Dynamic pricing, also known as surge pricing, demand pricing, or time-based pricing, is a pricing strategy where businesses adjust the prices of their offerings to account for A change in demand. And there's basically an art to how the price offered to customers is calculated to match what customers are actually willing to pay for what they're selling at that moment. industries to list a few-- hospitality, transportation, airlines and ridesharing, gas and electric companies, retail and e-commerce, event ticketing, and real estate. Some of the core factors that play into how price is determined are supply and demand, value, market trends, seasonality and time of year, competitor pricing.

Some pros of dynamic pricing are that, first, employees can be paid a higher wage during busier times. Second, dynamic pricing allows you to still sell in downtimes. The major con is that with dynamic pricing you run the risk that customers might feel cheated and trust you less.

Doubly Fair Dynamic Pricing

procedural equality or equality of application or formal equality means that everyone is subject to the same rules and standards justice should be blind . Equality of outcome or substantive equality does not mean treating people the same but requires treating them differently so as to achieve equal or equivalent results

Jianyu Xu **Dan Qiao** **Yu-Xiang Wang**
Computer Science Department,
University of California, Santa Barbara

Abstract

We study the problem of online dynamic pricing with two types of fairness constraints: a *procedural fairness* which requires the *proposed* prices to be equal in expectation among different groups, and a *substantive fairness* which requires the *accepted* prices to be equal in expectation among different groups. A policy that is simultaneously procedural and substantive fair is referred to as *doubly fair*. We show that a doubly fair policy must be random to have higher revenue than the best trivial policy that assigns the same price to different groups. In a two-group setting, we propose an online learning algorithm for the 2-group pricing problems that achieves $\tilde{O}(\sqrt{T})$ regret, zero procedural unfairness and $\tilde{O}(\sqrt{T})$ substantive unfairness over T rounds of learning. We also prove two lower bounds showing that these results on regret and unfairness are both information-theoretically optimal up to iterated logarithmic factors. To the best of our knowledge, this is the first dynamic pricing algorithm that learns to price while satisfying two fairness constraints at the same time.

1 Introduction

Pricing problems have been studied since Cournot (1897). In a classical pricing problem setting such as Kleinberg and Leighton (2003); Broder and Rusmevichientong (2012); Besbes and Zeevi (2015), the seller (referred as "we") sells identical products in the following scheme.

Online pricing. For $t = 1, 2, \ldots, T$:
1. The customer valuates the product as y_t.
2. The seller proposes a price v_t concurrently without knowing y_t.
3. The customer makes a decision $\mathbf{1}_t = \mathbf{1}(v_t \le y_t)$.
4. The seller receives a reward (revenue) $r_t = v_t \cdot \mathbf{1}_t$.

Here T is the time horizon known to the seller in advance[1], and y_t's are drawn from a fixed distribution independently. The goal is to approach an optimal price that maximizes the expected revenue-price function. In order to make this, we should learn gradually from the binary feedback and improve our knowledge on customers' valuation distribution (or so-called "demands" (Kleinberg and Leighton, 2003)).

In recent years, with the development of price discrimination and personalized pricing strategies, *fairness* issues on pricing arose social and academic concerns (Kaufmann et al., 1991; Chapuis, 2012; Richards et al., 2016; Eyster et al., 2021). Customers are usually not satisfied with price discrimination, which can lead to reduced willingness to purchase and a damaged reputation for the seller. In the online pricing problem defined above, when we are selling identical items to customers from different *groups* (e.g., divided by gender, race, age, etc.), it can be unfair to offer different optimal price to each group: Optimal prices in different groups are not necessarily the same, and unfairness occurs if different customers are provided or buying the same item with different prices. Inspired by the concept of *procedural and substantive unconscionability* (Elfin, 1988), we define a *procedural unfairness* measuring the difference of *proposed prices* between the two groups, and a *substantive unfairness* measuring the difference of *accepted prices* between the two groups. Given these notions, our goal is to approach the optimal pricing policy that maximizes the expected total revenue with no procedural and substantive unfairness.

The concept of procedural fairness has been well established in Cohen et al. (2022) as "price fairness", while the concept of the substantive fairness is new to this paper. In fact, both procedural and substantive fairness have significant impacts on customers' experience and social justice. For instance, these notions help prevent the following two scenarios:

- Perspective buyers who are women found that they are offered consistently higher average price than men for the same product.

[1] Here we assume T known for simplicity of notations. In fact, if T is unknown, then we may apply a "doubling epoch" trick as Javanmard and Nazerzadeh (2019) and the regret bounds are the same.

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206.

- Women who have bought the product found that they paid a higher average price than men who have bought the product.

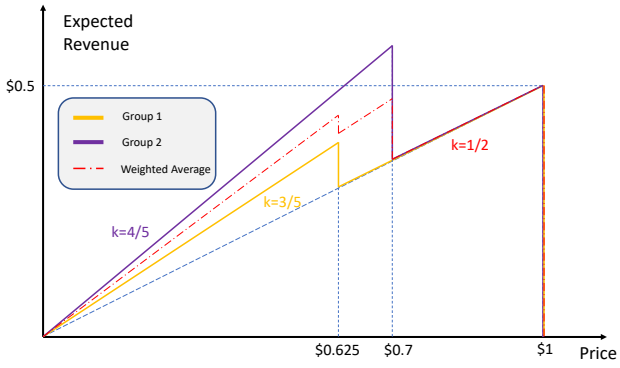
Therefore, a good pricing strategy has to satisfy both procedural and substantive fairness.

However, these constraints are very hard to satisfy even with full knowledge on customers' demands. If we want to fulfill those two sorts of fairness perfectly by proposing deterministic prices for different groups, the only thing we can do is to trivially set the same price in all groups and to maximize the weighted average revenue function by adjusting this uniformly fixed price with existing methods such as Kleinberg and Leighton (2003). Consider the following example:

Example 1. Customers form two disjoint groups, where 30% customers are in Group 1 and the rest 70% are in Group 2. For each price in $\{\$0.625, \$0.7, \$1\}$, customers in two groups have different acceptance rates:

Acceptance Rate	\$0.625	\$0.7	\$1
G_1 (30%)	3/5	1/2	1/2
G_2 (70%)	4/5	4/5	1/2

The figure below shows the expected revenue functions of prices in each group, where the red dashed line is their weighted average by population.



In Example 1, the only way to guarantee both fairness constraints is to propose the same price for both groups, and the optimal price is *\$1* whose expected revenue is $\$0.5$ as is shown in the figure.

