ONLINE LEARNING WITH A VIEW TOWARD FAIRNESS

A Dissertation
Presented to
The Academic Faculty

By

Jad Salem

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Mathematics
Algorithms, Combinatorics, and Optimization

Georgia Institute of Technology

April 2023

ONLINE LEARNING WITH A VIEW TOWARD FAIRNESS

Thesis committee:

Dr. Swati Gupta (advisor)
Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Mohit Singh
Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Vijay Kamble (reader)
College of Business Administration
University of Illinois Chicago

Dr. Santosh Vempala
College of Computing
Georgia Institute of Technology

Dr. Deven Desai
College of Business
Georgia Institute of Technology

Dr. Greg Blekherman
School of Mathematics
Georgia Institute of Technology

Date approved: TBD

For my family and friends

ACKNOWLEDGMENTS

TBD

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACRONYMS

Abbr.	Meaning	Reference
IF	Individual Fairness	Section 2.1.2
CDP	Conditional Demographic Parity	Definition 2.3
OF	Ordinal Fairness	Definition 3.3
MF	Meritocratic Fairness	Section 3.1.1
CFTD	Comparative Fairness at the Time of Decision	Definition 4.1

SUMMARY

The field of fairness in algorithmic decision-making can be traced back at least 25 years [1] and has flourished over the past decade [2]. However, the overwhelming majority of this research concerns offline decision-making for a set of individuals (e.g., deciding whom to grant loans to, or how to rank a set of job applicants), where the decision-maker observes a multiset of contexts capturing the socio-economic backgrounds of a diverse set of people and makes decisions on these contexts en masse. In practice, however, decisions are often made in real time with only partial information (e.g., resume filtering decisions are often made upon submission [3]). In this dissertation, I use the application domains of hiring and pricing to explore the themes of *uncertainty* and *comparative fairness* in online-decision-making. What does it mean to make fair decisions in real time? Can stringent fairness constraints prevent convergence to a good decision in online learning settings, thus hindering learning? How can one design online algorithms satisfying a given notion of fairness? I partially answer these questions in hiring and pricing by considering (1) capacitated online selection (specifically, the k-secretary problem) given partial ordinal rankings, motivated by applicant-screening, and (2) stochastic convex optimization with bandit feedback, motivated by multi-segment demand learning. The contents of this dissertation include joint work with Swati Gupta, Vijay Kamble, and Deven Desai.

The first line of inquiry I will discuss revolves around the question of how to address uncertainty in online decision-making. Specifically, I focus on the task of online applicant-screening where uncertainty (e.g., noise in evaluations, or knowledge that certain pairwise rankings may be corrupted due to bias) is captured by a partial order (i.e., a partial ranking) on the set of applicants. Modeling the screening process as a secretary problem with a capacity constraint [4], I prove lower bounds and upper bounds on the competitive ratio (i.e., the worst-case expected performance ratio). To prove the upper bound, I develop a novel way of thresholding in partial orders, which may have wider applications beyond compet-

itive analysis for secretary problems. Under stricter modeling assumptions (in particular, when applicants come from disjoint demographic groups and bias is group-specific), I provide simpler algorithms and analysis. Finally, I test these methods on a real-world dataset on employability outcomes for a pool of candidates based in India [5]. Using the algorithms I have developed in this part of the thesis, I show that the poset-based algorithms are less sensitive to distributional changes in data than bias-agnostic algorithms but more sensitive than quota-based algorithms; thus, poset-based methods provide a way to move toward "equal opportunity" without employing stringent and potentially illegal methods such as quotas.

The second line of inquiry revolves around the question of how to achieve some notion of comparative fairness (i.e., ensuring that similar individuals receive similar decisions) in an online learning setting. The challenge here is that stringent fairness constraints can prevent learning [6], thus resulting in poor performance. Specifically, I consider the problem of stochastic convex optimization with bandit feedback, where the utility function is smooth, strongly convex, and separable, and at each time period t, the decision maker chooses a point x_t in $\mathbb{R}^N$. This setting is motivated by multi-segment dynamic pricing, where prices are assigned to N segments (e.g., "youth" and "adult" for movie ticket pricing) at each time period, and noisy revenue is observed. I introduce a novel relaxation of individual fairness [7] which allows decisions to monotonically vary over time while maintaining one-sided comparative fairness across dimensions, thus balancing fairness with learnability. Under this constraint, I provide algorithms which attain big-Oh optimal regret for $N = 1, 2$ and sublinear regret for $N > 2$.

Finally, I turn to the law: what is allowed in addressing fairness and bias in algorithmic decision-making? To begin, I focus on the legality of fairness interventions in hiring in the U.S. In the U.S., protected information, such as race, can be used in hiring in limited contexts. One such context, affirmative action, is a temporary measure designed to counteract the effects of historic discrimination. Drawing on legislation and relevant affirmative action caselaw, I (1) discuss what may or may not be allowed in addressing discrimination in hiring, and (2) show that precedent supports the use of partial orders to account for uncertainty in hiring. Turning to dynamic pricing, I discuss open questions regarding (a) how privacy laws in the U.S. [8] and Europe [9] affect contextual pricing, and (b) how price gouging laws can be translated into temporal fairness constraints.

The results in this thesis (1) have been published in WINE 2020, WINE 2022, and FAccT 2022, (2) have been accepted for publication in Management Science, and (3) have received a Major Revision in Operations Research.

CHAPTER 1
INTRODUCTION

This chapter contains excerpts from a soon-to-be-published book chapter [10].

It is well-known that automated decision-making (e.g., decision-making using machine learning algorithms) is susceptible to errors and biases (e.g., [11]). With the increased adoption of automated decision-making practices in domains ranging from recommendation systems [12] and retail pricing [13] to banking [14], criminal justice [15], and healthcare [16], there has been a corresponding surge of interest in the academic community and among policymakers alike on the topic of *algorithmic fairness*, i.e., ensuring that humans are treated fairly and equitably by these systems [17, 18].

Strategies for reducing the impact of errors and biases naturally differ between automated and non-automated decision-making. While non-automated decision-making has relied primarily on trainings and process engineering [19], automation provides new and powerful tools that can be exploited to detect and mitigate errors. Data on decisions made and on the accuracy of those decisions can easily be tabulated, and errors can be compared across groups. With automated methods, one can encode notions of fairness as constraints or objectives and compute good or even optimal scorers or classifiers (e.g., [20]). Such a task would be intractable to a human decision-maker.

Automated decision-making, however, commonly takes place *over a period of time* [21]. A large company will receive job applications consistently over time and must make screening decisions as they arrive. Loan granting decisions are similarly made over time as applications arrive. Goods are priced periodically [22]. Not only are decisions made over time in each of these cases, but data changes over time as well. For example, as loan grantees pay back or default on their loans, this information can be used to inform future loan granting decisions. This raises important, complex questions about how existing (static) notions of fairness should be adapted to settings where data and decisions are *dynamic*.

To begin with, the introduction of time leads to **normative questions** about the meaning of fairness. Suppose that in the context of applicant screening, we have convinced ourselves that equalizing selection rates across groups (i.e., demographic parity [23]) is "fair" in the static setting. What, then, should fairness mean in a dynamic setting, where decisions are made over time, and feedback is received over time? Is it fair to equalize selection rates across groups across all time periods? This might involve constraining decisions made today by decisions made 10 years ago. For example, in a profession dominated by men, striving for demographic parity *across all time* would require selecting no men until cumulative selections (or selection rates) are equal. If we instead wanted demographic parity within a sliding window, we could require that selections (or selection rates) over the past, say, year, be equal. Which of these approaches, if any, is a fair way to make applicant screening decisions over time?

To complicate matters further, the introduction of time results in **fairness-learnability** trade-offs [6]. In many sequential decision-making problems, the "rewards" of different decisions are often not known in advance and must be learned over time through exploration. For example, in dynamic pricing with demand learning, the maximum-revenue price point is not known in advance, but is estimated over time by experimenting at different prices. If a fairness constraint is very stringent, our ability to learn and converge to an optimal decision can be hindered, and performance can suffer. Further, in contextual problems, decisions are made without full contextual information of the population [24]. For instance, in a rolling admissions scenario, a decision is made on the current applicant without knowledge of the quality of future applicants. These considerations should inform how we define fairness in sequential decision-making settings.

As this discourse develops, various task-specific notions of fairness have been introduced [25, 26, 27, 28, 29]. This is natural since the social backdrop of, say, loan-administration is very different from that of job applicant screening. Hence, the notions of fairness appropriate in different decision-making settings such as pricing, applicant-screening, loan-administration, etc., may be different. For instance, one may wish to equalize false negative rates of a cancer diagnosis algorithm across racial groups to ensure that no racial group is underdiagnosed. For a criminal risk prediction algorithm, on the other hand, it may be desirable to equalize false positive rates across groups to ensure that no group is under disproportionate scrutiny. In this dissertation, I discuss technical and practical aspects of fairness in online decision-making through the lenses of hiring and pricing.

In Chapter 2, I give relevant background information which will be used throughout the dissertation. This will include two frameworks for online decision-making, an introduction to partial orders and convex analysis, and two classical algorithms for the decision-making frameworks discussed in Chapters 3 and 4, respectively. In Chapter 3, I discuss the *secretary framework*, in which adversarially chosen inputs are presented to an algorithm in random order. This framework can be used to model the applicant screening. In this chapter, I propose a novel notion of fairness which accounts for uncertainty in evaluations, and I design and analyze algorithms subject to that constraint. In Chapter 4, I consider fairness in online learning (specifically in bandit convex optimization), motivated by a multi-segment pricing problem. I discuss trade-offs between fairness and learning in such settings, and I design and analyze algorithms satisfying a time-relaxed notion of comparative fairness. In Chapter 5, I discuss legal constraints on algorithm design; specifically, I discuss how law and caselaw in the U.S. support the use of protected information in hiring in limited circumstances, and how the law restricts hiring decisions in the presence of uncertainty. These findings directly support the algorithms presented in Chapter 3.

CHAPTER 2
BACKGROUND AND NOTATION

In this chapter, I present background knowledge which will be useful throughout the dissertation. I begin by discussing classical notions of fairness in Section 2.1. Next, in Section 2.2, I present two frameworks for online-decision-making: the secretary framework, which is discussed in Chapter 3, and the online learning framework, which is discussed in Chapter 4. In Section 2.3, I discuss three algorithms for the classical secretary problem. In Section 2.4, I discuss convex optimization, including analysis of the classical gradient descent algorithm, which will be useful in Chapter 4. Finally, in Section 2.5, I discuss partial orders, which will be important in Chapters 3 and 5.

2.1 Fairness in offline decision-making

Consider a generic offline decision-making setting where each input has a context $c \in \mathcal{C}$, for some context set $\mathcal{C}$. This set $\mathcal{C}$ may encode demographic information (e.g., $\mathcal{C} = \{< 40 \text{ years old}, \geq 40 \text{ years old}\}$) and other quality information of the input. The decision-maker observes a sequence of contexts $(c_1, \ldots, c_n) \in \mathcal{C}^n$ and chooses decisions

$$(x_1, \ldots, x_n) \in \mathcal{X}^n,$$

for some decision set $\mathcal{X}$. Here, x_i is the decision given to the ith context c_i.

This decision set $\mathcal{X}$ could be categorical or numerical. For example, when screening applicants, we could have $\mathcal{X} = \{\text{yes}, \text{no}\}$, where "yes" means that we admit the applicant to the next round of the hiring process. When deciding the risk of recidivism, the decision set might be $\mathcal{X} = [0, 1]$, where higher decisions correspond to a higher likelihood of recidivism. When $\mathcal{X}$ is categorical, the decision process is called *classification*, and when $\mathcal{X}$ is numerical, it is called *regression*. I discuss fairness in both cases next.

2.1.1 Classification

There are many notions of "equal treatment" or "equal opportunity" in classification across demographic groups [17]. In this section, I will introduce some well-known notions which will be relevant to this dissertation. For the sake of exposition, I will assume that there are two disjoint demographic groups G_1, G_2, with $\mathcal{C} = G_1 \cup G_2$, and that decisions are binary: $\mathcal{X} = \{0, 1\}$.[1]

The simplest notion of equality across groups is *demographic parity*, which requires that both groups have the same positive decision rate.

Definition 2.1 (demographic parity (deterministic))**.** Using the notation from above, the decisions $(x_1, \ldots, x_n)$ on contexts $(c_1, \ldots, c_n)$ satisfy *demographic parity* if $\frac{\sum_{i:c_i \in G_1} \mathbb{1}[x_i=1]}{\sum_{i:c_i \in G_1} 1} = \frac{\sum_{i:c_i \in G_2} \mathbb{1}[x_i=1]}{\sum_{i:c_i \in G_2} 1}$.

It is often useful for the analysis of algorithms to assume that contexts are randomly generated. Demographic parity can be rephrased to allow for such randomness.[2]

Definition 2.2 (demographic parity (stochastic) [17])**.** Suppose each context $C \in \mathcal{C}$ observed by the decision-maker is generated independently from the same distribution, and the decision $X = X(C) \in \mathcal{X}$ is a random variable. Then the decision process satisfies *demographic parity* if $\mathbb{P}(X = 1 \mid C \in G_1) = \mathbb{P}(X = 1 \mid C \in G_2)$.

Demographic parity enforces equality of outcomes across groups without regard to other feature information. In cases where prevalence varies across groups, decision rules satisfying demographic parity can be sub-optimal in terms of accuracy.[3] To ensure some

[1] The notions of fairness discussed in this section easily extend to multiple classes and multiple groups.

[2] Indeed, the deterministic version of demographic parity is often written in terms of probabilities, although this is an abuse of notation.

[3] "Sub-optimal," here, is with respect to the optimal classifier. In many cases, only noisy estimates of the inputs' true classes can be observed; in such cases, enforcing demographic parity could potentially improve the quality of selections [30].

regularity in decisions across groups while allowing better treatment for better contexts, one could instead enforce *conditional demographic parity* [31].

To motivate conditional demographic parity, suppose each input C has some feature $R(C) \in \mathbb{R}$ which the decision-maker would like to respect. For example, $R(C)$ might be an interview score given to a job candidate, and the decision-maker might believe that those with higher interview scores are stronger candidates. Thus if G_1 has higher R values than G_2 on average, then the decision-maker would like to allow for more selections from G_1. Under conditional demographic parity, for any r, demographic parity is satisfied among the inputs with $R(C) = r$.

