \in [0,1]}$ converges to the Wiener process for any d. This is a consequence of the Donsker's invariance principle. Note that when d goes to infinity as well, the above assertion is no longer true. For any k and j, the random variables $\frac{d}{\sqrt{d-1}}\Delta_{k,j}^d$ converge to zero in probability as d goes to infinity. Thus in this case, the limit of ξ_t becomes trivial.

7 Conclusion

In this paper, we study the problem of finding proportionally fair allocations under bandit feedback. We design two new algorithms for this problem, Fairly Greedy and Proportional Catch-Up. We provide theoretical guarantee for these two algorithms and empirically show that our methods outperforms existing methods for proportionally fair allocation problems. The Fairly Greedy and Proportional Catch-Up algorithms are also applied to the infants' mortality dataset by World Bank. The results for this real data experiments show that these two algorithms can quickly find proportionally fair allocations.

References

Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. (2011). Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320.

Agarwal, A., Dudík, M., and Wu, Z. S. (2019). Fair regression: Quantitative definitions and reduction-based algorithms. In *International Conference on Machine Learning*, pages 120–129. PMLR.

Agrawal, S. and Goyal, N. (2012). Analysis of Thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory*, pages 39–1.

Audibert, J.-Y., Munos, R., and Szepesvári, C. (2009). Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902.

Auer, P. (2002). Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422.

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. (1995). Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Proceedings of IEEE 36th Annual Foundations of Computer Science*, pages 322–331. IEEE.

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. (2002). The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77.

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

Bach, F. et al. (2013). Learning with submodular functions: A convex optimization perspective. *Foundations and Trends® in Machine Learning*, 6(2-3):145–373.

Bonald, T., Massoulié, L., Proutiere, A., and Virtamo, J. (2006). A queueing analysis of max-min fairness, proportional fairness and balanced fairness. *Queueing systems*, 53(1):65–84.

Brandt, F., Conitzer, V., Endriss, U., Lang, J., and Procaccia, A. D. (2016). *Handbook of computational social choice*. Cambridge University Press.

Bubeck, S. and Slivkins, A. (2012). The best of both worlds: stochastic and adversarial bandits. In *Conference on Learning Theory*, pages 42–1.

Cesa-Bianchi, N., Freund, Y., Haussler, D., Helmbold, D. P., Schapire, R. E., and Warmuth, M. K. (1997). How to use expert advice. *Journal of the ACM (JACM)*, 44(3):427–485.

Chen, W., Hu, W., Li, F., Li, J., Liu, Y., and Lu, P. (2016). Combinatorial multi-armed bandit with general reward functions. In *Advances in Neural Information Processing Systems*, pages 1651–1659.

Chen, W., Wang, Y., and Yuan, Y. (2013). Combinatorial multi-armed bandit: General framework and applications. In *International Conference on Machine Learning*, pages 151–159. PMLR.

Chen, X., Fain, B., Lyu, L., and Munagala, K. (2019). Proportionally fair clustering. In *International Conference on Machine Learning*, pages 1032–1041. PMLR.

Garivier, A. and Cappé, O. (2011). The KL–UCB algorithm for bounded stochastic bandits and beyond. In *Conference on Learning Theory*, pages 359–376.

Joseph, M., Kearns, M., Morgenstern, J. H., and Roth, A. (2016). Fairness in learning: Classic and contextual bandits. *Advances in neural information processing systems*, 29.

Kearns, M. and Roth, A. (2019). *The Ethical Algorithm: The Science of Socially Aware Algorithm Design.* Oxford University Press, USA.

Kelly, F. P., Maulloo, A. K., and Tan, D. K. H. (1998). Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research society*, 49(3):237–252.

Klartag, B. (2007). A central limit theorem for convex sets. *Inventiones mathematicae*, 168:91–131.

Krause, A. and Ong, C. S. (2011). Contextual Gaussian process bandit optimization. In *Advances in Neural Information Processing Systems*, pages 2447–2455.

Krishnaswamy, A., Jiang, Z., Wang, K., Cheng, Y., and Munagala, K. (2021). Fair for all: Best-effort fairness guarantees for classification. In *International Conference on Artificial Intelligence and Statistics*, pages 3259–3267. PMLR.

Kushner, H. J. and Whiting, P. A. (2004). Convergence of proportional-fair sharing algorithms under general conditions. *IEEE Transactions on Wireless Communications*, 3(4):1250–1259.

Lai, T. L. and Robbins, H. (1985). Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22.

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

Li, L., Chu, W., Langford, J., and Schapire, R. E. (2010). A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, pages 661–670.

Littlestone, N. and Warmuth, M. K. (1994). The weighted majority algorithm. *Information and computation*, 108(2):212–261.

Maillard, O.-A., Munos, R., and Stoltz, G. (2011). A finite-time analysis of multi-armed bandits problems with Kullback-Leibler divergences. In *Conference On Learning Theory*, pages 497–514.

Patil, V., Ghalme, G., Nair, V., and Narahari, Y. (2020). Achieving fairness in the stochastic multi-armed bandit problem. In *AAAI*, pages 5379–5386.

Robbins, H. (1952). Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535.

Seldin, Y. and Slivkins, A. (2014). One practical algorithm for both stochastic and adversarial bandits. In *International Conference on Machine Learning*, pages 1287–1295.

Shalev-Shwartz, S. et al. (2011). Online learning and online convex optimization. *Foundations and trends in Machine Learning*, 4(2):107–194.

Singh, A. and Joachims, T. (2018). Fairness of exposure in rankings. KDD '18, page 2219–2228, New York, NY, USA. Association for Computing Machinery.

Slivkins, A. (2014). Contextual bandits with similarity information. *The Journal of Machine Learning Research*, 15(1):2533–2568.

Srinivas, N., Krause, A., Kakade, S., and Seeger, M. (2010). Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning*.

Talebi, M. S. and Proutiere, A. (2018). Learning proportionally fair allocations with low regret. *Proc. ACM Meas. Anal. Comput. Syst.*, 2(2).

Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294.

Tsamados, A., Aggarwal, N., Cowls, J., Morley, J., Roberts, H., Taddeo, M., and Floridi, L. (2021). The ethics of algorithms: key problems and solutions. *AI & Society*, pages 1–16.

Wang, L., Bai, Y., Sun, W., and Joachims, T. (2021). Fairness of exposure in stochastic bandits. In *International Conference on Machine Learning*, pages 10686–10696. PMLR.

World Bank (2021). Infants' mortality rate dataset by the World Bank.

Zimmert, J., Luo, H., and Wei, C.-Y. (2019). Beating stochastic and adversarial semi-bandits optimally and simultaneously. In *International Conference on Machine Learning*, pages 7683–7692. PMLR.

Zimmert, J. and Seldin, Y. (2021). Tsallis-INF: An optimal algorithm for stochastic and adversarial bandits. *J. Mach. Learn. Res.*, 22:28–1.