However, if we instead propose a *random price distribution* to each group and inspect those fairness notions *in expectation*, then there may exist price distributions that satisfy both of the fairness constraints and achieve higher expected revenue than any fixed-price strategy. Here a price distribution is the distribution over the prices for customers, and the exact price for each customer is *sampled* from this distribution *independently*. This random price sampling process can be implemented by marketing campaigns or promotions such as random discounts or randomly-distributed coupons. Again, we consider Example 1 and the following random policy:

- For customers from G_1, propose *\$0.625* with probability $\frac{20}{29}$ and *\$1* with probability $\frac{9}{29}$.
- For customers from G_2, propose *\$0.7* with probability $\frac{25}{29}$ and *\$1* with probability $\frac{4}{29}$.

Under this policy, the expected proposed price and the expected accepted price in both groups are $\$\frac{43}{58}$ and $\$\frac{8}{11}$ respectively. Furthermore, the expected revenue is $\$\frac{74}{145} > \$0.5$, which means that this random policy performs better than the best fixed-price policy. It is worth mentioning that this is exactly the optimal doubly-fair random policy in this specific setting, but the proof of its optimality is highly non-trivial (and we put it in Appendix B.3 as part of the proof of Theorem 9).

In this work, we consider a two-group setting and we denote a *policy* as the tuple of two price distributions over the two groups respectively. Therefore, we can formally define the optimal policy as follows:

$$\begin{aligned}\pi_* = \underset{\pi=(\pi^1,\pi^2)}{\operatorname{argmax}}\ & q\cdot \mathop{\mathbb{E}}_{v_t^1\sim\pi^1, y_t^1\sim\mathbb{D}^1}[v_t^1\cdot \mathbb{1}(v_t^1\le y_t^1)] \\ & + (1-q)\cdot \mathop{\mathbb{E}}_{v_t^2\sim\pi^2, y_t^2\sim\mathbb{D}^2}[v_t^2\cdot \mathbb{1}(v_t^2\le y_t^2)] \\ s.t.\quad & \mathbb{E}_{\pi^1}[v_t^1] = \mathbb{E}_{\pi^2}[v_t^2] \\ & \mathbb{E}_{\pi^1,\mathbb{D}^1}[v_t^1|\mathbb{1}(v_t^1\le y_t^1)=1] \\ & = \mathbb{E}_{\pi^2,\mathbb{D}^2}[v_t^2|\mathbb{1}(v_t^2\le y_t^2)=1]\end{aligned} \tag{1}$$

Here $\pi^1, v_t^1, y_t^1, \mathbb{D}^1$ and $\pi^2, v_t^2, y_t^2, \mathbb{D}^2$ are the proposed price distributions, proposed prices, customer's valuations and valuation distributions of Group 1 and Group 2 respectively, and q is the share (proportion) that Group 1 takes. From (1), the optimal policy under the in-expectation fairness constraints should be random in general[2]. However, even we know the exact $\mathbb{D}^1$ and $\mathbb{D}^2$, it is still a very hard problem to get π_*: Both sides of the second constraint in (1) are conditional expectations (i.e., fractions of expected revenue over expected acceptance rate) and is thus not convex (and also not quasiconvex). To make it more realistic (and also harder), the seller actually has no direct access to customers' demands $\mathbb{D}^1$ and $\mathbb{D}^2$ at the beginning. Therefore, in this work we consider a T-round *online* learning and pricing setting, where we could learn these demands from those *Boolean-censored* feedback (i.e., customers' decisions) and improve our pricing policy to approach π_* in (1).

In order to measure the performance of a specific policy, we define a *regret* metric that equals the expected revenue dif-

[2]Notice that a fixed-price policy can also be considered as "random".

ference between this policy and the optimal policy. We also quantify the procedural and substantive unfairness that are equal to the absolute difference of expected proposed/accepted prices in two groups. We will establish a more detailed problem setting in Section 3.

Summary of Results Our contributions are threefold:

- We design an algorithm, FPA, that achieves an $O(\sqrt{T}d^{\frac{3}{2}}\log\frac{d\log T}{\epsilon})$ cumulative regret with 0 procedural unfairness and $O(\sqrt{T}d^{\frac{3}{2}}\log\frac{d\log T}{\epsilon})$ substantive unfairness, with probability at least $(1-\epsilon)$. Here d is the total number of prices allowed to be chosen from. These results indicate that our FPA is asymptotically no-regret and fair as T gets large.
- We show that the regret of FPA is optimal with respect to T, as it matches $\Omega(\sqrt{T})$ regret lower bound up to $\log\log T$ factors.
- We show that the unfairness of FPA is also optimal with respect to T up to $\log\log T$ factors, as it has no procedural unfairness and its substantive unfairness matches the $\Omega(\sqrt{T})$ lower bound for any algorithm achieving an optimal $O(\sqrt{T})$ regret.

To the best of our knowledge, we are the first to study a pricing problem with multiple fairness constraints, where the optimal pricing policy is necessary to be random. We also develop an algorithm that is able to approach the best random pricing policy with high probability and at the least cost of both revenue and fairness.

Technical Novelty. Our algorithm is a "conservative policy-elimination"-based strategy that runs in epochs with doubling batch sizes as in Auer et al. (2002a). We cannot directly apply the action-elimination algorithm for multi-armed bandits as in Cesa-Bianchi et al. (2013), because the policy space is an infinite set and we cannot afford to try each one out. The fairness constraints further complicate things. Our solution is to work out just a few representative policies that are "good-and-exploratory", which can be used to evaluate the revenue and fairness of all other policies, then eliminate those that are unfair or have suboptimal revenue. Since we do not have direct access to the demand function, the estimated fairness constraints are changing over epochs due to estimation error. Therefore, it is non-trivial to keep the target optimal policy inside our "good policy set" during iterations. We settle this issue by setting the criteria of a "good policy" conservatively.

Our lower bound is new too and it involves techniques that could be of independent interest to the machine learning theory community. Notice that it is possible to have a perfectly fair algorithm by trivially proposing the same fixed price for both groups. It is highly non-trivial to show the unfairness lower bound within the family of regret-optimal algorithms. We present our result in Section 5.3 by establishing two similar problem settings that any algorithm cannot distinguish them efficiently and showing that a mismatch would cause a compatible amount of regret and substantive unfairness.

2 Related Works

Here we discuss some literature closely related to this work. Please refer to Appendix A for a broader discussion.