Definition 2.3 (conditional demographic parity (stochastic) [31])**.** Suppose each context $C \in \mathcal{C}$ observed by the decision-maker is generated independently from the same distribution, and the decision $X = X(C) \in \mathcal{X}$ is a random variable. Then the decision process satisfies *conditional demographic parity* if for every $r \in \text{supp}(R)$,

$$\mathbb{P}(X = 1 \mid C \in G_1, R(C) = r) = \mathbb{P}(X = 1 \mid C \in G_2, R(C) = r).$$

If R is discrete, then a deterministic version of CDP (similar to Definition 2.1) can be stated as well. Note that one can obtain other group notions of fairness by enforcing equality of any statistical quantity (e.g., those involving false positives and false negatives) across groups. Different scenarios may warrant specific notions of fairness over others, and a plethora of papers have explored methods for attaining these notions of fairness. Importantly, statistical notions of fairness are sometimes mutually incompatible (e.g., see [32, 33, 34]), which means that one must choose which notions to enforce.

Strictly enforcing statistical notions of fairness may be impractical, since (1) performance can suffer, and (2) often only the discrete decisions are available, and equality across groups can not be obtained due to rounding issues. In response to these issues, relaxed notions of group fairness have emerged (e.g., see [35]). For example, if one wishes to ensure that $Q(G_1) = Q(G_2)$, where $Q(\cdot)$ is some statistical measure of performance, then one relaxation is to ensure that $|Q(G_1) - Q(G_2)| \leq \varepsilon$. Multiplicative measures of fairness can be considered as well.

2.1.2 Regression

Here, we assume that decisions are numerical, not categorical. For example, c_i could encode medical information of patient i, and x_i could represent the dosage of a medicine given to patient i. As opposed to the case of classification, the concept of "positive labels" or "false positives" do not necessarily make sense in regression. In this case, one common interpretation of fairness is to ensure that similar inputs receive similar decisions. If c_1 and c_2 are "close" in some sense, then x_1 and x_2 should be close in some sense as well.

To formalize this "closeness," we will assume that there are distance functions $d_{\mathcal{C}} : \mathcal{C} \times \mathcal{C} \to [0, \infty]$ and $d_{\mathcal{X}} : \mathcal{X} \times \mathcal{X} \to [0, \infty]$ on the context space $\mathcal{C}$ and the decision set $\mathcal{X}$, respectively. Thus we can formulate a notion of fairness, which we call *comparative fairness*, as follows.

Definition 2.4 (L-comparative fairness)**.** Let $L > 0$, $(c_1, \ldots, c_n)$ a sequence of contexts, and $(x_1, \ldots, x_n)$ the corresponding sequence of decisions. This decision process satisfies *L-comparative fairness* if for all $i, j \in [n]$,

$$d_{\mathcal{X}}(x_i, x_j) \leq L d_{\mathcal{C}}(c_i, c_j).$$

When $d_{\mathcal{C}}$ and $d_{\mathcal{X}}$ are metrics, comparative fairness reduces to the well-known notion of *individual fairness* of Dwork et al. [7].

2.1.3 Gaps in the literature: uncertainty, online decision-making

While the literature on algorithmic fairness is growing rapidly, there are some areas that have received little attention. In this section, I briefly outline two such areas which will be addressed in this dissertation.

Uncertainty

First is that of fairness in the presence of uncertainty. When predictions are prone to larger errors on group G_1 than on group G_2, those in group G_2 may be harmed, as shown in the example below.

Example 2.1. Suppose there are two disjoint groups, G_1 and G_2, and $|G_1| = |G_2| = n$, and suppose that everyone, regardless of group, has a true (latent) score of 0. The decision-maker is tasked with selecting c people from $G_1 \cup G_2$. However, the scores observed by the decision-maker are corrupted by independent additive noise. If the noise were identically distributed, independent of group membership, then the probability that all selections are from G_1 would be 2^{-c}.

Now suppose that the noise from G_1 is uniformly distributed over $[-2, 2]$, and the noise for G_2 is uniformly distributed over $[-1, 1]$. Then

$$\lim_{n \to \infty} \mathbb{P}(\text{all selections are from } G_1) \geq \lim_{n \to \infty} \mathbb{P}(\text{at least } c \text{ members of } G_1 \text{ have noise} > 1) = 1.$$

Thus, despite the fact that the noise is mean-zero for both groups, we see that the disparity in variance is resulting in a stark disparity in treatment.

The use of quotas have been proposed to account for disparities in variance [30], but this approach is blunt and, depending on the application, may come under legal scrutiny. Another approach involves forcing equal treatment whenever two inputs cannot be separated with high enough confidence [24]. In Chapter 3, I will argue that this notion is overly stringent and propose a new method to account for these disparities.

Online decision-making

Despite the fact that online decision-making is commonplace in industry, the literature on algorithm fairness focuses mostly on offline decision-making. While work on online fairness has begun to emerge (e.g., [36, 37]), this field remains underdeveloped. Chapters 3-4 discuss fairness in two online decision-making settings, which are introduced in Section 2.2.

2.2 Frameworks for online decision-making

In order to obtain theoretical results for online algorithms, one must clarify the framework in which the decisions are made. In particular, I consider two frameworks: an online optimization framework in which a fixed set of inputs arrive in random order, and an online learning framework, in which feedback on decisions can be used to inform future decisions.

2.2.1 The Secretary framework

Chapter 3 of this dissertation concerns the secretary problem [38], which can be described as an online optimization problem. I use the term “online optimization” to signal that feedback on decisions is not observed, and so learning (in the traditional sense of supervised learning) does not take place. In secretary problems, the goal is to select a subset of the weighted elements $\{a_1, \ldots, a_N\}$ subject to some constraint, where information about the elements is revealed online and decisions are made online. In generality, this framework can be described as follows:

1. N distinct positive numbers $w(a_1), \ldots, w(a_N)$ are generated by an adversary.

2. A uniform random permutation $\sigma : [N] \to [N]$ is generated.

3. For $t = 1, \ldots, N$:

    (a) Information about $w(a_{\sigma(1)}), \ldots, w(a_{\sigma(t)})$ is revealed.

(b) An irrevocable select/reject decision on $a_{\sigma(t)}$ is made.

As is done for many online optimization problems, to assess the performance of a secretary algorithm, we use the *competitive ratio* [39], which is the worst-case expected performance ratio. For the classical secretary problem, this can be stated as follows:

Definition 2.5 (α-competitive [40])**.** A secretary algorithm $\mathcal{A}$ is *α-competitive*[4] if the expected performance ratio is at least $\frac{1}{\alpha}$. In other words, it is *α-competitive* if for any scoring w,

$$\rho(w) := \mathbb{E}\left[\frac{w(\mathsf{Alg}(w))}{\textsc{opt}(w)}\right] \geq \frac{1}{\alpha}\,,$$

where $\textsc{opt}(I)$ is the weight of the optimal feasible subset (i.e., the largest total score which is achievable with knowledge of the scores of all elements), and $\mathsf{Alg}(w)$ is the output of $\mathcal{A}$. The expectation is taken over the possible orderings of elements and any internal randomness in the algorithm.

In the *classical secretary problem* [38], at most one element can be selected, and line 3(a) in the secretary framework is replaced with "$w(a_{\sigma(t)})$ is revealed."

As a side note, although the secretary problem is not a supervised learning problem, the random ordering assumption makes it feel like a learning problem. Throughout the time horizon, more information on the score distribution is revealed, and better decisions can be made. In many secretary algorithms, the first segment of elements are simply observed and not selected, resembling the training phase of a ML algorithm.

2.2.2 Online learning

The main difference between online optimization (described above) and online learning is the presence of *feedback*. In the online learning framework, decisions are made iteratively over time, and numerical feedback is observed after each decision. In Chapter 4, I discuss

[4]This is referred to as *strictly α-competitive* in [39].

an online learning problem called *bandit convex optimization* [41], in which the goal is to minimize a convex function $f : \mathbb{R}^N \to \mathbb{R}$. In particular, the BCO framework is as follows:

1. For $t = 1, \ldots, T$:
   (a) A decision $x_t \in \mathcal{X} \subseteq \mathbb{R}^N$ is made.
   (b) $f(x_t) + \varepsilon_t$ is observed, where ε_t is mean-zero sub-Gaussian noise.

In online learning problems, performance is typically measured in terms of *regret*, which we define as follows.

Definition 2.6 (regret [41]). Let $\mathcal{F}$ be the class of possible functions disclosed to the algorithm. Let $X_1, \ldots, X_T$ be the (random) decisions made by a BCO algorithm. The *regret*[5] of this algorithm is

$$\sup_{f \in \mathcal{F}} \mathbb{E}\left[\sum_{t=1}^{T} f(X_t) - T \min_{x \in \mathcal{X}} f(x)\right].$$

Since the time horizon T can be quite large, online learning algorithms are often compared by the big-Oh class of their regret. For example, suppose we are given that $\mathcal{F}$ is the set of one-dimensional convex functions which are 2-Lipschitz over $\mathcal{X} = [-1, 1]$, and the decisions made by an algorithm are deterministically $x_1 = x_2 = \cdots = x_T = 1$. By considering the function $f(x) = x^2 \in \mathcal{F}$, we see that the regret of this algorithm is at least T. Since $f(x) - f(y)$ is bounded due to Lipschitzness, the regret of any algorithm is $\mathcal{O}(T)$, which means that the algorithm in question can be classified as an $\Theta(T)$-regret algorithm.

Typically, the set of possible objective functions $\mathcal{F}$ and the decision set $\mathcal{X}$ are chosen so that the instantaneous regret at any time t (that is, $f(X_t) - \min_{x \in \mathcal{X}} f(x)$) is bounded by a constant. In this case, any algorithm would attain $\mathcal{O}(T)$ regret; so, an algorithm having linear regret (i.e., one with a regret of $\Omega(T)$) has the worst-possible regret, in terms of big-Oh analysis.

[5] In [41], this is referred to as *expected regret*.

2.2.3 Relationship between competitive ratio and regret

I introduced two performance measures for online decision-making in this section: the competitive ratio, in Section 2.2.1, and regret, in Section 2.2.2. To better understand the difference between these two measures, I express both using the same notation.

To that end, let $c = (c_1, \ldots, c_T) \in \mathcal{C}^T$ be the contexts seen at times $t = 1, \ldots, T$. In a non-contextual problem like the Secretary problem, we would have $|\mathcal{C}| = 1$. Let $x = (x_1, \ldots, x_T) \in \mathcal{X}^T$ be the decisions made at times $t = 1, \ldots, T$. In the case of the secretary problem, we have $\mathcal{X} = \{0, 1\}$, where a decision of 1 corresponds to an acceptance. To account for the fact that not every sequence of decisions is feasible, we let $\mathcal{X}' \subseteq \mathcal{X}^T$ denote the feasible sequences. For example, in the classical secretary problem, we would have $\mathcal{X}' = \{x \in \{0,1\}^T : \sum_t x_t \leq 1\}$.

Now let $F(c, x)$ denote the cost of the decision sequence x given the context sequence c, and let $\mathcal{F}$ denote the possible cost functions. For example, in the (non-contextual) convex optimization settings, we can have $F(x) = \sum_t f(x_t)$, where f is a convex function; in the secretary setting, we can have $F(c, x) = \sum_t \mathbb{1}[x_t = 1](-c_t)$, where c_t is the score of the tth element to arrive. We can therefore express the regret and competitive ratio in this setting as follows:[6]

$$\text{Regret:} \qquad \sup_{F \in \mathcal{F}} \sup_{c} \left[\mathbb{E}[F(c, x)] - \inf_{x^* \in \mathcal{X}'} F(c, x^*) \right]$$

$$\text{Competitive ratio:} \qquad \sup_{F \in \mathcal{F}} \sup_{c} \frac{\mathbb{E}[F(c, x)]}{\inf_{x^* \in \mathcal{X}'} F(c, x^*)}$$

where the expectation is taken over randomness in the problem (e.g., the ordering of the elements in the secretary problem) and randomness in the algorithm. Thus, we see that the main difference between regret and the competitive ratio is that one is additive and one is multiplicative.

Of course, both of these definitions are minimized at the same point x^*. The difference between them is how they differentiate between suboptimal decisions. For example, sup-

[6]There are other variants of both regret and competitive ratio. For example, the cost function could be random, or the contexts could be random.

pose x is the decision sequence produced by a 2-competitive algorithm. Then for any F and any c, we have that $\mathbb{E}[F(c,x)] \leq 2\inf_{x^*\in\mathcal{X}'} F(c,x^*)$; further, suppose that for some F', c', this inequality is tight. The regret of this algorithm can therefore be bounded as follows:

$$\text{Regret} \geq \mathbb{E}[F'(c',x)] - \inf_{x^*\in\mathcal{X}'} F'(c',x^*) = \inf_{x^*\in\mathcal{X}'} F'(c',x^*).$$

For many reasonable cost functions F', this would show that $\text{Regret} = \Omega(T)$. For example, if $\mathcal{F}$ were composed of functions $F(c,x) = \sum_{t=1}^{T}(x_t - a_t)^2 + b_t$, where $b_t > \varepsilon > 0$ for all t, then this would imply linear regret. Hence, we have shown that an algorithm with a constant competitive ratio (typically considered the low end of competitive ratios) can incur linear regret (typically consider the high end of regret). Thus, depending on how good of an approximation is achievable, one of these measures may be more informative than the other. That said, some have analyzed their algorithms in terms of both regret and competitive ratio [42].

Historically speaking, the competitive ratio has been popular in online optimization as an online variant of the *performance ratio* used to describe approximation algorithms [43]. Many of the problems studied in this field are NP-hard optimization problems for which the best-possible algorithm is a constant factor approximation.

Regret, on the other hand, emerged in online learning settings [44], where the decisions are independent over time and might converge. When decisions are independent over time, the quality of a single decision x_t can be measured in terms of its *instantaneous regret* $h_t = f(x_t) - f(x^*)$, where $F(x) = \sum_t \big(f(x_t) - f(x^*)\big)$, and it is often of interest to decide whether (or how quickly) $h_t \to 0$. Conveniently, regret is simply the sum of these instantaneous regret terms, whereas the connection between h and the competitive ratio is more complicated.

As outlined above, either performance measure can be used in online decision-making problems, but some problems can be more appropriately or more easily analyzed in terms of one rather than the other.

2.3 Secretary algorithms and analysis

In Chapter 3, I discuss algorithm design for a variant of the secretary problem with *partial ordinal information*. That is, instead of observing an element's score $w(a_i)$ upon its arrival, the algorithm observes ordinal comparisons between a_i and the previously seen elements according to a partial order. As a warm-up, in this section, I discuss algorithms for the classical secretary problem (defined in Section 2.2.1), where each element reveals its score upon arrival, and at most one element can be selected.

In the 1960s, Dynkin gave an e-competitive algorithm which operates by observing the first $1/\mathrm{e}$ fraction of the elements, then selecting the next element with score higher than all previous elements [38].