Dynamic Pricing Single product dynamic pricing problem has been well-studied through Kleinberg and Leighton (2003); Besbes and Zeevi (2009); Wang et al. (2014); Chen et al. (2019); Wang et al. (2021). The crux is to learn and approach the optimal of a revenue curve from Boolean-censored feedback. In specific, Kleinberg and Leighton (2003) proves $\Theta(\log\log T)$, $\Theta(\sqrt{T})$ and $\Theta(T^{\frac{2}{3}})$ minimax regret bounds under noise-free, infinitely smooth and stochastic/adversarial valuation assumptions, sequentially. Wang et al. (2021) further shows a $\Theta(T^{\frac{K+1}{2K+1}})$ minimax regret bound for K^{th}-smooth revenue functions. In all these works, the decision space is continuous. In our problem setting, we require the prices to be chosen from a fixed set of d prices, and show a bandit-style $\Omega(\sqrt{dT})$ regret lower bound similar to Auer et al. (2002b).

Fairness in Machine Learning Fairness is a long-existing topic that has been extensively studied. In the machine learning community, fairness is defined from mainly two perspectives: the *group fairness* and the *individual fairness*. In a classification problem, for instance, (Dwork et al., 2012) defines these two notions as follows: (1) A group fairness requires different groups to have identical result distributions in statistics, which further includes the concepts of "demographic parity" (predictions independent to group attributes) and "equalized odds" (predictions independent to group attributes *conditioning on* the true labels). In Agarwal et al. (2018), these group fairness are reduced to linear constraints. The two fairness definitions we make in this work, the *procedural fairness* and the *substantive fairness*, belong to group fairness. (2) An individual fairness (Hardt et al., 2016) requires the difference of predictions on two individuals to be upper bounded by a distance metric of their intrinsic features. The notion "time fairness" is often considered as individual fairness as well. We provide a more detailed discussion on the line of work that address fairness concerns or stochastic constraints with online learning techniques in Appendix A.

Fairness in Pricing Recently there are many works contributing to pricing fairness problems (Kaufmann et al., 1991; Frey and Pommerehne, 1993; Chapuis, 2012; Richards et al., 2016; Priester et al., 2020; Eyster et al., 2021; Yang et al., 2022). As is stated in Cohen et al. (2022), in a pricing problem with fairness concerns, the concept of fairness in existing works is modeled either as a utility or

budget that trades-off the revenue or as a hard constraint that prevent us from taking the best action directly. Cohen et al. (2022) chooses the second model and defines four different types of fairness in pricing: price fairness, demand fairness, surplus fairness and no-purchase valuation fairness, each of which indicates the difference of prices, the acceptance rate, the surplus (i.e., (valuation − price) if bought and 0 otherwise) and the average valuation of not-purchasing customers in two groups is bounded, sequentially. They show that it is impossible to achieve any pair of different fairness notions simultaneously (with deterministic prices). In fact, this can be satisfied if they allow random pricing policies. Maestre et al. (2018) indeed builds their fairness definition upon random prices by introducing a "Jain's Index", which indicates the homogeneity of price distributions among different groups (i.e., our procedural fairness notion). They develop a reinforcement-learning-based algorithm to provide homogeneous prices, with no theoretic guarantees.

Cohen et al. (2021) and Chen et al. (2021) study the online-learning-fashion pricing problem as we do. Cohen et al. (2021) considers both group (price) fairness and individual (time) fairness, and their algorithm FaPU solves this problem with sublinear regret while guaranteeing fairness. They further study the pricing problem with demand fairness that are unknown and needs learning. In this setting, they propose another FaPD algorithm that achieves the optimal $\tilde{O}(\sqrt{T})$ regret and guarantees the demand fairness "almost surely", i.e., upper bounded by $\delta \cdot T$ as a budget. Chen et al. (2021) considers two different sorts of fairness constraints: (1) Price fairness constraints (as in Cohen et al. (2022)) are enforced; (2)Price fairness constraints are generally defined (and maybe not accessible), where they adopt "soft fairness constraints" by adding the fairness violation to the regret with certain weights. In both cases, they achieve $\tilde{O}(T^{\frac{4}{5}})$ regrets. These learning-based fairness requirements are quite similar to our problem setting, but in our setting the fairness constraints are non-convex (while theirs are linear) and are also optimized to corresponding information-theoretic lower bounds without undermining the optimal regret.

3 Problem Setup

In this section, we describe the problem setting of online pricing, introduce new fairness definitions and set the goal of our algorithm design.

Problem Description. We start with the online pricing process. The whole selling session involves customers from two groups (G_1 and G_2) and lasts for T rounds. Prices are only allowed to be chosen from a *known* and *fixed* set of d prices: $\mathbf{V} = \{v_1, v_2, \dots, v_d\}$, where $0 < v_1 < v_2 < \dots < v_d \leq 1$. Denote $\Delta^d = \{x \in \mathbb{R}_+^d, \|x\|_1 = 1\}$ as the probabilistic simplex. At each time $t = 1, 2, \dots, T$, we propose a pricing policy $\pi = (\pi^1, \pi^2)$ consisting of two probabilistic distributions $\pi^1, \pi^2 \in \Delta^d$ over all d prices. A customer then arrives with an observable group attribution G_e ($e \in \{1, 2\}$), and we propose a price by sampling a v_t^e from $\mathbf{V}$ according to distribution π^e. At the same time, the customer generates a valuation y_t^e *in secret*, where y_t^e is sampled independently and identically from some fixed unknown distribution $\mathbb{D}_e$. Afterward, we observe a feedback $\mathbf{1}_t^e = \mathbf{1}(v_t^e \leq y_t^e)$ and receive a reward(revenue) $r_t^e = \mathbf{1}_t^e \cdot v_t^e$.

Key Quantities. Here we define a few quantities and functions that is necessary to formulate the problem. Denote $\mathbf{v} := [v_1, v_2, \dots, v_d]^\top$, $[d] := \{1, 2, \dots, d\}$ and $\mathbb{1} := [1, 1, \dots, 1]^\top \in \mathbb{R}^d$ for simplicity. Denote $F_e(i) := \Pr_{\mathbb{D}_e}[y_t^e \geq v_i], e = 1, 2, i \in [d]$ as the probability of price v_i being accepted in G_e. Since $v_i < v_j$ for $i < j$, we know that $F_e(1) \geq F_e(2) \geq \dots \geq F_e(d)$. Notice that all $F_e(i)$'s are *unknown* to us. Define a matrix $F_e := \mathrm{diag}(F_e(1), F_e(2), \dots, F_e(d))$.