Proposition 2.1 ([38]). *Assume that all scores are distinct.Dynkin's algorithm for the classical secretary problem is $\big(\mathrm{e} + o(1)\big)$-competitive as the number of element $N \to \infty$.*

Proof. Let a^* be the highest-score element. Then the competitive ratio is simply

$$1/\mathbb{P}(a^* \text{ is selected}).$$

Now note that if a^* arrives at time $i > N/\mathrm{e}$, then a^* is selected if and only if no previous elements were selected, which in turn happens if and only if the highest-weight element arriving before a^* was among the first $\lfloor N/\mathrm{e} \rfloor$. Since elements arrive in uniform random order, the probability that the highest-weight element arriving before a^* is among the first

$\lfloor N/\mathrm{e} \rfloor$ is $\lfloor N/\mathrm{e} \rfloor/(i-1)$. Thus we have that

$$\begin{aligned}
\mathbb{P}(a^* \text{ is selected}) &= \frac{1}{N} \sum_{i>N/\mathrm{e}} \mathbb{P}(a^* \text{ is selected} \mid a^* \text{ arrives at time } i) \\
&= \frac{1}{N} \sum_{i>N/\mathrm{e}} \frac{\lfloor N/\mathrm{e} \rfloor}{i-1} \\
&\geq \frac{1}{\mathrm{e}} \sum_{i>N/\mathrm{e}} \frac{1}{i-1} \\
&= \frac{1}{\mathrm{e}} \big(H(N-1) - H(\lfloor N/\mathrm{e} \rfloor) \big).
\end{aligned}$$

Now, to prove the claim, we must show that $H(N-1) - H(\lfloor N/\mathrm{e} \rfloor) \geq 1 + o(1)$. To that end, note that asymptotically in n, we have $H(n) = \ln(n) + \gamma + o(1)$, for some constant γ. Now let $\varepsilon_1 = H(N-1) - \ln(N-1) - \gamma$, $\varepsilon_2 = H(\lfloor N/\mathrm{e} \rfloor) - \ln(\lfloor N/\mathrm{e} \rfloor) - \gamma$, and $\varepsilon_3 = \ln\left(1 - \frac{1}{N}\right)$, and choose N large enough that

$$|\varepsilon_1|, |\varepsilon_2|, |\varepsilon_3| < \delta/3 \, .$$

For such an N, we can bound $H(N-1) - H(\lfloor N/\mathrm{e} \rfloor)$ as follows:

$$\begin{aligned}
H(N-1) - H(\lfloor N/\mathrm{e} \rfloor) &= \ln(N-1) + \gamma + \varepsilon_1 - \ln(\lfloor N/\mathrm{e} \rfloor) - \gamma - \varepsilon_2 \\
&= \ln\left(\frac{N-1}{\lfloor N/\mathrm{e} \rfloor}\right) + \varepsilon_1 - \varepsilon_2 \\
&\geq \ln\left(\frac{N-1}{N/\mathrm{e}}\right) + \varepsilon_1 - \varepsilon_2 \\
&= 1 + \varepsilon_1 - \varepsilon_2 + \varepsilon_3.
\end{aligned}$$

Thus, we have shown that $\mathbb{P}(a^* \text{ is selected}) \geq \frac{1}{\mathrm{e}} - o(1)$, which proves the claim. $\square$

The analysis of Dynkin's algorithm is slightly messing due to the rounding of the sample size, $\lfloor N/\mathrm{e} \rfloor$. If N were divisible by e, then the competitive ratio would be precisely e for all N. Generalizations of Dynkin's approach to the k-secretary problem (where at most

k elements can be selected) have an added complication. It is often useful in the analysis to consider the event where multiple specific elements are in the sample (e.g., among the first $\lfloor N/\mathrm{e} \rfloor$ elements for Dynkin's algorithm). However, when the sample size is fixed, the sampling of different elements is dependent: if a_1 is in the sample, then it is less likely that a different element a_2 is in the sample.

Both of these issues are resolved using *independent sampling* of elements. In particular, if one wishes to sample every element independently with probability p, this can be accomplished by simply drawing $M \sim \mathrm{Bin}(N, p)$ and sampling the first M elements (for a proof of this, I direct the reader to [45]). Using this approach, Soto adapted Dynkin's algorithm so that each element is sampled independently with probability $1/\mathrm{e}$; the analysis of Soto's algorithm is much less clunky and shows a competitive ratio of $1/\mathrm{e}$ [46].

One final algorithm which I would like to highlight is that of Buchbinder et al. [47], which was calculated using linear programming. In particular, suppose that an algorithm $\mathcal{A}$ for the classical secretary problem never selects an element if a previous element was better.

For such an algorithm, let p_i (for $i \in [N]$) be the probability of selecting the ith element to arrive. From the vector p, the authors show that the competitive ratio of the algorithm is $\frac{1}{N} \sum_{i=1}^{N} i p_i$ and the for all $i \in [N]$, the following is satisfied: $i \cdot p_i \leq 1 - \sum_{j=1}^{i-1} p_j$. Moreover, they show that any p satisfying these inequalities induces an algorithm with the same competitive ratio. Thus, an optimal classical secretary algorithm can be found by solving the following linear program:

$$\text{LP}(\mathcal{P}): \quad \begin{array}{ll} \text{maximize} & \frac{1}{N} \sum_{i=1}^{N} i \cdot p_i \\ \text{subject to} & i \cdot p_i \leq 1 - \sum_{j=1}^{i-1} p_j \text{ for all } i = 1, \ldots, N \\ & p \geq 0 \end{array}$$

I will explore this approach in the case of partial ordinal feedback in Section 3.4.

Algorithm 1: GRADIENT DESCENT [48]

input: initial point x_1, time horizon T, step size $\eta \in \mathbb{R}^T$

for $t = 1, \ldots, T$ **do**

    $x_{t+1} \leftarrow x_t - \eta_t \nabla f(x_t)$

2.4 Convex optimization

In this section, I will discuss the problem of minimizing a differentiable function f over $\mathbb{R}^N$. When f is convex, the standard approach is to use *gradient descent* [48], or some variant thereof. Gradient descent, presented below, is an iterative method which chooses the next point x_{t+1} by moving in the direction of the negative gradient at the current point x_t.

Note that if this algorithm were applied to a non-convex function, the iterates may converge to a local minimum which is not the global minimum, so convergence to the minimizer is not guaranteed. In general, the analysis of this algorithm (and its variants) depends on the assumptions made on f. In Chapter 4, I will discuss the setting where f is strongly convex and smooth, which I define next.

Definition 2.7 (α-strong convexity [48])**.** A function $f : \mathbb{R}^N \to \mathbb{R}$ is called α-strongly convex if for all $x, y \in \mathbb{R}^N$,

$$f(y) \geq f(x) + \nabla f(x)^\top (y - x) + \frac{\alpha}{2}\|y - x\|^2.$$

Definition 2.8 (β-smoothness [48])**.** A function $f : \mathbb{R}^N \to \mathbb{R}$ is called β-smooth if for all $x, y \in \mathbb{R}^N$,

$$f(y) \leq f(x) + \nabla f(x)^\top (y - x) + \frac{\beta}{2}\|y - x\|^2.$$

Essentially, strong convexity linearly lower bounds the rate of change of the gradient of f, and smoothness linearly upper bounds it. We can therefore think of functions which are smooth and strongly convex as "almost" quadratic function. For completeness, I prove this below.

Lemma 2.1. *Let $f : \mathbb{R}^N \to \mathbb{R}$ be a differentiable α-strongly convex and β-smooth function. Then for all $x, y \in \mathbb{R}^N$,*

$$\alpha\|y - x\|_2 \leq \|\nabla f(y) - \nabla f(x)\|_2 \leq \beta\|y - x\|_2.$$

Proof. The second inequality is a well-known fact about convex smooth functions. Now I show that the first inequality holds. Let $x, y \in \mathbb{R}^N$. If $x = y$, then the inequality hold. Otherwise, by strong convexity, we have the following inequalities:

$$f(y) \geq f(x) + \nabla f(x)^\top (y - x) + \frac{\alpha}{2}\|y - x\|^2$$
$$f(x) \geq f(y) + \nabla f(y)^\top (x - y) + \frac{\alpha}{2}\|x - y\|^2.$$

Adding them together, we get

$$\alpha\|y - x\|^2 \leq (\nabla f(y) - \nabla f(x))^\top (y - x) \leq \|\nabla f(y) - \nabla f(x)\|\|y - x\|$$

using the Cauchy-Schwarz inequality. □

Turning back to gradient descent, I provide a proof of convergence rate under smooth and strong convexity assumptions, courtesy of Elad Hazan.

Proposition 2.2 ([48]). *Let $f : \mathbb{R}^N \to \mathbb{R}$ be an α-strongly convex and β-smooth function. Then Algorithm 1, with step size $\eta_t = \frac{1}{\beta}$, attains a convergence rate of*

$$h_{t+1} \leq h_1 \mathrm{e}^{-(\alpha/\beta)t} ,$$

where $h_t = f(x_t) - \min_x f(x)$.

Proof [48]. First, observe that

$$\begin{aligned} h_{t+1} - h_t &= f(x_{t+1}) - f(x_t) \\ &\leq \nabla f(x_t)^\top (x_{t+1} - x_t) + \frac{\beta}{2}\|x_{t+1} - x_t\|^2 && \text{by smoothness} \\ &= -\frac{1}{\beta}\|\nabla f(x_t)\|^2 + \frac{1}{2\beta}\|\nabla f(x_t)\|^2 \\ &= -\frac{1}{2\beta}\|\nabla f(x_t)\|^2. \end{aligned}$$

Thus we have a decrease in h dependent on the current gradient. By strong convexity, however, we can show that the current gradient can be bounded from below by h. In particular, we know that by strong convexity, we have that

$$f(y) \geq \min_z \left\{ f(x) + \nabla f(x)^\top (z - x) + \frac{\alpha}{2}\|z - x\|^2 \right\}.$$

The expression being minimized on the right-hand side has gradient $\nabla f(x) - \alpha x + \alpha z$ and thus is minimized at $z = x - \frac{1}{\alpha}\nabla f(x)$. This gives us that

$$f(y) \geq f(x) + \nabla f(x) \left(-\frac{1}{\alpha}\nabla f(x) \right) + \frac{\alpha}{2} \left\| -\frac{1}{\alpha}\nabla f(x) \right\|^2 = f(x) - \frac{1}{2\alpha}\|\nabla f(x)\|^2.$$

If we set y to by $x^* := \arg\min_x f(x)$ and $x = x_t$, this reduces to

$$2\alpha h_t \leq \|\nabla f(x_t)\|^2.$$

Plugging this into our bound on $h_{t+1} - h_t$ from above, we get that

$$h_{t+1} - h_t \leq -\frac{1}{2\beta}\|\nabla f(x_t)\|^2 \leq -\frac{\alpha}{\beta} h_t.$$

In other words, we have proven a contraction: $h_{t+1} \leq (1 - \alpha/\beta) h_t$. The result follows. □

The above algorithm and analysis assume perfect first-order information. In other words, they assume that for each decision x_t, the algorithm observes $\nabla f(x_t)$. In Chapter 4, I will consider a bandit feedback setting, where the algorithm only receives noisy function value observations; in particular, for a decision x_t, the algorithm observes $f(x_t)+\varepsilon_t$, where $(\varepsilon_1, \ldots, \varepsilon_T)$ is a sequence of mean-zero, sub-Gaussian, independent noise.

In this bandit setting, it will be useful to characterize the sample complexity for obtaining function value estimates and gradient estimates. Hoeffding's inequality will be useful for both.

Lemma 2.2 (general Hoeffding's inequality [49])**.** *Let $\varepsilon_1, \ldots, \varepsilon_n$ be independent mean-zero sub-Gaussian random variables. Then, there is a constant C such that for every $s \geq 0$, we have*

$$\mathbb{P}\Big(\Big|\sum_{t=1}^{n} \varepsilon_i\Big| \geq s\Big) \leq 2\exp\Big(-\frac{Cs^2}{\sum_{t=1}^{n} \|\varepsilon_i\|_{\psi_2}^2}\Big).$$

Using Hoeffding's inequality, we can bound the accuracy of a derivative estimate obtained from samples at two points, $x, y \in \mathbb{R}$.

Lemma 2.3 (sandwich lemma)**.** *Let $f : \mathbf{R} \to \mathbf{R}$ be an α-strongly convex and β-smooth function, and let $x < y$. Suppose that querying f at any point z yields a noisy observation $f(z) + \varepsilon_t$ for the t^{th} sample, and suppose that the noise variables $\varepsilon_1, \varepsilon_2, \ldots$ are independent, mean-0, and have sub-Gaussian norm at most $E_{\max}$. Now fix $p \in (0,1)$, and let $\overline{f}(x)$, $\overline{f}(y)$ be the averages of $\frac{64E_{\max}^2 \log \frac{2}{p}}{C\alpha^2(y-x)^4}$ samples at x and y, respectively, where C is the absolute constant from Lemma 2.2. Then the estimated secant $g = \frac{\overline{f}(y)-\overline{f}(x)+\frac{\alpha(y-x)^2}{4}}{y-x}$ satisfies $\nabla f(x) \leq \frac{f(y)-f(x)}{y-x} \leq g \leq \nabla f(y)$, with probability at least $(1-p)^2$.*

Proof. Let $\varepsilon = \alpha(y-x)^2/4$. Then by Hoeffding's inequality (Lemma 2.2) with $n = \frac{64E_{\max}^2 \log \frac{2}{p}}{C\alpha^2(y-x)^4}$,

$$\mathbb{P}\Big(|\overline{f}(x) - f(x)| \geq \varepsilon/2\Big) \leq 2\exp\Big(-\frac{C(\varepsilon/2)^2 n}{E_{\max}^2}\Big) = p,$$

and similarly for y. So, we have that

$$\big|\overline{f}(x) - f(x)\big| \leq \frac{\varepsilon}{2} \quad \text{and} \quad \big|\overline{f}(y) - f(y)\big| \leq \frac{\varepsilon}{2} \tag{2.1}$$

with probability at least $(1-p)^2$. For the remainder of the proof, we assume that (2.1) holds. Next, we bound $g - \frac{f(y)-f(x)}{y-x}$ from above and below:

$$\begin{aligned} g - \frac{f(y)-f(x)}{y-x} &= \frac{\overline{f}(y) - \overline{f}(x) + \varepsilon}{y-x} - \frac{f(y)-f(x)}{y-x} && \text{(by def'n of } g\text{)} \\ &= \frac{1}{y-x}\Big[\big(\overline{f}(y) - f(y)\big] + \big(f(x) - \overline{f}(x)\big) + \varepsilon\Big] \\ &\in \left[0, \frac{2\varepsilon}{y-x}\right]. \end{aligned}$$

By definition of ε, this gives us:

$$0 \leq g - \frac{f(y)-f(x)}{y-x} \leq \frac{\alpha}{2}(y-x) \qquad \text{with probability at least } (1-p)^2. \tag{2.2}$$

Since we have just shown that g is close to the secant $\frac{f(y)-f(x)}{y-x}$, the only thing remaining is to show that $\nabla f(y)$ is sufficiently far away from the secant. We can do this by using strong convexity:

$$\frac{f(y)-f(x)}{y-x} \leq \frac{\nabla f(y)(y-x) - \frac{\alpha}{2}(y-x)^2}{y-x} = \nabla f(y) - \frac{\alpha}{2}(y-x).$$

We've thus shown that $0 \leq g - \frac{f(y)-f(x)}{y-x} \leq \frac{\alpha}{2}(y-x) \leq \nabla f(y) - \frac{f(y)-f(x)}{y-x}$. It follows that the estimated gradient g satisfies $\nabla f(x) \leq \frac{f(y)-f(x)}{y-x} \leq g \leq \nabla f(y)$, where the first inequality follows from convexity, thus proving the lemma. □

2.5 Partial orders

As discussed in Section 2.1.3, differences in error variance across groups can result in differing outcomes across groups. Accounting for error variance can thus be motivated as a group fairness intervention. Additionally, accounting for error variance forces the decision-maker to reckon with the accuracy of their selection criteria.

Suppose, now, that an evaluator outputs a numerical score and a confidence interval for each input. If we have two inputs with intervals $[1, 3]$ and $[4, 6]$, and we are tasked with selecting one of the inputs, we would likely choose the latter. On the other hand, if the intervals were $[1, 3]$ and $[2, 4]$, the correct action is less clear. In this case, there is a substantial amount of uncertainty in the relative ranking of the two inputs. How, then, should the decision-maker act?

One option for handling this uncertainty in relative rankings, which I propose in Chapter 3, is to make decisions based *only* on relative rankings which are known with sufficient certainty. This can be done by representing the uncertainty in a *partial order* [50] (defined next) and making decisions based on this partial order.

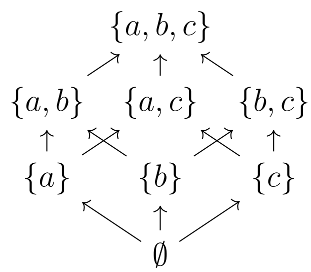
Definition 2.9 (partially ordered set)**.** Let S be a set and $\preceq$ a binary relation on S. The pair $\mathcal{P} = (S, \preceq)$ is a *partially ordered set*, or *poset*, if the following conditions hold:

1. **Reflexivity:** for all $a \in S$, $a \preceq a$;
2. **Antisymmetry:** for all $a, b \in S$, if $a \preceq b$ and $b \preceq a$, then $a = b$; and
3. **Transitivity:** for all $a, b, c \in S$, if $a \preceq b$ and $b \preceq c$, then $a \preceq c$.

In this case, $\preceq$ is called a *partial order* on S.

For example, the subset relation $\subseteq$ is a partial order on any collection of sets. Similarly, the divisibility relation $|$ is a partial order on any subset of positive integers.[7] A poset $\mathcal{P} = (S, \preceq)$ is often pictorially represented with a "Hasse diagram," which is a drawing of a directed $D = (S, A)$, where $ab \in A$ if and only if $a \neq b$, $a \preceq b$, and there is no $c \in S \setminus \{a, b\}$ for which $a \preceq c$ and $c \preceq b$ [51]. In other words, the arcs of D represent all relative rankings which are not implied by reflexivity or transitivity. Typically, Hasse diagrams are drawn so that all arcs are oriented upward in the plane. For example, below is a Hasse diagram for the subset relation on the power set of $\{a, b, c\}$:

[7]For any integers a, b with $a \neq 0$, we say that "a divides b," and write $a \mid b$, if $b/a \in \mathbb{Z}$.



Note that $x \preceq y$ in a poset if and only if there is a directed path from x to y in the Hasse diagram. If $x \npreceq y$ and $y \npreceq x$, then x and y are called *incomparable*. For example, in the Hasse diagram above, the elements $\{a, b\}$ and $\{c\}$ are incomparable, since neither is a subset of the other.

Now suppose that a given partial order accurately compares inputs; that is, if $a \neq b$ and $a \preceq b$, then b is better than a. In this case, the more information carried in the poset, the better decisions one can make. One way to quantify the informativeness of a partial order is with its *width*.

Definition 2.10 (width [51])**.** The *width* of a partial order $\mathcal{P}$, denoted $\omega(\mathcal{P})$, is the maximum number of mutually incomparable elements.

For example, the width of the power set of $\{a, b, c\}$ is 3.

If all elements of a poset were mutually comparable, then decisions can be made using traditional techniques (e.g., thresholding in the case of a classification task). A subset with this property is called a *chain*.

Definition 2.11 (chain [51])**.** Let $\mathcal{P} = (S, \preceq)$ be a poset. A subset $C \subseteq S$ is called a *chain* if for all $a, b \in C$, we have that $a \preceq b$ or $b \preceq a$.

While not every poset is a chain, every poset can be partitioned into chains. Trivially, given a poset $\mathcal{P} = (S, \preceq)$, one can partition S into singletons, and each singleton is a chain. This, however, is not useful, since all ordinal information is lost when considering each singleton separately. Partitioning a poset into the minimum number of chains, however, can be useful in some settings. It is well-known that for any poset of width ω, there exists a partition into ω chains.

Proposition 2.3 (Dilworth [52])**.** *Let* $\mathcal{P} = (S, \preceq)$ *be a poset. There exists a partition* $(C_1, \ldots, C_{\omega(\mathcal{P})})$ *of* S *such that each* C_i *is a chain.*

This result will be useful in the analysis in Chapter 3.

CHAPTER 3
PARTIAL ORDERS AND UNCERTAINTY

This chapter contains excerpts from [53] and includes joint work with Swati Gupta.

With the rise of big data and the proliferation of machine learning in important decision-making processes, automated résumé-screening and selection algorithms have become commonplace [54, 25]. Hiring decisions are often made based on numerical evaluations of applicants, such as grades, standardized test scores, number of publications, ratings by selection committees, etc., and companies have emerged to assist with automating these decisions (e.g., Indeed, [5]). Although scoring applicants provides a numeric scale on which to compare them, it is often unclear what the impact of their training or their socio-economic background is on test scores. Studies have shown that experiences with racial and gender stereotyping can have adverse effects on test performance (see, e.g., [55, 56]). Further, economic status differences and the differences in perceived rewards and costs of the tests can also produce differences in ability test scores between high-income and low-income individuals [57]. Even if performance is equal, applicants may be penalized due to implicit bias of evaluators [58], and simply scrubbing protected information can be ineffective in mitigating such biases [59, 7]. This further becomes a problem in practice as automated screening methods that use real-world data pick up on biased trends in the underlying data. For example, Amazon experimented with such a résumé-filtering algorithm, but scrapped the project when evidence of algorithmic sexism[1] emerged [60].

There is growing literature to model and address such bias-related issues behaviorally [61], mathematically [31, 7, 62], and by studying societal systems [63, 64]. However, little is known about the impact of bias and the effectiveness of these measures when dealing

[1]This might have been due to many reasons; e.g., the data that was fed into Amazon's algorithm might have already incorporated existing biases of the human hiring committees.

with implicit bias in data. Kleinberg and Raghavan were the first to show that in offline selection, the Rooney Rule can in fact increase the scores of selected applicants (under certain parameters of their model). Our attempt is to *provide a data-driven model of bias and to answer similar questions for online selection*.

We model bias in evaluations using partially ordered sets as follows: applicants whose relative ranking is uncertain due to biases or inaccuracies are considered *incomparable*. Given a batch of applicants and a partial order (encoding uncertainties or biases) over them, it is easy to make offline selection decisions by (randomly) selecting the highly ranked applicants in the poset. However, given the competitive nature of the market and the high volume of applicants, there is an online aspect to selection processes in hiring.

In this chapter, apart from proposing partially ordered sets to counteract bias in evaluations data, we develop secretary algorithms that are competitive given partial ordinal information. Our work is the first to consider this adaptation of the secretary problem for multiple selections. After discussing algorithm design and analysis, I will provide experiments results and discuss real-world impact in applicant-screening.

3.1 Background and Main Results

In this section, I first discuss related work in secretary problems and bias mitigation, drawing connections between the fairness notion proposed in this chapter and existing fairness notions. Next, I outline the contributions of this chapter.

3.1.1 Related work

Secretary Problems: Our work adds to the expansive literature on secretary problems by relaxing the assumption that the algorithm has access to true scores of applicants. The classical secretary problem is to select in online fashion at most one element from a randomly ordered scored set of known size. The classical algorithm for the problem sets a threshold based on a sample of the elements and selects the first non-sampled element exceeding the threshold [38].

We consider the k-secretary problem, where up to k elements can be selected. There are two techniques which commonly appear in the secretary literature to handle multiple selections: *adaptive thresholding*, in which the threshold is periodically updated, and *random partitioning*, in which elements are randomly partitioned into multiple sets, and selection capacity is distributed across the sets. Our two main algorithms, ADATHRESHOLD and POSETLABEL, use these techniques, respectively. Adaptive thresholding was used by [65] to generalize the classical secretary algorithm while maintaining its e-competitiveness and was notably used by [4] to obtain a tight (as $k \to \infty$) competitive ratio of $\left(1 - \frac{5}{\sqrt{k}}\right)^{-1}$. The random partitioning technique has been used in matroid secretary problems [66, 46], which we discuss below.

Many other interesting constraints have been studied in the secretary literature, such as knapsack constraints, under which each element has both a score and a weight, and the total weight of selected elements cannot exceed some number [65]. In another line of work, the elements are assumed to form the ground set of some matroid, and selected elements are required to form an independent set. While an $\mathcal{O}(\log \log \mathrm{rank})$-competitive algorithm was given in [45], it remains open whether this problem admits a constant-competitive algorithm. When scores are randomly assigned to the elements, however, one can exploit the principal partition of the matroid to achieve a constant competitive ratio [46]. These problems all assume access to the true rankings of elements, as opposed to our problem, which assumes only a *partial* ranking.

Several limited-information variants of the secretary problem have emerged. *Ordinal secretary problems* assume access to total rankings of elements, but not numerical scores [67]. Some variants limit information even further, allowing for comparisons between only select pairs of applicants. Among such problems is the Q-queue, J-Choice, K-best secretary problem, which assumes applicants are randomly split into Q groups of equal size, and comparisons cannot be made between applicants of different groups. This differs from our setting, in which the partial order is deterministic and not restricted to be a union of disjoint chains.

More generally, some work has been done on secretary problems with partial ordinal information on rankings, with the objective of picking any maximal element [68]. There are three differences between the work of Kumar et al. and ours: (1) we allow for multiple selections, whereas their work allows for only one selection. Their techniques do not easily extend to our multiple-choice setting, nor does their lower bound; (2) they assume prior knowledge of the number of elements *and* the number of maximal elements, whereas we only assume access to the number of elements; and (3) their objective is to select any maximal element, whereas ours is to maximize the total score of selections according to some unknown scoring that is consistent with the ranking. Our objective is harder in the sense that routinely selecting the worst (by score) of the maximal elements will yield good performance by their objective and bad performance by ours. We consider the first two differences to be significant, and the third to be minor.

Fairness in Hiring and Selection Problems: Our work adds to the literature on selection under biased information. The works of [69] and [70] consider the scenario where candidates come from various demographic groups, and their observed scores are scaled by a constant dependent on their group membership. In contrast, the work of [71] assumes observed scores are unbiased estimates of true scores with group-dependent noise variance. In both cases, the assumed bias corrupts certain comparisons between candidates, and thus a partial order can be constructed to represent the bias. In this way, our methods provide an alternative approach to the quota-based methods discussed in the above papers. The other main differences between our work and theirs are that (1) our problem is online, and (2) we make no assumption on the underlying bias except that it can be encoded in a partial order.

In terms of fairness constraints, ordinal fairness (OF) shares some characteristics with individual fairness [7] and conditional demographic parity. Individual fairness (IF) requires that people who are similar (according to some metric) receive similar decisions; in this case, treatment depends heavily on the underlying metric. Analogously, under OF, individuals who are similar (according to some partial ranking) must receive similar decisions; thus, treatment depends on the underlying partial order. In contrast, however, the purpose of IF is to avoid large disparities in treatment of similar individuals, whereas the purpose of OF is to account for uncertainty in evaluations. One advantage of OF over individual fairness is its basis in the law: as discussed in Chapter 5, U.S. caselaw supports the use of partial rankings in the hiring process; it is unclear, however, how the sensitivity of decisions to a similarity metric for individual fairness is perceived from a legal perspective (e.g., what happens if two seemingly valid metrics produce different outcomes?).

Conditional demographic parity (CDP),[2] on the other hand, equalizes treatment conditional on some (presumably predictive) feature. As we will discuss in Section 3.2.2, OF implies CDP, where the feature on which we condition is isomorphism class (CDP, however, does not imply OF). It is worth noting that OF does not imply demographic parity nor any outcomes-based measure, even when the partial order is derived from group bias (cf. Example 3.1).

In terms of accounting for uncertainty, the closest notion of fairness to ours is the "meritocratic fairness" (MF) notion proposed by [24] in the multi-armed bandits context. Under MF, a confidence interval is maintained for each arm, and arms with overlapping confidence intervals must be treated equally. For example, given confidence intervals of $[1, 4]$, $[3, 6]$, and $[5, 8]$, their constraint would force equal treatment of the three applicants, despite the third being confidently better than the first. In contrast, ordinal fairness with respect to the induced interval order would allow for better treatment of the third arm in this case. The other significant difference between MF and ordinal fairness is that the partial order in ordinal fairness need not be derived from intervals, thus allowing for a broader notion of uncertainty (cf. Section 3.2.1).

Work has been done on long-term effects of bias-aware algorithms as well. While our

[2]CDP was defined in Section 2.1.

work focuses on single-round hiring and mitigating effects of biased evaluations, one may hope that such interventions will lead to long-term reduction of bias in evaluations. Other work has studied long-term effects of affirmative action and conditions under which certain policies can effectively stabilize the system, resulting in an equilibrium which selects a proportional number of candidates from each demographic group [72]. In a similar vein of research, Coate and Loury have studied the interplay of affirmative action and employer biases, and the conditions under which a temporary application of affirmative action can lead to long-term benefits for disenfranchised groups [73]. Our work complements these by providing a mathematical basis for selecting candidates fairly in spite of bias in evaluations.

3.1.2 Results

1. **Poset Model and Fairness.** We propose the *poset model of bias*, which allows the algorithm access to a partial order (encoding uncertainties and biases) on the set of applicants consistent with the total order induced by true scores (Sec. 3.2.1). This is the first paper to use partial orders as part of a bias mitigation strategy. We show how this model generalizes several natural notions of bias such as the group model of bias [69]. Next, we propose *ordinal fairness* (OF) which expands on the notion of conditional demographic parity, incorporating the partial ranking of applicants and enforcing monotonicity in selection rates with increasing rank. We propose the poset secretary problem, where the goal is to select k applicants with the highest *true* total score from a pool of N, while the algorithm has access only to an a priori unknown partial order on the applications. Our proposed algorithms for the poset secretary problem all satisfy ordinal fairness.