As a result, for a customer from G_e ($e = 1, 2$), we know that

- The expected proposed price is $\mathbf{v}^\top \pi^e$.
- The expected reward(revenue) is $\mathbf{v}^\top F_e \pi^e$.
- The expected acceptance rate is $\mathbb{1}^\top F_e \pi^e$.
- The expected accepted price is $\frac{\mathbf{v}^\top F_e \pi^e}{\mathbb{1}^\top F_e \pi^e}$.

Denote the proportion of G_1 in all potential customers as q ($0 < q < 1$) which is fixed and *known* to us, and we assume that every customer is chosen from all potential customers uniformly at random. As a consequence, we can define the expected revenue of a policy π.

Definition 2 (Expected Revenue)**.** *For any pricing policy $\pi = (\pi^1, \pi^2) \in \Pi$, define its expected revenue (given F_1 and F_2) as the weighted average of the expected rewards of G_1 and G_2.*

$$\begin{aligned} R(\pi; F_1, F_2) :=& \Pr[\textit{Customer is from } G_1] \cdot \mathbb{E}[r_t^1] \\ &+ \Pr[\textit{Customer is from } G_2] \cdot \mathbb{E}[r_t^2] \\ =& q \cdot \mathbf{v}^\top F_1 \pi^1 + (1-q) \cdot \mathbf{v}^\top F_2 \pi^2 \end{aligned} \tag{2}$$

Also, we can define the two different unfairness notions based on these results above.

Definition 3 (Procedural Unfairness)**.** *For any pricing policy $\pi \in \Pi$, define its procedural unfairness as the absolute difference between the expected* proposed *prices of two groups.*

$$U(\pi) := |\mathbf{v}^\top \pi^1 - \mathbf{v}^\top \pi^2| = |\mathbf{v}^\top (\pi^1 - \pi^2)|. \tag{3}$$

Procedural unfairness is totally tractable as we have full access to $\mathbf{v}^\top$ and π. Therefore, we can define a policy family $\Pi := \{\pi = (\pi^1, \pi^2), U(\pi) = 0\}$ that contains all policies with no procedural unfairness. Now we define a substantive unfairness as another metric.

subset of the unit simplex in which the sum of the elements of the vector is exactly one, i.e., n i=1xi = 1 i = 1 n x i = 1. In two dimensions, the unit simplex is the triangle formed by coordinates (0,0), (0,1) and (1,0), whereas the probability simplex is the line joining (1,0) and (0,1).

Definition 4 (Substantive Unfairness). *For any pricing policy $\pi \in \Pi$, define its substantive unfairness as the difference between the expected* accepted *prices of two groups.*

$$\begin{aligned} S(\pi; F_1, F_2) :=& \Big| \mathbb{E}[v^1 | v^1 \sim \pi^1, v^1 \text{ being accepted }] \\ & - \mathbb{E}[v^2 | v^2 \sim \pi^2, v^2 \text{ being accepted }] \Big| \\ =& \left| \frac{\mathbf{v}^\top F_1 \pi^1}{\mathbb{1}^\top F_1 \pi^1} - \frac{\mathbf{v}^\top F_2 \pi^2}{\mathbb{1}^\top F_2 \pi^2} \right|. \end{aligned} \tag{4}$$

Substantive unfairness is not as tractable as procedural unfairness, as we have no direct access to the true F_1 and F_2. Ideally, the optimal policy that we want to achieve is:

$$\begin{aligned} \pi_* = & \operatorname*{argmax}_{\pi=(\pi^1,\pi^2)\in\Pi} R(\pi; F_1, F_2) \\ s.t. \quad & U(\pi) = 0, \quad S(\pi; F_1, F_2) = 0. \end{aligned} \tag{5}$$

The feasibility of this problem is trivial: policies such as $\pi^1 = \pi^2 = [0, \dots, 0, 1, 0, \dots, 0]^\top$ (i.e., proposing the same fixed price despite the customer's group attribution) are always feasible. However, this problem is in general highly non-convex and non-quasi-convex. Finally, we define a *(cumulative) regret* that measure the performance of any policy π:

Definition 5 (Regret). *For any algorithm $\mathcal{A}$, define its cumulative regret as follows:*

$$\begin{aligned} Reg_T(\mathcal{A}) :=& \sum_{t=1}^{T} Reg(\pi_t; F_1, F_2) \\ :=& \sum_{t=1}^{T} R(\pi_*; F_1, F_2) - R(\pi_t; F_1, F_2). \end{aligned} \tag{6}$$

Here π_t is the policy proposed by $\mathcal{A}$ at time t.

Notice that we define the per-round regret by comparing the performance of π_t with the optimal policy π_* under constraints. Therefore, $Reg(\pi_t; F_1, F_2)$ is possible to be negative if $\pi_t \in \Pi$ but $S(\pi_t; F_1, F_2) > 0$. Similarly, we define a *cumulative substantive unfairness* as $S_T(\mathcal{A}) := \sum_{t=1}^{T} S(\pi_t; F_1, F_2)$.

Goal of Algorithm Design Our ultimate goal is to approach π_* in the performance. In the online pricing problem setting we adopt, however, we cannot guarantee $S(\pi_t; F_1, F_2) = 0$ for all π_t we propose at time $t = 1, 2, \dots, T$ since we do not know F_1 and F_2 in advance. Instead, we may suffer a gradually vanishing unfairness as we learn F_1 and F_2 better. Therefore, our goal in this work is to design an algorithm that guarantees an optimal regret while suffering 0 cumulative procedural unfairness and the least cumulative substantive unfairness.

Technical Assumptions. Here we make some mild assumptions that help our analysis.

Assumption 1 (Least Probability of Acceptance). *There exists a fixed constant $F_{\min} > 0$ such that $F_e(d) \geq F_{\min}, e = 1, 2$.*

Assumption 1 not only ensures the definition of expected accepted price to be sound (by ruling out these unacceptable prices), but also implies $S(\pi, F_1, F_2)$ to be Lipschitz. Besides, we can always achieve this by reducing v_d. We will provide a detailed discussion in Section 6.