2. **A Lower Bound.** In Section 3.3, I show that any algorithm for the poset secretary problem incurs a competitive ratio of at least ω, where ω is the width[3] of the poset.

3. **Poset Secretary Algorithms via Linear Programming.** In Section 3.4, I discuss

[3]The width of a poset is the maximum number of mutually incomparable elements.

how algorithms for selecting one element from a fixed (known) poset can be generated using linear programming. This is done by formulating a poset-specific linear program and showing that the feasible points of this LP are in one-to-one correspondence with 1-secretary algorithms. This does not, however, answer solve the poset secretary problem, as it assumes advance knowledge of the poset and only handles $k = 1$ selection.

4. **Poset Secretary Algorithms for Small k.** We discuss challenges to developing poset secretary algorithms in Section 3.5.1, and we provide an algorithm based on the random partitioning technique from the matroid secretary literature [66] in Section 3.5.2 which requires an estimate of the width as input. In Section 3.5.3, we show how this assumption on a width estimate can be dropped while maintaining order optimality. The main challenge to this approach is learning how "informative" the partial ranking is, which is important for deciding how extensively to sample. We address this by estimating the width of the poset from an initial sample.

5. **An Asymptotically Tight Poset Secretary Algorithm.** Though the above algorithms are order-optimal and implementable in practice due to their explainability, their competitive ratios do not diminish as the number of selections increases, contrary to results for the vanilla secretary problem [4]. To address this, in Section 3.6, we provide a novel notion of thresholding in partially ordered sets with the property that applicants with relatively high potential (according to the partial order) meet the threshold. Our algorithm periodically updates the threshold, thus extending Kleinberg's adaptive thresholding technique to the partial ordinal setting [4]. The resulting algorithm achieves a tight competitive ratio of $\omega \cdot (1 + o(1))$ as $k \to \infty$ under the regime where $(\log N)/\sqrt{k} \to 0$.

6. **Algorithms for the Special Case of Group Bias.** A special case of the poset bias model is the group bias model, which assumes that applicants belong to g disjoint

groups, and bias is consistent within groups. This lends a total order on applicants within the same group, and no comparisons on applicants across different groups. In Sec. 3.7, we simplify our poset algorithms to this setting, resulting in an $\mathcal{O}(g)$ competitive ratio, and we obtain an improved competitive ratio of $\mathcal{O}(1)$ when the group membership and scores are assigned randomly (see Table 3.1). Moreover, we show that typical vanilla secretary algorithms can be parallelized across the groups in a way that increases their competitive ratio by only a factor of $\mathcal{O}(g)$. This gives a recipe to convert any algorithm in the vanilla secretary setting to the group-bias setting.

7. **Computational Experiments.** In Section 3.8, we test our algorithms on the AMEO 2015 dataset,[4] which includes gender, GPA, college tier, and computer programming test scores for college graduates, among other features. We train a linear regression model to predict computer programming scores for three different shifted distributions of scores, and use gender-specific error distributions in the resulting predictions to construct partial orders over applicants. We compare selection rates of four algorithms across gender in the secretary setting and observe that enforcing uniform quotas is insensitive to distributional changes in the data, whereas the poset-based algorithms follow the distribution of scores in the data better. Conversely, poset-based methods are less sensitive to distributional changes in data than vanilla algorithms, since the former give applicants more benefit of the doubt by design.

3.2 Problem Formulation: Bias, Fairness, and the Secretary Problem

In Section 3.2.1, we introduce the poset model of bias as a way to represent uncertainties in evaluations and provide examples showing how the model can be used to account for multiple kinds of bias or uncertainty. In Section 3.2.2, we introduce ordinal fairness as

[4] The dataset is available at http://research.aspiringminds.com/resources/#datasets.

	Competitive ratio	Algorithm
Poset model		
ω known	$(\omega+1)\frac{\mathrm{e}^2}{\mathrm{e}-1}$ (Corollary 3.1)	PROXYLABEL (Algorithm 2)
ω unknown	$\frac{\mathrm{e}^3}{\mathrm{e}-1}(4\omega+2)\big(1+o(1)\big)$ (Proposition 3.6)	POSETLABEL (Algorithm 3)
ω unknown $\omega \le \log k$	$\omega\Big(1-\frac{38\log N}{\sqrt{k}}\Big)^{-1}$ (Corollary 3.2)	ADATHRESHOLD (Algorithm 4)
Group model		
Adversarial	$(g+1)\frac{\mathrm{e}^2}{\mathrm{e}-1}$ (Corollary 3.1)	PROXYLABEL (Algorithm 2)
Adversarial	$gf(k/g)$ (Proposition 3.9)	GAP (Algorithm 5)
Stochastic	$2\mathrm{e}\big(1+o(1)\big)$ (Proposition 3.10)	GAPCAP (Algorithm 6)

Table 3.1: Summary of poset secretary algorithms for selecting k out of N applicants. In the poset model of bias, the algorithm has access to partial ordinal rankings of applicants, and the competitive ratio is given in terms of the width ω of the poset. In the group model of bias, the number of groups g is known to the algorithm; in the stochastic setting, both group membership and scores are random. For GAP, the algorithm being parallelized has competitive ratio $f(k)$ for k selections. All of these algorithms satisfy ordinal fairness.

a desirable quality of selection under poset bias. Finally, in Section 3.2.3, we formally introduce the poset secretary problem.

3.2.1 Modeling Bias

We model uncertainty in evaluations using a partial order on the set $\mathcal{C}$ of elements,[5] in which pairs of elements that cannot be ranked against each other with high enough certainty are deemed incomparable. Each element $a \in \mathcal{C}$ has a true (inaccessible) score $w(a) \ge 0$ signifying their ability or utility to the selection algorithm. We assume that these true scores are distinct, so that they induce a total order $(\mathcal{C}, \preceq_t)$ on the elements, which is not known to the algorithm; rather, the set of such scores will be used to compute the worst-case performance benchmark. The partial order $\mathcal{P} = (\mathcal{C}, \preceq_p)$ which constitutes the feedback

[5] For the sake of congruency with the secretary literature, we henceforth refer to applicants as elements.

must be consistent with the total order; i.e.,

$$a \prec_p b \implies a \prec_t b \iff w(a) < w(b).$$

We next provide examples illustrating how uncertainties can be translated to partial orders.

Example 3.1 (Group bias). A special case of our model is the *group bias* model of [69], which assumes that elements belong to disjoint[6] demographic groups $G_1, \ldots, G_g$, and each group G_i has an associated bias factor $\beta_i \geq 1$. The observed (biased) score of an element $a \in G_i$ is $\widetilde{w}(a) = w(a)/\beta_i$. This induces a natural partial order where for any pair of elements $a_1, a_2 \in G_i$, $a_1 \prec a_2$ if $\widetilde{w}(a_1) < \widetilde{w}(a_2)$. Pairs of elements in different groups are considered incomparable.

Example 3.2 (Discrete evaluations). Suppose each element a has an attribute $x_a \in \{1, \ldots, 10\}$, and that this attribute is unbiased and predictive of utility. Then we can define a partial order where $a \prec b$ if and only if $x_a < x_b$. Note that elements with the same attribute value are incomparable.

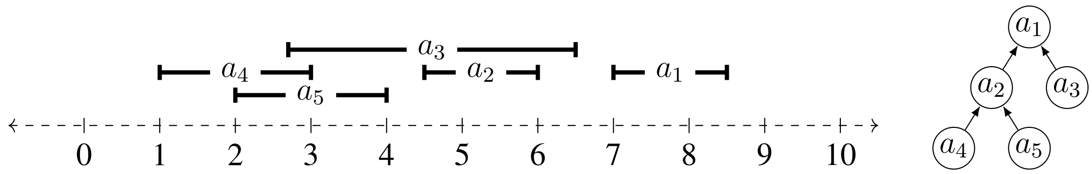
Example 3.3 (Multiple evaluations). Suppose five elements, $a_1, \ldots, a_5$, are evaluated by three metrics or by three committee members. The scores assigned to the elements, respectively, are

$$\begin{bmatrix} 7 \\ 8 \\ 8.5 \end{bmatrix}, \quad \begin{bmatrix} 4.5 \\ 5.5 \\ 6 \end{bmatrix}, \quad \begin{bmatrix} 2.7 \\ 4 \\ 6.5 \end{bmatrix}, \quad \begin{bmatrix} 1 \\ 3 \\ 3 \end{bmatrix}, \quad \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix}.$$

Since the scores of a_1 are between 7 and 8.5, it may seem likely that the "true" score of a_1 is in the interval $[7, 8.5]$. In this way, these scores naturally induce the following intervals and rankings:[7]

[6] If, for example, the groups G_1 and G_2 intersect nontrivially, then we can simply consider the disjoint groups $G_1 \setminus G_2, G_2 \setminus G_1$, and $G_1 \cap G_2$. This approach non-ideal if there are many demographics being considered, as the number of intersectional groups is exponential in the number of demographics.

[7] Suppose one wished to select three elements from this poset. Certainly a_1 should be selected, since it is

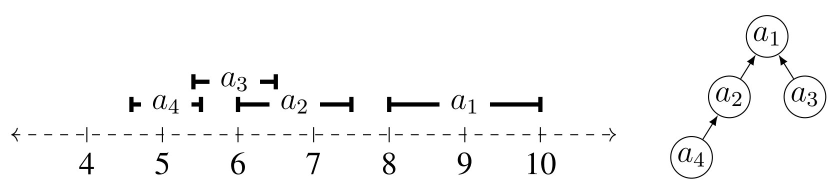


Note that the poset derived in Example 3.3 was constructed from a set of intervals, where comparable elements corresponded to disjoint intervals. Partial orders constructed in this manner constitute an important class called *interval orders*:

Definition 3.1 (interval order)**.** An *interval order* is a set $\{a_1, \ldots, a_N\}$ along with a binary relation $\prec$ that satisfies the following condition: there exists a correspondence between each element a_i and an interval $[\ell_{a_i}, u_{a_i}] \subset \mathbb{R}$, such that $a_i \prec a_j$ if and only if $u_{a_i} < \ell_{a_j}$.

Another way that interval orders can arise is through estimations of bias factors:

Example 3.4 (Bias factor-derived intervals). Suppose the observed score of element a is $\widetilde{w}(a) = w(a)/\beta_a$ and that $a_1, a_2 \in G_1$ and $a_3, a_4 \in G_2$. Suppose we know that for any $a \in G_1$, $\beta_a \in [1, 1.25]$, and for any $a \in G_2$, $\beta_a \in [1.25, 1.5]$. If the algorithm observes scores $\widetilde{w}(a_1) = 8$, $\widetilde{w}(a_2) = 6$, $\widetilde{w}(a_3) = 4.33$, and $\widetilde{w}(a_4) = 3.67$, then it can be inferred that $w(a_1) \in [8, 10]$, $w(a_2) \in [6, 7.5]$, $w(a_3) \in [5.41, 6.5]$, $w(a_4) \in [4.59, 5.51]$, resulting in the following intervals and rankings:



Notice that adopting the group model of bias here would result in a possibly erroneous ranking of a_3 and a_4, as the group model does not allow for variation in bias factors within groups.

the maximum element. Selection of the other two elements should involve some amount of hedging, since it is unclear which elements are the top three. That said, one might deterministically select a_2 as well, since a_2 is among the top three elements in every linear extension of the poset, in which case only the third selection would involve hedging.

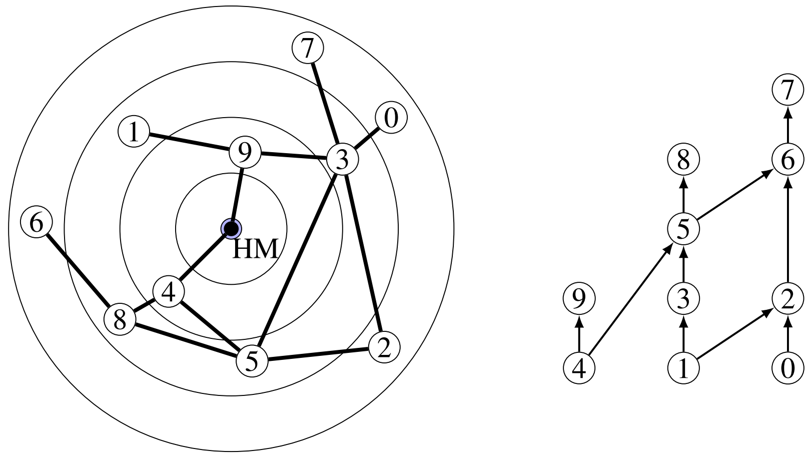


Figure 3.1: Example of a partial ranking derived from an instance of network bias.

Example 3.5 (network bias). To account for bias stemming from proximity to the employer, we propose the *network bias* model, in which the hiring manager (HM) and elements (applicants) are vertices in a graph (see Figure 3.1). The bias that element a experiences depends on the distance[8] $d(a, \text{HM})$ between the element and the hiring manager in the graph: the greater the distance, the greater the bias experienced. For example, the observed score of an element a can be $\widetilde{w}(a) = w(a)/\beta(d(a, \text{HM}))$, where $\beta(\cdot)$ is an increasing function of the distance $d(a, \text{HM})$. In this case, elements at the same distance from HM are comparable (i.e., they form a chain).

Note that the class of posets derived from such networks is different from the class of posets derived from group bias (cf. Example 3.1). The latter consists of posets that are *disjoint* unions of chains; the former, however, admits some additional comparisons between these chains. Network bias posets and group bias posets are further not subsumed by the class of interval orders.[9]

We next discuss fairness in the context of selection from a poset.

[8]i.e., the length of the shortest path.

[9]For example, the poset $\{a, b, c, d\}$ with relations $a \prec b$ and $c \prec d$ (only) cannot be induced by intervals, but can be induced by an instance of network bias or an instance of group bias.

3.2.2 Fairness under Uncertainty

Equal opportunity for equally qualified applicants is a natural goal to strive for and is often formalized as *conditional demographic parity* (CDP): given a predictive attribute A, group membership G, and binary decision X, $\mathbb{P}(X = 1 \mid A = a, G = g) = \mathbb{P}(X = 1 \mid A = a)$ [31].

If one had access to an unbiased and predictive attribute of applicants (e.g., their true scores), then it might make sense to impose CDP: two applicants with the same attribute value must have the same probability of selection. This fairness constraint ensures that irrelevant attributes are not informing or biasing decisions. One drawback of conditional demographic parity is its reliance on an unbiased attribute, which may or may not be available to the decision-maker.