Assumption 2 (Number of Possible Prices). *We treat d, the number of prices, as an amount independent from T. Also, we assume $d = O(T^{\frac{1}{3}})$.*

Assumption 2 is a necessary condition of applying $\Omega(\sqrt{dT})$ regret lower bound, and we will explain more in Appendix B.2. As we are more curious about how the fairness constraints affect the interactive pricing process over time, this assumption separates the dependence on d and helps to show the optimality of our algorithm w.r.t. T.

4 Algorithm

In this section, we propose our Fairly Pricing Algorithm (FPA) in Algorithm 1 and then discuss the techniques we develop and apply to achieve the "no-regret" and "no-unfairness" goal.

4.1 Algorithm Components

Algorithm 1 takes the following inputs: time horizon T, price set $\mathbf{V}$, error probability ϵ, a universal constant L as the coefficient of the performance-fairness tradeoff on constraint relaxations (see Lemma 14), and q as the proportion that G_1 takes. We also adopt the following techniques and components that contribute to its no-regret and no-unfairness performance.

4.1.1 Before Epochs.

In this stage, we keep proposing the highest price v_d for $\tau_0 = O(\log T)$ rounds to estimate(lower-bound) the least accepting probability $F_{\min}$.

Before Epochs:

Initialize counter $M_{0,e} = 0$ and $N_{0,e} = 0$ for $e = 1, 2$.
for $t = 1, 2, \dots, \tau_0 = 2 \log T \log \frac{16}{\epsilon}$ **do**
  Denote the customer's group index as $e_t \in \{1, 2\}$.
  Set $M_{0,e_t} += 1$.
  Propose the highest price v_d.
  If accepted, set $N_{0,e_t} += 1$.
end for
Output $\hat{F}_{\min} = \min\{\frac{N_{0,1}}{2M_{0,1}}, \frac{N_{0,2}}{2M_{0,2}}\}$.

4.1.2 Doubling Epochs.

Despite the "before epochs" stage, we divide the whole time space into epochs $k = 1, 2, \dots$, where each epoch k has a length $\tau_k = O(\sqrt{T} \cdot 2^k)$ that doubles the length of epoch $(k-1)$. Intuitively, a longer epoch can improve the estimates and help the algorithm select better policies, which would in

Algorithm 1 Fairly Pricing Algorithm (FPA)

Input: Time horizon T, prices set $\mathbf{V}$, error probability ϵ, universal constant L, proportion q.
Before Epochs: Keep proposing the highest price v_d and estimate the lowest accepting probability as $\hat{F}_{\min}$. (See Section 4.1.1.)
Initialization: Candidate policy set $\Pi_1 = \Pi := \{\pi = (\pi^1, \pi^2), U(\pi) = 0\}$ and price index set $I_0^1 = I_0^2 = [d]$.
for Epoch $k = 1, 2, \ldots$ **do**
  Set epoch length τ_k, reward uncertainty $\delta_{k,r}$ and unfairness uncertainty $\delta_{k,s}$.
  Select good-and-exploratory policies from Π_k and form a set A_k. (See Section 4.1.3.)
  Estimate acceptance probabilities $F_e(i)$ as $\tilde{F}_{k,e}(i)$ for $e = 1, 2$ and $i = 1, 2, \ldots, d$. (See Section 4.1.4.)
  Let $\hat{F}_{k,e} = \mathrm{diag}(\bar{F}_{k,e}(1), \bar{F}_{k,e}(2), \ldots, \bar{F}_{k,e}(d)), e = 1, 2$.
  Get empirical optimal policy $\hat{\pi}_{k,*}$: Solve (7) with Algorithm 2

$$\hat{\pi}_{k,*} = \operatorname*{argmax}_{\pi \in \Pi_k} R(\pi, \hat{F}_{k,1}, \hat{F}_{k,2}), \text{ s.t. } S(\pi, \hat{F}_{k,1}, \hat{F}_{k,2}) \le \delta_{k,s}. \tag{7}$$

  Update the policy set Π_k: Solve (8) with Algorithm 3

$$\Pi_{k+1} = \{\pi : \pi \in \Pi_k, S(\pi, \hat{F}_{k,1}, \hat{F}_{k,2}) \le \delta_{k,s}, R(\pi, \hat{F}_{k,1}, \hat{F}_{k,2}) \ge R(\hat{\pi}_{k,*}, \hat{F}_{k,1}, \hat{F}_{k,2}) - \delta_{k,r} - L \cdot \delta_{k,s}\}. \tag{8}$$

end for

return reduce the regret in the next epoch with even longer time horizon.

4.1.3 Good-and-Exploratory Policies.

A profitable pricing policy might not be suitable of running in consideration of exploration, which is important for estimating $F_1(i)$ and $F_2(i)$ that facilitates the policy elimination. We resolve this issue by keeping a set of *good-and-exploratory* policies: After eliminating sub-optimal policies at the end of previous epoch, for each price v_i in group G_e we find out a policy in the remaining policies that maximizes the probability of proposing v_i in G_e at the beginning of current epoch. The larger this probability is, the more times v_i can be chosen in G_e, which would lead to a better estimate of $F_e(i)$. Here we give up to estimate the acceptance probability of those v_i with $\le \frac{1}{\sqrt{T}}$ to be chosen by the optimal policy π_*, as it would not affect the elimination process and the performance substantially.

Select good-and-exploratory policies:

Initialize $A_k = \emptyset$, $I_k^1 = I_{k-1}^1$ and $I_k^2 = I_{k-1}^2$.
for Group $e = 1, 2$ and for price index $i \in I_{k-1}^e$, **do**
  {Pick up policy maximizing each probability:}
  Get $\tilde{\pi}_{k,i,e} = \operatorname{argmax}_{\pi \in \Pi_k} \pi^e(i)$.
  if $\tilde{\pi}_{k,i,e}^e(i) \ge \frac{1}{\sqrt{T}}$ **then**
    Let $A_k = A_k \cup \{\tilde{\pi}_{k,i,e}\}$
  else
    Remove i from I_k^e.
  end if
end for
Output A_k.

4.1.4 Probability Estimates.

Within epoch k, we run a set of "good-and-exploratory policies" with equal shares of τ_k and then update the estimates of F_1 and F_2. This will in return help the algorithm update the set of "good-and-exploratory policies" as the next epoch starts.