Under the poset model of bias, one can analogously try to treat equally *ranked* or *positioned* applicants *in the poset* equally, in which case conditional demographic parity would be satisfied with respect to their *positions* in the poset. For example, consider a partial order over applicants $\{a, b, c\}$ in which the only comparisons are $b \prec a$ and $c \prec a$ (so, b and c are incomparable). Then, the probability of selection of b should be the same as that of c. In this way, given a partial ranking of applicants, we can apply this poset-variant of conditional demographic parity as an approximation of equal opportunity. To formalize this, we must first define what it means for two elements to be "equally ranked" in a poset. We do so using the notion of an order isomorphism.

Definition 3.2 (order isomorphism)**.** Let $\mathcal{P}_1 = (X, \prec_1)$ and $\mathcal{P}_2 = (Y, \prec_2)$ be posets. A bijection $\varphi : X \to Y$ is an *order isomorphism* if for all $x_1, x_2 \in X$, we have $x_1 \preceq_1 x_2 \iff \varphi(x_1) \preceq_2 \varphi(x_2)$.

Any two order-isomorphic posets are structurally the same—that is, they have the same number of elements, the same number of minimal and maximal elements, the same set of totally ordered subsets, the same width, etc. An order isomorphism captures this equivalence by mapping minimal elements to minimal elements, totally ordered subsets to totally ordered subsets, and so on. So, if element a can be mapped to element b under an order isomorphism, then the two elements can be considered equally ranked. We can now define ordinal fairness, which is our interpretation of equal opportunity under partial ordinal information.

Definition 3.3 (ordinal fairness)**.** We say that a selection algorithm satisfies *ordinal fairness* (OF) with respect to a partial order if the following conditions hold:

1. **(order parity)** for any order isomorphism φ and any element a, $\mathbb{P}(a \text{ is selected}) = \mathbb{P}(\varphi(a) \text{ is selected})$; and
2. **(monotonicity)** whenever $a \prec b$, $\mathbb{P}(a \text{ is selected}) \leq \mathbb{P}(b \text{ is selected})$.

The first condition is an analog of conditional demographic parity in the partial ordinal setting; in fact, when it is applied to the poset in Example 3.2, we recover the classical notion of conditional demographic parity, as explained below in Example 3.6; more generally, order parity enforces CDP with respect to the isomorphism classes of the poset.[10]

Example 3.6. In this example, I formalize the relationship between ordinal fairness and conditional demographic parity. Suppose there is a predictive attribute A which takes values in $\{1, 2, 3, 4, 5\}$, and you would like to make fair decisions with respect to A across demographic groups. Conditional demographic parity would require that $\mathbb{P}(X = 1 \mid A = a, G = g) = \mathbb{P}(X = 1 \mid A = a)$ for any g, where X is the binary decision and G is the group membership.

One can construct a partial order on the elements by defining $a_i \prec a_j$ if and only their A values satisfy $A(a_i) < A(a_j)$. See Figure 3.2 for an example of this construction on eight elements. Note that under this construction, any two elements with the same A value are order isomorphic. Thus under ordinal fairness, any two elements in the same

[10]Note that the act of *banding* of U.S. caselaw discussed in Chapter 5 allows for (but does not require) incomparable elements to be treated equally. Ordinal fairness similarly does not require all incomparable elements to be treated equally, only those which are order isomorphic.

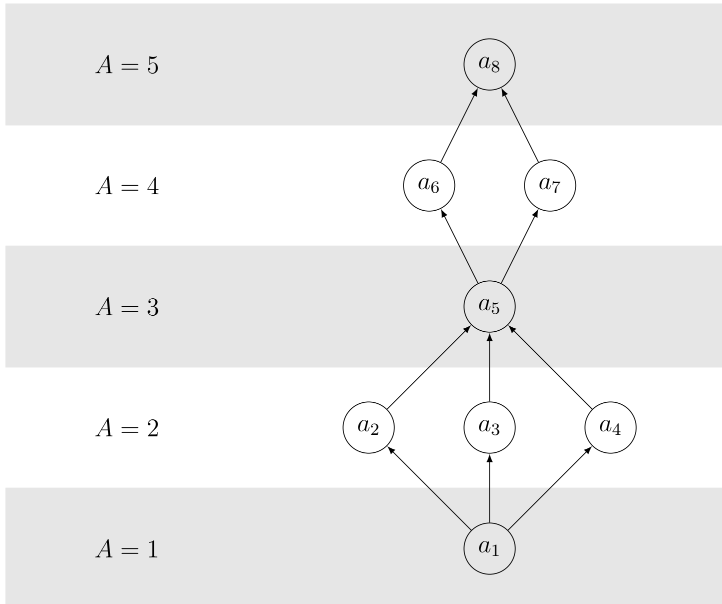


Figure 3.2: Partial ranking of eight elements according to the discrete attribute A. Elements in the same horizontal band have the same attribute value and are order isomorphic.

horizontal band in Figure 3.2 must be treated equally. Ordinal fairness thus directly implies conditional demographic parity for this construction, as elements with the same A value must be treated equally.

The second condition, monotonicity, requires that decisions be more favorable to higher ranking elements. As with CDP, OF does not impose group-specific quotas. Indeed, if all members of group G_1 are ranked higher than all members of G_2 in the poset, an algorithm which exclusively selects from G_1 may satisfy OF despite violating demographic parity.

The fairness of OF comes from the construction of the partial ranking, just as the fairness of individual fairness comes from the construction of the metric. OF (resp. individual fairness) simply ensures adherence to the partial order (resp. metric) which is designed to capture some notion of similarity between applicants. Constraining a selection algorithm to satisfy OF will ensure that the algorithm makes decisions based on the partial ranking, thereby benefiting from any fairness properties the partial ranking might confer.

3.2.3 The Poset Secretary Problem

The poset secretary problem is defined as follows: given a set $\mathcal{C}$ of N elements, there is an underlying poset $\mathcal{P} = (\mathcal{C}, \preceq)$ such that $w(a) < w(b)$ whenever $a \prec b$. In the poset secretary problem, one makes irrevocable online selection decisions as elements arrive (in a uniform random order), given access to N and a selection capacity k. Our arrival and decision structure is typical of secretary problems[11] except for differences in the information that is observed. At any point in time, the algorithm can observe rankings according to $\mathcal{P}$ on the elements that have arrived so far. The objective is to minimize the competitive ratio subject to ordinal fairness (Definition 3.3), where competitive ratio is defined as follows:

Definition 3.4. An online algorithm[12] $\mathcal{A}$ is *α-competitive*[13] if the expected performance ratio is at least $\frac{1}{\alpha}$. In other words, it is *α-competitive* if for any instance $I = (w, \mathcal{P})$,

$$\rho(I) := \mathbb{E}\left[\frac{w(\mathsf{Alg}(I))}{\textsc{opt}(I)}\right] \geq \frac{1}{\alpha} \ ,$$

where $\textsc{opt}(I)$ is the offline optimum (i.e., the largest total score which is achievable with knowledge of the true scores of all elements), and $\mathsf{Alg}(I)$ is the output of $\mathcal{A}$. The expectation is taken over the possible orderings of elements and any internal randomness in the algorithm. If there is randomness in the instance generation, then the expectation is taken over that randomness as well; note that in this case, $\textsc{opt}(I)$ would be a random variable.

Given the uniform random arrival order of applicants, we claim that satisfying ordinal fairness is quite easy as shown in the next proposition; the more challenging aspect of our problem will be minimizing the competitive ratio.

Proposition 3.1. *Let $\mathcal{P}$ be a poset of size N, and suppose a poset secretary algorithm has*

[11] In fact, a constant competitive ratio cannot be achieved if N is unknown [74].

[12] In this paper, we use $\mathcal{A}$ to denote a generic algorithm. When convenient, we include the set of elements and k as parameters to $\mathcal{A}$. For example, $\mathcal{A}(G, k)$ refers to some k-secretary algorithm run on the set G of elements.

[13] This is referred to as *strictly α-competitive* in [39].

the property that $a \prec b$ implies $\mathbb{P}(a \text{ is selected}) \leq \mathbb{P}(b \text{ is selected})$. If the algorithm makes decisions based solely on arrival order and $\mathcal{P}$, then it will satisfy ordinal fairness.

Proof. Let φ be any order isomorphism of $\mathcal{P}$. Let Alg denote the set of selected elements, S_N the set of permutations of N elements, and τ_a the transposition of a and $\varphi(a)$. Then

$$\begin{aligned}\mathbb{P}(a \in \mathsf{Alg}) &= \sum_{\sigma \in S_N} \frac{1}{N!} \mathbb{P}(a \in \mathsf{Alg} \mid \sigma) \\ &\overset{(*)}{=} \sum_{\sigma \in S_N} \frac{1}{N!} \mathbb{P}\big(\varphi(a) \in \mathsf{Alg} \mid \sigma \circ \tau_a\big) \\ &= \mathbb{P}\big(\varphi(a) \in \mathsf{Alg}\big),\end{aligned}$$

where $(*)$ follows since a and $\varphi(a)$ are order-isomorphic. □

Having defined sufficient conditions for satisfying ordinal fairness, we point an interested reader to Proposition 3.4 for an example of an algorithm which does not satisfy ordinal fairness.

3.3 A lower bound

We begin by providing a lower bound on competitive ratio. Intuitively, the less information contained in the poset, the less capacity we have to make good selections. We quantify this intuition using the width of the poset, which can be thought of as a measure of sparsity.

Proposition 3.2. *Let k and ω be fixed. There is an instance $\mathcal{I} = (\mathcal{P}, w)$ of the poset secretary problem with k selections and width ω such that $\frac{OPT(\mathcal{I})}{\mathbb{E}[w(\mathsf{Alg}(\mathcal{I}))]} \geq \omega$ for any algorithm.*

To justify this, we construct an instance of group bias with ω groups, where most of the score is concentrated in one group. We show that as the amount of total score assigned to the other $\omega - 1$ groups goes to zero, the competitive ratio tends to ω. The full proof is below.

Proof of Proposition 3.2. Choose the number of elements $n = \min\{n : n \geq \omega k \text{ and } \omega \mid n\}$. Then choose the ordinal structure shown in Figure 3.3. Let a_j^i be the jth best element in the ith chain.

Let the algorithm $\mathcal{A}$ be fixed, and let p_j^i be the probability of selection of the jth best element in the ith chain. Without loss of generality, let $\omega = \arg\min_i \sum_j p_j^i$. Then $\sum_j p_j^\omega \leq k/\omega$.

We now define a scoring of the elements. First, fix some $M > 0$, which will represent the optimal total score. Let $w'(e_j^\omega) = \frac{M}{k}$ for all j, and let $w'(a_j^i) = 0$ otherwise. For the sake of generating distinct scores, we will define a perturbation w of the scoring w'. In particular, fix $\varepsilon > 0$, and let $w(\cdot)$ be a perturbation of $w'(\cdot)$ where $w(a_j^i) = w'(a_j^i) + \varepsilon_j^i$ are distinct, $0 \leq \varepsilon_j^i < \frac{M}{2k}$ for all $i < \omega$ and j, $|\varepsilon_j^\omega| < \frac{M}{2k}$ for all j, $\sum_{i<\omega,j} \varepsilon_j^i \leq \frac{\varepsilon \cdot M}{\omega}$, and $\sum_j \varepsilon_j^\omega = 0$. Then the optimum with respect to $\mathcal{I} = (\mathcal{P}, w)$ is $\textsc{opt}(\mathcal{I}) = M$. So, letting $\mathsf{Alg}(\mathcal{I})$ denote the output of the algorithm, we have that

$$\begin{aligned}
\mathbb{E}[w(\mathsf{Alg}(\mathcal{I}))] &= \sum_{i<\omega,j} p_j^i \varepsilon_j^i + \sum_j p_j^\omega \left(\frac{\text{OPT}(\mathcal{I})}{k} + \varepsilon_j^\omega \right) \\
&\leq \sum_{i<\omega,j} \varepsilon_j^i + \sum_j p_j^\omega \frac{\text{OPT}(\mathcal{I})}{k} \\
&\leq \frac{(1+\varepsilon)\text{OPT}(\mathcal{I})}{\omega}.
\end{aligned}$$

Letting $\varepsilon \to 0$, we see that $\frac{\text{OPT}(\mathcal{I})}{\mathbb{E}[w(\mathsf{Alg}(\mathcal{I}))]} \geq \omega$. Thus, for any algorithm $\mathcal{A}$ and any fixed k and ω, we have that $\sup_{\mathcal{I}} \frac{\text{OPT}(\mathcal{I})}{\mathbb{E}[w(\mathsf{Alg}(\mathcal{I}))]} \geq \omega$.

□

3.4 LP-Based Algorithms

In the vanilla k-secretary setting, Buchbinder et al. showed that optimal algorithms can be computed using linear programming [47]. The main advantages of this approach are that (1) additional constraints, such as the *position independence* constraint studied in [47], can

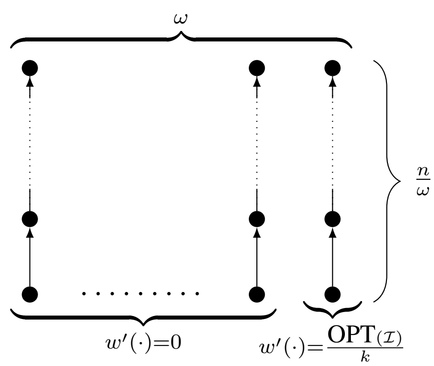


Figure 3.3: The poset constructed in the proof of Proposition 3.2.

be incorporated into the LP, and (2) analysis of the LPs can give lower bounds on achievable competitive ratios. The main drawback to this approach is that the asymptotic analysis of these linear programs as $N \to \infty$ is quite difficult. The method of Buchbinder et al. was extended to the disjoint union of chains for $k = 1$ selections by Kumar et al. [68], but remained open for general posets. In this section, I extend the methods of Buchbinder et al. to the general poset setting for $k = 1$ selections.

The central idea behind using linear programming to develop a secretary algorithm is to solve for p_i, the probability that the algorithm selects the ith element to arrive. Under some minor assumptions on how the algorithm makes decisions, these probabilities uniquely induce an algorithm. In particular, in this section, I will focus on *structure-agnostic algorithms*, which I introduce next.

Definition 3.5 (record)**.** Let $\mathcal{P} = (\{a_1, \dots, a_N\}, \preceq)$ be a poset to be presented to a poset secretary algorithm. In a given run of the algorithm, let $\mathcal{P}_t$ be the poset induced by the first t elements presented to the algorithm. If the element a arrives at time t and a is maximal in $\mathcal{P}_t$, then a is called a *record*.

Definition 3.6 (structure agnostic)**.** A *structure agnostic* poset secretary algorithm (for $k = 1$ selection) is one which (1) only selects elements which are records, and (2) makes selection decisions based only on record status and time of arrival.

Now let R_i be the event that the ith element to arrive is a record, and define $\mu^i_j = \mathbb{P}(R_j \mid$

R_i) and $\Pi_i = \mathbb{P}(R_i)$. The linear program that I will use to generate secretary algorithms, described next, uses these expressions. Note that advance knowledge of the poset is needed to calculate μ^i_j and Π_i.

$$\text{LP}(\mathcal{P}): \quad \begin{array}{ll} \text{maximize} & \frac{1}{N}\sum_{i=1}^{N} \frac{p_i}{\Pi_i^{\mathcal{P}}} \\ \text{subject to} & \frac{p_i}{\Pi_i} \leq 1 - \sum_{j=1}^{i-1} \frac{\mu^i_j}{\Pi_j} p_j \text{ for all } i = 1, \dots, N \\ & p \geq 0 \end{array}$$

Lemma 3.1. *Let $\mathcal{A}$ be a structure agnostic poset 1-secretary algorithm which selects the ith element to arrive with probability p_i. Then p is a feasible point of LP($\mathcal{P}$). Moreover, the competitive ratio of $\mathcal{A}$ is $(\frac{1}{N}\sum_{i=1}^{N} \frac{p_i}{\Pi_i})^{-1}$.*

Proof. For ease of notation, let R_i be the event that the ith element is a record, let S_i be the event that the algorithm selects the ith element to arrive. Then

$$\begin{aligned} p_i &= \mathbb{P}(S_i) \\ &= \mathbb{P}(S_i \cap R_i) \\ &= \mathbb{P}(S_i \mid R_i)\Pi_i \\ &\leq \Big[1 - \mathbb{P}(S_1 \cup \cdots \cup S_{i-1} \mid R_i)\Big]\Pi_i \\ &= \left[1 - \sum_{j=1}^{i-1} \mathbb{P}(S_j \mid R_i)\right]\Pi_i \\ &= \left[1 - \sum_{j=1}^{i-1} \mu^i_j \mathbb{P}(S_j \mid R_j, R_i)\right]\Pi_i \\ &= \left[1 - \sum_{j=1}^{i-1} \mu^i_j \mathbb{P}(S_j \mid R_j)\right]\Pi_i \\ &= \left[1 - \sum_{j=1}^{i-1} \frac{\mu^i_j}{\Pi_j} p_j\right]\Pi_i. \end{aligned}$$

So, p is feasible.

Next, I argue that the competitive ratio of $\mathcal{A}$ is $(\frac{1}{N}\sum_{i=1}^{N}\frac{p_i}{\Pi_i})^{-1}$. To see this, let i^* be the arrival time of the best element, and note that

$$\begin{aligned}\mathbb{P}(S_{i^*}) &= \sum_{i=1}^{N}\mathbb{P}(S_{i^*}\cap\{i^*=i\})\\ &= \frac{1}{N}\sum_{i=1}^{N}\mathbb{P}(S_{i^*}\mid\{i^*=i\})\\ &= \frac{1}{N}\sum_{i=1}^{N}\mathbb{P}(S_i\mid R_i)\\ &= \frac{1}{N}\sum_{i=1}^{N}\frac{p_i}{\Pi_i}.\end{aligned}$$

Thus, the competitive ratio is the reciprocal of this value. □

Lemma 3.2. *Let p be a feasible point in LP$(\mathcal{P})$. Consider the following structure agnostic poset 1-secretary algorithm $\mathcal{A}(p)$: if the algorithm sees a record at time i and has not yet made any selections, the current element is selected with probability $\frac{p_i/\Pi_i}{1-\sum_{j=1}^{i-1}\frac{\mu_j^i}{\Pi_j}p_j}$. Then $\mathcal{A}(p)$ selects the ith element to arrive with probability p_i, and the competitive ratio of $\mathcal{A}(p)$ is $(\frac{1}{N}\sum_{i=1}^{N}\frac{p_i}{\Pi_i})^{-1}$.*

Proof. I prove this by induction. As above, let R_i be the event that the ith element is a record, let S_i be the event that the algorithm selects the ith element to arrive. First observe that $\mathbb{P}(S_1) = \frac{p_1/\Pi_1^{\mathcal{P}}}{1} = p_1$, since the first element is always a record. Now suppose that

$\mathbb{P}(S_i) = p_i$ for some $1 \leq i < N$. Then

$$\begin{aligned}
\mathbb{P}(S_{i+1}) &= \mathbb{P}(S_{i+1} \cap R_{i+1}) \\
&= \mathbb{P}(S_{i+1} \mid R_{i+1})\Pi_{i+1} \\
&= \mathbb{P}(S_{i+1} \mid \overline{S_1 \cup \cdots \cup S_i}, R_{i+1})\Pi_{i+1}\mathbb{P}(\overline{S_1 \cup \cdots \cup S_i} \mid R_{i+1}) \\
&= \frac{p_{i+1}/\Pi_{i+1}}{1 - \sum_{j=1}^{i} \frac{\mu_j^{i+1}}{\Pi_j} p_j} \Pi_{i+1}\mathbb{P}(\overline{S_1 \cup \cdots \cup S_i} \mid R_{i+1}) \\
&= \frac{p_{i+1}/\Pi_{i+1}}{1 - \sum_{j=1}^{i} \frac{\mu_j^{i+1}}{\Pi_j} p_j} \Pi_{i+1}\Big(1 - \mathbb{P}(S_1 \cup \cdots \cup S_i \mid R_{i+1})\Big) \\
&= \frac{p_{i+1}/\Pi_{i+1}}{1 - \sum_{j=1}^{i} \frac{\mu_j^{i+1}}{\Pi_j} p_j} \Pi_{i+1}\left(1 - \sum_{j=1}^{i} \mathbb{P}(S_j \mid R_{i+1})\right) \\
&= \frac{p_{i+1}/\Pi_{i+1}}{1 - \sum_{j=1}^{i} \frac{\mu_j^{i+1}}{\Pi_j} p_j} \Pi_{i+1}\left(1 - \sum_{j=1}^{i} \frac{\mu_j^{i+1}}{\Pi_j} p_j\right) \\
&= p_{i+1}.
\end{aligned}$$

Finally, I argue that the competitive ratio of $\mathcal{A}(p)$ is $(\frac{1}{N}\sum_{i=1}^{N} \frac{p_i}{\Pi_i})^{-1}$. To see this, let i^* be the arrival time of the best element, and note that

$$\begin{aligned}
\mathbb{P}(S_{i^*}) &= \sum_{i=1}^{N} \mathbb{P}(S_{i^*} \cap \{i^* = i\}) \\
&= \frac{1}{N} \sum_{i=1}^{N} \mathbb{P}(S_{i^*} \mid \{i^* = i\}) \\
&= \frac{1}{N} \sum_{i=1}^{N} \mathbb{P}(S_i \mid R_i) \\
&= \frac{1}{N} \sum_{i=1}^{N} \frac{\mathbb{P}(S_i \cap R_i)}{\mathbb{P}(R_i)} \\
&= \frac{1}{N} \sum_{i=1}^{N} \frac{\mathbb{P}(S_i)}{\Pi_i} \\
&= \frac{1}{N} \sum_{i=1}^{N} \frac{p_i}{\Pi_i}.
\end{aligned}$$

□

Combining the lemmas above, we see that there is a one-to-one correspondance between structure agnostic algorithms for the poset 1-secretary algorithm on $\mathcal{P}$ and solutions to LP($\mathcal{P}$). This implies the following result:

Proposition 3.3. *Let $\mathcal{P}$ be a poset. The optimal competitive ratio for any structure agnostic poset 1-secretary algorithm, with advance knowledge of $\mathcal{P}$, is* $1/\operatorname{LP}(\mathcal{P})$.

As noted above, this approach requires advance knowledge of the poset, and so does not give us algorithms for the setting described in Section 3.2.3. Sections 3.5-3.6 provides algorithms which require no structural knowledge of the poset except for its size N.

3.5 Random Partition Algorithms

In this section, I discuss random partitioning approaches to the poset secretary problem. Random partitioning is a technique used in some matroid secretary problems which involves (1) partitioning the elements randomly and (2) selecting at most one element from each part. As a warm-up, in Section 3.5.1, I discuss how a partition of the poset into chains, if available, would allow for the design of a competitive algorithm. Next, I propose and analyze an algorithm requiring an estimate of the width of the poset in Section 3.5.2, and finally, I show how this assumption of a width estimate can be dropped in Section 3.5.3.

3.5.1 Chain Decomposition

Here we discuss the structure of posets and how it might be exploited in the development of a poset secretary algorithm. In particular, we will attempt to bridge the gap between classical secretary methods and the partial-ordinal setting, and in the process, discuss how these methods can and cannot be extended to account for partial-ordinal information. Since classical secretary algorithms assume a total ordering on elements, the simplest way to extend these algorithms is to run them over totally-ordered subsets (i.e., *chains*) of the

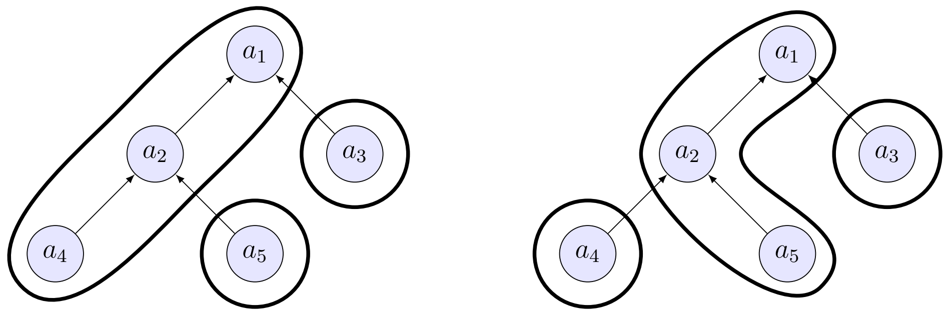


Figure 3.4: Two chain partitions of a 5-element interval order. An arrow from a_i to a_j means that $a_i \prec a_j$.

poset. As we discuss below, this idea fails to solve our problem, but we carry out the example for pedagogical purposes.

Consider a *chain decomposition* of the poset $\mathcal{P} = (\mathcal{C}, \preceq_p)$; that is, a partition $C_1, \ldots, C_\omega$ of $\mathcal{C}$ such that each C_j is a chain and ω is the width of $\mathcal{P}$ (e.g., see Figure 3.4).[14] Suppose, for the sake of this example, that one had access to chain sizes $|C_1|, \ldots, |C_\omega|$ and an oracle which outputs membership in $C_1, \ldots, C_\omega$. We will now bound the competitive ratio of the algorithm $\mathcal{A}_p$ which chooses $j \in [\omega]$ uniformly at random and runs an α-competitive algorithm (for some constant α) on C_j. Letting Alg_p be the output of $\mathcal{A}_p$, $\mathsf{Alg}(C_j)$ the output of the α-competitive algorithm on C_j, $\textsc{opt}(C_j)$ the total score of the top k elements in C_j, and $\textsc{opt}$ the overall optimum, we have that

$$\mathbb{E}\big(w(\mathsf{Alg}_p)\big) = \frac{1}{\omega}\sum_{j=1}^{\omega}\mathbb{E}\big(w(\mathsf{Alg}(C_j,k))\big) \geq \frac{1}{\omega\alpha}\sum_{j=1}^{\omega}\textsc{opt}(C_j) \geq \frac{\textsc{opt}}{\omega\alpha} ,$$

thus achieving an order-optimal competitive ratio of $\omega\alpha$.

There are, however, two issues with this approach. First, the structure of the poset is not known in advance, nor are chain sizes and membership oracles. Second, depending on

[14] This can always be done, as discussed in Section 2.5.

how the chain partition was constructed, the algorithm may not satisfy ordinal fairness,[15,16] since ordinal information is lost in restricting to chains. The next statement proves this claim.

Proposition 3.4 (chain parallelization does not satisfy OF)**.** *Let $\mathcal{A}$ be an algorithm for the vanilla k-secretary problem. Consider the poset secretary algorithm $\mathcal{A}_p$ which, given a chain decomposition $C_1, \ldots, C_\omega$ of optimal size and a chain-membership oracle, chooses a chain $i \in [\omega]$ uniformly at random and runs $\mathcal{A}$ to select at most k elements from C_i. The algorithm $\mathcal{A}_p$ does not necessarily satisfy OF with respect to the original poset.*

Proof. Consider the algorithm $\mathcal{A}$ which selects each arriving element with probability k/n until k selections have been made. Now consider the poset $\{a_1, a_2, a_3\}$, where $a_2 \prec a_1$, $a_3 \prec a_1$, and a_2 is incomparable with a_3, and let $k = 1$. Suppose the algorithm is given the chain decomposition $C_1 = \{a_2\}$ and $C_2 = \{a_1, a_3\}$. Then $\mathcal{A}_p$ will select a_2 with probability $1/2$ but will select a_3 with probability $3/16$. Since a_2 is order-isomorphic to a_3, the algorithm $\mathcal{A}_p$ does not satisfy OF. □

I will address these issues in the next subsections by presenting algorithms that do not assume prior knowledge of the structure of the poset.

3.5.2 ProxyLabel: Algorithm assuming access to a width proxy

As a step toward introducing the first of our two main algorithms, we consider the special case where a *width estimate* ω' of the poset is known. Having this (albeit minor) structural

[15]This approach allowed us to achieve a $3e^2\omega'$-competitive algorithm for interval orders using [75] (see Appendix F in [53]), where ω' is a known upper bound on the width of the poset. However, (i) in general, such an approach may not be possible, since online chain partitioning itself is challenging for general posets, (ii) the competitive ratio incurred additional factors due to the limitations of online chain partitioning, and (iii) this approach requires an upper bound on the width of the poset. For these reasons, I am excluded its discussion from this dissertation.

[16]This example additionally highlights a tension between equality of opportunity and equality of outcomes when these chains are demographic groups. Elements in different groups might have the same probability of selection, but all of the selected elements will be from one group. One can think of this as a disparity in *ex-post* (after sampling a group) treatment despite no disparity in *ex-ante* (pre-sampling) treatment. This gives the appearance of discrimination, and certainly does not promote diversity. The algorithms presented in Sections 3.5.3 and 3.7 do not have this issue.

Algorithm 2: PROXYLABEL for poset bias with a width proxy.

input: no. of elements N, estimate ω' on width (i.e., the width proxy), selection capacity k

Assign a label $\ell(a) \in [k]$ uniformly at random to each a as they arrive

Let $m \sim \mathrm{Bin}(N, \frac{\omega'}{\omega'+1})$ and $S \leftarrow \{\text{first } m \text{ elements}\}$ `// sample m elements`

for $a \notin S$ *with* $\ell(a) = \ell$ **do**

  if *a is maximal with respect to previously seen ℓ-labeled elements, and no ℓ-labeled element has been selected* **then**

    Select a

knowledge will allow us to develop a competitive algorithm which we will adapt in the next subsection to the general case. The idea behind this width-proxy-based algorithm, which we call PROXYLABEL, is to randomly partition the elements into k sets, collect a random sample (with sampling probability tuned using ω'), and then select the first maximal element from each of the random sets. Randomly partitioning the elements is one way to "break up" high-scoring elements, thus reducing the competition between them. By not restricting to chains, we are considering all available ordinal information, thus avoiding the fairness issues discussed in the previous subsection.