Estimate acceptance probabilities:

Initialize $M_{k,e}(i) = N_{k,e}(i) = 0, \forall i \in [d], e = 1, 2$.
for each policy $\pi \in A_k$, **do**
  Run π for a batch of $\frac{\tau_k}{|A_k|}$ rounds.
  For each time a price v_i is *proposed* in G_e, set $M_{k,e}(i) += 1$.
  For each time a price v_i is *accepted* in G_e, set $N_{k,e}(i) += 1$.
end for
For $e = 1, 2$, set $\bar{F}_{k,e}(i) = \max\{\frac{N_{k,e}(i)}{M_{k,e}(i)}, \hat{F}_{\min}\}$ for $i \in I_k^e$, and $\bar{F}_{k,e}(i) = \hat{F}_{\min}$ otherwise.
Output vectors $\bar{F}_{k,e}$ for $e = 1, 2$.

Policy Eliminations. At the end of each epoch k, we update the candidate policy set by eliminating those substantially sub-optimal policies: Firstly, we select an empirical optimal policy $\hat{\pi}_{k,*}$ that maximizes $R(\pi, \hat{F}_{k,1}, \hat{F}_{k,2})$ while guaranteeing $S(\pi, \hat{F}_{k,1}, \hat{F}_{k,2}) \le \delta_{k,s}$. After that, we eliminate those policies that meet one of the following two criteria:

- Large unfairness:

$$S(\pi, \hat{F}_{k,1}, \hat{F}_{k,2}) > \delta_{k,s},$$

- Large regret:

$$R(\pi, \hat{F}_{k,1}, \hat{F}_{K,2}) < R(\hat{\pi}_{k,*}, \hat{F}_{k,1}, \hat{F}_{k,2}) - \delta_{k,r} - L \cdot \delta_{k,s}.$$

Here we adopt two subtractors on the regret criteria: $\delta_{k,r}$ for the estimation error in $R(\pi)$ caused by $\hat{F}_{k,e}$, and $L \cdot \delta_{k,s}$ for the possible increase of optimal reward by allowing $S(\pi) \leq \delta_{k,s}$ instead of $S(\pi) = 0$. In this way, we can always ensure the optimal policy π_* (i.e., the solution of (5)) to remain and also guarantee the other remaining policies perform similarly to π_*.

4.2 Computational Cost

Here we show that our FPA algorithm is efficient in computation.

On the one hand, FPA is *oracle-efficient* due to the doubling-epoch design, as we only run each oracle and update each parameter for $O(\log T)$ times. On the other hand, these arg-max oracles can also be implemented in a time-efficient way, although both (7) and (8) are highly non-convex on the constraints. In fact, a sufficient condition of solving (7) is to solve the following constrained optimization problem:

$$\begin{aligned}(\hat{\pi}_{k,*}, w^*) = & \mathop{\arg\max}_{\pi \in \Pi_k, w \in [0, \frac{1}{\hat{F}_{\min}}]} R(\pi, \hat{F}_{k,1}, \hat{F}_{k,2}), \\ s.t. \quad & v^\top F_1 \pi^1 = w \cdot \mathbb{1}^\top F_1 \pi^1, \\ & v^\top F_2 \pi^2 \geq (w - \delta_{k,s}) \cdot \mathbb{1}^\top F_2 \pi^2, \\ & v^\top F_2 \pi^2 \leq (w + \delta_{k,s}) \mathbb{1}^\top F_2 \pi^2.\end{aligned} \tag{9}$$

Since $w \in [0, \frac{1}{\hat{F}_{\min}}]$ is a scalar, the optimization problem in (9) is a linear programming for any fixed w. Therefore, we may solve this problem by conducting a linear search over a series of $\{w_\ell\} \subset [0, \frac{1}{\hat{F}_{\min}}]$ (since $\hat{F}_{\min} \leq F_{\min}$) and solve the linear programming at each fixed $w = w_\ell$, which can be implemented as Algorithm 2.

Notice that we choose the linear searching step $\epsilon = \frac{\delta_{k,s}}{2}$ due to the Lipschitzness of the objective function and all constraints. In this way, the discretization error is upper bounded by $O(\delta_{k,s})$, which is in the same order of algorithmic regret w.r.t. T and d (as we will show in the proof of Theorem 6). Since we only solve the linear programming problem for $O(\frac{1}{\delta_{k,s}})$ times (which is a polynomial of T and d) and a linear programming problem can be solved in polynomial time, we know that Algorithm 2 is time-efficient. Similarly, we have the following Algorithm 3 that solves (8) efficiently as well.

5 Regret and Unfairness Analysis

In this section, we analyze the regret and unfairness of our FPA algorithm. We first present an $\tilde{O}(\sqrt{T} d^{\frac{3}{2}})$ regret upper bound along with an $\tilde{O}(\sqrt{T} d^{\frac{3}{2}})$ unfairness upper

Algorithm 2 Oracle: Empirical Optimal

Input: Matrix $\hat{F}_{k,1}, \hat{F}_{k,2}$, policy set Π_k, parameter $\hat{F}_{\min}$, allowed estimation error $\delta_{k,s}$, step length $\epsilon = \frac{\delta_{k,s}}{2}$.
Initialization: Pick a $\hat{\pi}_{k,*} \in \Pi_k$ arbitrarily.
for $\ell = 0, 1, 2, \ldots$ **do**
  Let $w_\ell = \ell \cdot \epsilon$
  if $w_\ell > \frac{1}{\hat{F}_{\min}}$ **then**
    Break.
  end if
  Solve the following linear program and get $\hat{\pi}_{k,*}$.

$$\begin{aligned}\hat{\pi}_{k,\ell,*} = & \mathop{\arg\max}_{\pi \in \Pi_k} R(\pi, \hat{F}_{k,1}, \hat{F}_{k,2}), \\ s.t. \quad & v^\top F_1 \pi^1 = w_\ell \cdot \mathbb{1}^\top F_1 \pi^1, \\ & v^\top F_2 \pi^2 \geq (w_\ell - \delta_{k,s}) \cdot \mathbb{1}^\top F_2 \pi^2, \\ & v^\top F_2 \pi^2 \leq (w_\ell + \delta_{k,s}) \mathbb{1}^\top F_2 \pi^2.\end{aligned}$$

  {Compare with best existing solution.}
  if $R(\hat{\pi}_{k,\ell,*}, \hat{F}_{k,1}, \hat{F}_{k,2}) > R(\hat{\pi}_{k,*}, \hat{F}_{k,1}, \hat{F}_{k,2})$ **then**
    Substitute $\hat{\pi}_{k,*} \leftarrow \hat{\pi}_{k,\ell,*}$.
  end if
end for
Return $\hat{\pi}_{k,*}$.

Algorithm 3 Oracle: Policy Elimination

Input: Matrix $\hat{F}_{k,1}, \hat{F}_{k,2}$, policy set Π_k, parameter $\hat{F}_{\min}$, $\delta_{k,s}$, $\delta_{k,r}$, constant L, policy $\hat{\pi}_{k,*}$ step length $\epsilon = \frac{\delta_{k,s}}{2}$.
Initialization: Feasible set $\Pi_{k+1} = \emptyset$.
for $\ell = 0, 1, 2, \ldots$ **do**
  Let $w_\ell = \ell \cdot \epsilon$
  if $w_\ell > \frac{1}{\hat{F}_{\min}}$ **then**
    Break.
  end if
  Solve the following linear program and get a feasible set $\Pi_{k+1,w}$.

$$\begin{cases} v^\top F_1 \pi^1 = w_\ell \cdot \mathbb{1}^\top F_1 \pi^1, \\ v^\top F_2 \pi^2 \geq (w_\ell - \delta_{k,s}) \cdot \mathbb{1}^\top F_2 \pi^2, \\ v^\top F_2 \pi^2 \leq (w_\ell + \delta_{k,s}) \mathbb{1}^\top F_2 \pi^2 \\ R(\pi, \hat{F}_{k,1}, \hat{F}_{k,2}) \geq R(\hat{\pi}_{k,*}, \hat{F}_{k,1}, \hat{F}_{k,2}) - \delta_{k,r} - L \cdot \delta_{k,s}. \end{cases}$$

  Update $\Pi_{k+1} \leftarrow \Pi_{k+1} \cup \Pi_{k+1,w}$
end for
Return Π_{k+1}.

bound. Then we show both of them are optimal (w.r.t. T) up to $\log\log T$ factors by presenting matching lower bounds.

5.1 Regret Upper Bound

First of all, we propose the following theorem as the main results for our Algorithm 1 (FPA).

Theorem 6 (Regret and Unfairness)**.** *FPA guarantees an $O(\sqrt{T}d^{\frac{3}{2}}\log\frac{d\log T}{\epsilon})$ regret with no procedural unfairness and an $O(\sqrt{T}d^{\frac{3}{2}}\log\frac{d\log T}{\epsilon})$ substantive unfairness with probability $1-\epsilon$.*

Proof sketch. We prove this theorem by induction w.r.t. epoch index k. Firstly, we start with the induction assumption that $\pi_* \in \Pi_k$, which naturally holds as $k=1$. Meanwhile, we show a high-probability bound on the estimation error of each $F_e(i)$ for epoch k, according to concentration inequalities. With this, we derive the estimation error bound of $R(\pi, F_1, F_2)$ and $S(\pi, F_1, F_2)$ for each policy $\pi \in \Pi_k$ in epoch k, given that we are always taking a policy that maximizes the probability of proposing a specific price. After that, we bound the regret and unfairness of each policy remaining in Π_{k+1}, and therefore bound the regret and unfairness of epoch $(k+1)$ with high probability. Finally, we show that the optimal fair policy π_* (defined in (5)) is also in Π_{k+1}, which matches the induction assumption for Epoch $(k+1)$. By adding up these performance over epochs, we get the cumulative regret and unfairness respectively. Please refer to Appendix B.1 for a detailed proof. ■

Remark 7. *Our algorithm guarantees $O(\sqrt{T}\log\log T)$ regret and unfairness simultaneously, whose average-over-time match the generic estimation error of $O(\frac{1}{\sqrt{T}})$. It implies that these fairness constraints do not bring informational obstacles to the learning process. In fact, these upper bounds are tight up to $O(\log\log T)$ factors, which are shown in Theorem 8 and Theorem 9.*

5.2 Regret Lower Bound

Here we show the regret lower bound of the pricing problem.

Theorem 8 (Regret lower bound)**.** *Assume $d \le T^{\frac{1}{3}}$. Given the online two-group fair pricing problem and the regret definition as (6), any algorithm would at least suffer an $\Omega(\sqrt{dT})$ regret.*

We may prove Theorem 8 by a reduction to online pricing problem with no fairness constraints: Given a problem setting where the two groups are identical, i.e. $F_1(i) = F_2(i), \forall i \in [d]$, and let $q = 0.5$. Notice that any policy satisfying $\pi^1 = \pi^2$ is procedurally and substantively fair, and the optimal policy is to keep proposing the best fixed price. Therefore, this can be reduced to an online identical-product (i.e., non-contextual) pricing problem, and we present a bandit-style lower bound proof in Appendix B.2 inspired by Auer et al. (2002b).

5.3 Unfairness Lower Bound with Optimal Revenue

Here we show that any optimal algorithm has to suffer an $\Omega(\sqrt{T})$ substantive unfairness.

Theorem 9 (Substantive Unfairness Lower Bound)**.** *For any constant C_x, there exists constants $C_u > 0$ such that any algorithm with an $C_x \cdot T^{\frac{1}{2}}$ cumulative regret and zero procedural unfairness has to suffer an $C_u \cdot T^{\frac{1}{2}}$ substantive unfairness.*

It is worth mentioning that this result is different from ordinary lower bounds on the regret, as it also requires the algorithm to be optimal. In general, we propose 2 different problem settings, and we show the following four facts:

- No algorithm can perform well (i.e. low regret and low substantive unfairness) in both settings.
- Any algorithm cannot efficiently distinguish between the two settings.
- Failing in distinguishing between them would lead to a large substantive unfairness.
- Avoid distinguishing between them would suffer even larger regret or substantive unfairness.

In order to prove these, we make use of Example 1 presented in Section 1. One of the settings is exactly Example 1, and the other one is identical to it except these 0.5 acceptance rates (i.e., \$0.7 for G_1 and \$1 for both groups) are now $(0.5-\zeta)$ in both groups. We get close-form solutions to both problem settings. We further show that the two settings are indistinguishable in information theory, and we either fail or avoid distinguishing them at least for $\Omega(T)$ rounds. Also, we show that the (reward or fairness) loss of avoiding distinguishing is more than $\Omega(\frac{1}{\sqrt{T}})$, and that the fairness loss of failing to distinguish is $\Omega(\frac{1}{\sqrt{T}})$. As we have a tight budget on the cumulative regret, we have to suffer a $\Omega(T\cdot\frac{1}{\sqrt{T}}) = \Omega(\sqrt{T})$ unfairness lower bound as a trade-off. Please refer to Appendix B.3 for more details.

6 Discussion

Here we discuss some open issues and potential extensions of this work. For more discussions on settings, techniques and social impacts, please refer to Appendix C.

Improvement on Technical Assumptions. In this work, we assume the existence of a lower bound $F_{\min} > 0$ of the acceptance rate of all prices for both groups. This assumption is stronger than our expectation, as the seller would not know the highest price that customers would accept. We assume this for two reasons: (1) Without assuming $F_e(i) > 0$, the substantive unfairness function might be undefined. For instance, if a pricing policy is completely unacceptable in G_1 (with *no* accepted prices) but is acceptable in G_2, then

is it a fair policy? (2) With a constantly large probability of acceptance, we can estimate every $F_e(i)$ and bound it away from 0 and therefore leads to the Lipschitzness of $S(\pi, \hat{F}_1, \hat{F}_2)$. However, there might exist an algorithm that works for $F_e(i) > 0$ generally and maintains these optimalities as well, which is an open problem to the future.

Feelings of Fairness in FPA. In algorithm 1, notice that we run each $\tilde{\pi} \in A_k$ for a continuous batch of $\frac{\tau_k}{|A_k|} = \Omega(\sqrt{T} \cdot 2^k)$, which is long enough for customers to experience the fairness: Customers would figure out their average seeing and buying prices by comparing these amounts with customers from the other group. On the contrary, if we run an adaptive algorithm that changes the running policy at every time, the customers would feel like they are treated by different policies without knowing that all policies are quite fair (since now each customer has no comparison).

Relaxation on Substantive Fairness. In this work, our algorithm approaches the optimal policy (as the solution of (5)) through an online learning framework. This ensures an asymptotic fairness as $T \to +\infty$, but we still cannot guarantee perfect any-time fairness precisely (i.e., $S(\pi_t) = 0, \forall t$). Therefore, it is more practical to consider the following inequality-constraint optimization problem:

$$\begin{aligned} \pi_{\delta,*} = & \operatorname*{argmax}_{\pi=(\pi^1,\pi^2)\in\Pi} R(\pi; F_1, F_2) \\ s.t. \quad & U(\pi) = 0, \quad S(\pi; F_1, F_2) \le \delta. \end{aligned} \tag{10}$$

Comparing (5) with (10), we know that $R(\pi_*) \le R(\pi_{\delta,*})$. According to Lemma 14, we further know that $R(\pi_*) \ge R(\pi_{\delta,*}) - L \cdot \delta$. Naturally, the substantive unfairness definition is now $\max\{0, S(\pi; F_1, F_2) - \delta\}$. If we still consider this problem under the framework of online learning, then two questions arose naturally: What are the optimal regret rate and (substantive) unfairness rate like? And how can we achieve them simultaneously? From our results in this work, we only know that (1) If $\delta = 0$, then both rates are $\Theta(\sqrt{T})$, and (2) if $\delta \ge 1$, then the optimal regret is $\Theta(\sqrt{T})$ and the optimal unfairness is 0 (as it is reduced to the unconstrained pricing problem). In fact, for $\delta = O(\sqrt{1/T})$, we may still achieve $O(\sqrt{T})$ regret and unfairness, but it is not clear if they are always optimal. For $\delta > \sqrt{1/T}$, we conjecture that the optimal regret is still $\Theta(\sqrt{T})$ and the optimal unfairness could be $\Theta(1/(\sqrt{T}\delta))$.

Trade-offs between Procedural and Substantive Fairness. We conjecture that it is not likely to trade-off substantive fairness with procedural fairness, as the lower bound on substantive fairness comes from the indistinguishability between two similar settings. To set unfair prices intentionally would not substantially speed-up the learning process and make the two environments more distinguishable.

Optimal Policy on the Continuous Space. In this work, we restrict our price choices in a fixed price set $\mathbf{V} = \{v_1, v_2, \ldots, v_d\}$ and aims at the optimal distributions on these v_i's. However, if we are allowed to propose any price within $[0, 1]$, then the optimal policy could be a tuple of two *continuous* distributions that outperforms any policy restricted on $\mathbf{V}$. Even if we know that customers' valuations are all from $\mathbf{V}$, the optimal policy is not necessarily located inside $\mathbf{V}$ due to the fairness constraints. This optimization problem is even harder than (5), and the online-learning scheme further increases its hardness. Existing methods such as continuous distribution discretization (Xu and Wang, 2022) might work, but would definitely lead to an exponential time complexity.

From Two Groups to Multi Groups Our problem setting assumes that there are two groups of customers in total. We choose to study a two-group setting to simplify the presentation. In practice, however, it is very common that customers are coming from many groups with different valuations even on the same product. In fact, we believe it straightforward to extend our techniques and results to G-group settings, as long as we determine a metric of multi-group unfairness. For instance, if we choose to define the multi-group unfairness as the summation of pairwise unfairness of all $O(G^2)$ pairs of groups, we may adjust our algorithm by lengthening each epoch by $G/2$ times and keeping everything the same as in this paper. In this way, the upper regret bound would be $\tilde{O}(G^3\sqrt{T}d^{2/3})$, which is $O(G^3)$ times larger than the existing regret bound. Therefore, it is still optimal w.r.t. T up to iterative-log factors. However, this notion of pairwise unfairness is somewhat controversial in multi-group settings, and we will provide more discussions regarding this generalization on appendix C.1.

7 Conclusion

In this work, we study the online pricing problem with fairness constraints. Specifically, we introduce two fairness notions, a *procedural fairness* and a *substantive fairness*, which respectively ensure the equality of proposed and accepted prices between two different groups. To satisfy these two constraints simultaneously, we adopt *random* pricing policies and establish the objective function and rewards in expectation. To solve this problem with unknown demands, we develop a policy-elimination-based algorithm FPA that achieves an $\tilde{O}(\sqrt{T})$ regret with zero procedural unfairness and within an $\tilde{O}(\sqrt{T})$ substantive unfairness. We show that our algorithm is optimal in both regret and unfairness up to $\log\log T$ factors, by proving an $\Omega(\sqrt{T})$ regret lower bound and an $\Omega(\sqrt{T})$ unfairness lower bound for any optimal algorithm with $O(\sqrt{T})$ regret.

Acknowledgment

The authors are very grateful to Xi Chen and Yining Wang for their helpful discussions and suggestions. This research is partially supported by the Adobe Data Science Award and a start-up grant from the UCSB Department of Computer