Proposition 3.5. *Let ω' be the estimated width of the partial order, and let ω be the true width. If $\omega' \geq \frac{1}{c}\omega$, for some $c \geq 1$, then* PROXYLABEL *(Algorithm 2) has competitive ratio at most $(1-(1-1/k)^k)^{-1}\mathrm{e}^c(\omega'+1) \leq \frac{\mathrm{e}^{c+1}}{\mathrm{e}-1}(\omega'+1)$ for any $k \geq 1$.*

Corollary 3.1. *If the width ω is known, then Algorithm 2 has competitive ratio at most $\left(1-(1-\frac{1}{k})^k\right)^{-1}\mathrm{e}(\omega+1) \leq \frac{\mathrm{e}^2}{\mathrm{e}-1}(\omega+1)$ for $k \geq 1$, and this is order-optimal.*

We next present the proof of the above proposition. The key idea is to first bound the probability that an element a_j among the top k by score is also a maximum-score element in its assigned label ℓ_j, and to then bound the probability that a maximum-score element in a labeled group is selected by PROXYLABEL. The lower bound for the latter follows from considering an arbitrary chain partition of the ℓ_j-labeled elements: depending on which of the top elements in each chain were sampled, one can bound the selection probability of the maximum-score element.

Proof of Proposition 3.5. Let $w(a_1) > w(a_2) > \cdots > w(a_k)$ be the scores of the top k elements. Consider any element a_j among these k elements, and let $\ell(a_j) = \ell_j$. Then the probability that a_j has maximum score in its labeled group can be bounded as follows:

$$\mathbb{P}\Big(\big\{w(a_j) = \max\{w(a) : \ell(a) = \ell_j\}\big\}\Big) = \mathbb{P}\left(\bigcap_{i=1}^{j-1}\{\ell(a_i) \neq \ell_j\}\right) = \left(1 - \frac{1}{k}\right)^{j-1}.$$

We can then bound the probability of selection of a_j by analyzing elements with label ℓ_j. Letting Alg denote the set of elements returned by the algorithm, we have:

$$\begin{aligned}
\mathbb{P}\Big(\{a_j \in \mathsf{Alg}\}\Big) &\geq \mathbb{P}\Big(\big\{a_j \in \mathsf{Alg}\big\} \cap \big\{w(a_j) = \max\{w(a) : \ell(a) = \ell_j\}\big\}\Big) && (3.1)\\
&= \mathbb{P}\Big(\{w(a_j) = \max\{w(a) : \ell(a) = \ell_j\}\}\Big) && (3.2)\\
&\quad \cdot \mathbb{P}\Big(\{a_j \in \mathsf{Alg}\} \mid \{w(a_j) = \max\{w(a) : \ell(a) = \ell_j\}\}\Big) \\
&= \left(1 - \frac{1}{k}\right)^{j-1} \underbrace{\mathbb{P}\Big(\{a_j \in \mathsf{Alg}\} \mid \{w(a_j) = \max\{w(a) : \ell(a) = \ell_j\}\}\Big)}_{=:A} && (3.3)
\end{aligned}$$

Note that the inequality in (3.1) may not be an equality, since a_j can be selected even if it is not a maximum-score ℓ_j-labeled element. To bound A, consider a partition $\{C_1, C_2, \ldots, C_{\omega_{\ell_j}}\}$ of the ℓ_j-labeled elements into $\omega_{\ell_j} \leq \omega$ chains, with $a_j \in C_1$. For each chain C_r, let $f_s^{(r)}$ denote the s-th best element in C_r. Then a_j will be selected if (but not only if) the following conditions hold[17]:

1. a_j is not sampled (i.e., $a_j \notin S$),

2. the second best element in C_1 is sampled (i.e., $f_2^{(1)} \in S$), and

[17] if $f_2^{(1)}$ does not exist, then we can replace the corresponding event with the empty event. This only increases the chances of selecting a_j.

3. the best elements in all other chains are sampled (i.e., $f_1^{(i)} \in S$ for $i = 2, \ldots, \omega_{\ell_j}$).

It follows that

$$
\begin{aligned}
A &\geq \mathbb{P}\left(\{a_j \notin S\}\right) \mathbb{P}\left(\{f_2^{(1)} \in S\}\right) \mathbb{P}\left(\bigcap_{i=2}^{\omega_{\ell_j}} \{f_1^{(i)} \in S\}\right) \\
&\geq \left(\frac{1}{\omega' + 1}\right)\left(1 - \frac{1}{\omega' + 1}\right)^{\omega}. \qquad (3.4)
\end{aligned}
$$

Now suppose that $\frac{1}{c}\omega \leq \omega'$, for some $c \geq 1$. Then, using (3.3) and (3.4), we get:

$$
\begin{aligned}
\mathbb{P}(\{a_j \in \mathsf{Alg}\}) &\geq \left(1 - \frac{1}{k}\right)^{j-1} A \\
&\geq \left(1 - \frac{1}{k}\right)^{j-1} \left(\frac{1}{\omega' + 1}\right)\left(1 - \frac{1}{\omega' + 1}\right)^{\omega} \\
&\geq \left(1 - \frac{1}{k}\right)^{j-1} \frac{1}{\mathrm{e}^c(\omega' + 1)}.
\end{aligned}
$$

Thus, $\mathbb{E}\big(w(\mathsf{Alg})\big)$ can be bounded below using the Chebyshev sum inequality:

$$
\begin{aligned}
\sum_{i=1}^{k} \mathbb{P}(\{a_i \in \mathsf{Alg}\}) w(a_i) &\geq \sum_{i=1}^{k} \frac{\left(1 - \frac{1}{k}\right)^{i-1} w(a_i)}{\mathrm{e}^c(\omega' + 1)} \\
&\geq \frac{1}{k} \sum_{i=1}^{k} \frac{\left(1 - \frac{1}{k}\right)^{i-1} \mathrm{OPT}}{\mathrm{e}^c(\omega' + 1)} \\
&= \frac{\left(1 - \left(1 - \frac{1}{k}\right)^{k}\right) \mathrm{OPT}}{\mathrm{e}^c(\omega' + 1)}.
\end{aligned}
$$

□

We have just shown that when a width estimate $\omega' \geq \omega / c$ is known, one can design an $\mathcal{O}(\mathrm{e}^c \omega')$-competitive algorithm for the secretary problem under partial ordinal information. In the next section, we will show how to circumvent this need for a width estimate.

Algorithm 3: POSETLABEL for partial ordinal information.

input: number of elements N, number to hire k

Assign a label $\ell(a) \in [k]$ uniformly at random to each a as they arrive

Let $m_1 \sim \mathrm{Bin}(N, \frac{1}{2})$, $S' \leftarrow$ {first m_1 elements}, and ω' the width of the subposet S'

Let $m_2 \sim \mathrm{Bin}(N, \frac{2\omega'}{2\omega'+1})$ and $S \leftarrow$ {first $\max\{m_1, m_2\}$ elements} (so, $S \supseteq S'$)

for $a \notin S$ *with* $\ell(a) = \ell$ **do**

  if a is maximal with respect to previously seen ℓ-labeled elements, and no ℓ-labeled element has been selected **then** Select a

3.5.3 PosetLabel: Algorithm not assuming access to a width proxy

In this section, we present the first poset secretary algorithm, POSETLABEL, which only requires the knowledge of the number of elements as described in our problem formulation (Section 3.2.3). This algorithm is an adaptation of PROXYLABEL which foregoes the assumption on a width estimate.

Recall that PROXYLABEL uses a width proxy ω' to set a sampling probability. To get around this, POSETLABEL implements a two-stage sampling procedure. First, an initial sample is taken for the purpose of width-estimation. A secondary sample is then taken based on this estimate. This two-stage sampling procedure allows the sampling to begin *without knowledge of the ultimate sample size*, which is crucial since the width is a priori unknown. Before analyzing textscPosetLabel, I illustrate it by providing an example run.

Example 3.7. Consider the instance of poset bias where $k = 2$ selections can be made, and the input poset is shown on the left in Figure 3.5. POSETLABEL will begin by randomly choosing a sequence of k-nary labels: $(0, 0, 1, 0, 1)$. Suppose the sequence of elements that will be given to the algorithm is

$$a_2,\ \ a_4,\ \ a_1,\ \ a_3,\ \ a_5.$$

This means that $\{a_2, a_3, a_4\}$ will receive label 0, and $\{a_1, a_5\}$ will receive label 1, as shown by the coloring on the right in Figure 3.5.

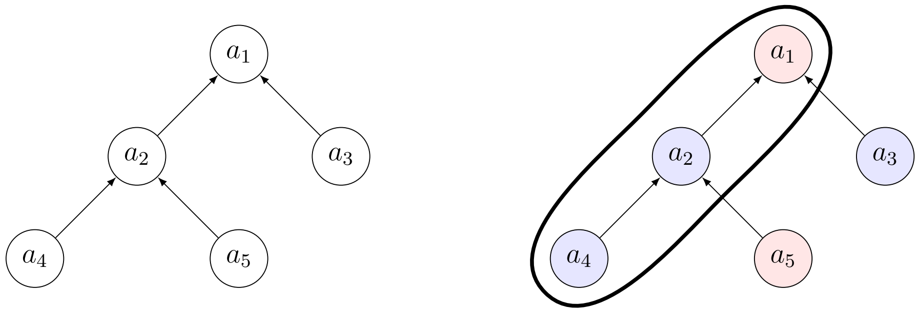


Figure 3.5: The input poset is shown on the left. On the right, the same poset has been annotated to show the sample S (circled) as well as the random labeling, where the blue vertices (a_2, a_3, a_4) are assigned the first label, and the red vertices (a_1, a_5) are assigned the second label.

Next, the algorithm will draw a number, say 2, from $\mathrm{Bin}(5, 1/2)$. This will produce an initial random sample $S' = \{a_2, a_4\}$. Since the width of S' is 1, we will next draw a number, say 3, from $\mathrm{Bin}(5, 2/3)$. The full sample S will then be the first $\max\{2, 3\} = 3$ elements, i.e., $S = \{a_1, a_2, a_4\}$.

Finally, the algorithm enters its selection phase. It first observes a_3, which is indeed maximal in its label and thus is selected. Element a_5 is then observed but not selected, since it is ranked lower than a_1. In this example, the set of elements returned by the algorithm would be $\{a_3\}$.

I next show that POSETLABEL is $\mathcal{O}(\omega)$-competitive.

Proposition 3.6. *Under the poset bias setting, Algorithm 3 is* $\frac{2e^3}{e-1}\big(2\omega + 1\big)\big(1 + o(1)\big)$*-competitive, as the number of elements* $N \to \infty$*. Moreover, this algorithm satisfies ordinal fairness, and does not need to know the width of the partial order in advance.*

As was the case with PROXYLABEL, the random partition of elements reduces the probability that the best elements are clustered in one chain. Then, by forming a chain partition of elements using the minimum number of chains, we can bound the probability of selecting a maximal element within its labeled group. Finally, we can show that our estimated width ω' is within a factor of 4 of the true width ω with large probability, giving us the $\mathcal{O}(\omega)$ competitive ratio in the proposition.

Proof of Proposition 3.6. Let S' be the initial sample of $m_1 \sim \mathrm{Bin}(N, 1/2)$ elements, and let ω' be the width of S'. Let $\mathsf{Alg}_2(\omega')$ denote the output of PROXYLABEL (Algorithm 2) with width proxy ω', and let Alg_3 denote the output of POSETLABEL (Algorithm 3).

We begin by bounding ω'. Let $A = \{a_1, \ldots, a_\omega\}$ be a (maximal) set of mutually incomparable elements, and let $|A \cap S'| = X \sim \mathrm{Bin}(\omega, \frac{1}{2})$. In order to apply our results from Proposition 3.5, we must bound the probability that ω' is close to ω. To that end, letting $X_\omega = \left\{\omega' \geq \frac{\omega}{4}\right\}$,

$$\mathbb{P}(X_\omega) \geq \mathbb{P}\Big(\Big\{X \geq \frac{1}{4}\omega\Big\}\Big) \geq \frac{1}{2}. \tag{3.5}$$

After sampling m_1 elements, the algorithm uses its estimated width ω' to sample $m_2 \sim \mathrm{Bin}(N, \frac{2\omega'}{2\omega'+1})$. Now in order to reduce to the analysis of Proposition 3.5, we need to bound the probability that $m_2 \geq m_1$. Letting $Y_N = \{m_2 \geq m_1\}$ (where the subscript N refers to the number of elements), we have that $\mathbb{P}(Y_N) \geq \mathbb{P}(\{m_2 \geq m_1 \geq 1\})$. When $m_1 \geq 1$, we have that $\omega' \geq 1$, and so the probability $\frac{2\omega'}{2\omega'+1}$ of inclusion into S is at least $\frac{2}{3}$. It follows that for $Z_N \sim \mathrm{Bin}(N, \frac{2}{3})$,

$$\mathbb{P}(Y_N) \geq \mathbb{P}(\{m_2 \geq m_1 \geq 1\}) \geq \mathbb{P}(\{Z_N \geq m_1 \geq 1\}) \xrightarrow{N\to\infty} 1.$$

This is because $\mathbb{E}(m_1) = \frac{N}{2}$, and $\mathbb{E}(Z_N) = \frac{2N}{3}$. Since these expectations differ by $\Theta(\sqrt{N})$ standard deviations, Y_N occurs with high probability, and since Y_N happens with high probability,

$$\mathbb{P}(X_\omega \cap Y_N) \xrightarrow{N\to\infty} \mathbb{P}(X_\omega)\,. \tag{3.6}$$

Conditioning on the event $X_\omega \cap Y_N$, it is as if we simply ran Algorithm 2 (PROXYLABEL), so we can use the analysis from that algorithm.

$$\begin{aligned}
\mathbb{E}(w(\mathsf{Alg}_3)) &\geq \mathbb{P}\left(X_\omega \cap Y_N\right) \mathbb{E}\left(w(\mathsf{Alg}_3) \;\middle|\; X_\omega \cap Y_N\right) \\
&= \big(1 - o(1)\big)\mathbb{P}(X_\omega)\mathbb{E}\left(w\big(\mathsf{Alg}_2(2\omega')\big) \;\middle|\; \left\{\omega' \geq \frac{\omega}{4}\right\}\right) && \text{by (3.6)} \\
&\geq \frac{\big(1 - o(1)\big)}{2}\mathbb{E}\left(w\big(\mathsf{Alg}_2(2\omega')\big) \;\middle|\; \left\{\omega' \geq \frac{\omega}{4}\right\}\right) && \text{by (3.5)} \\
&\geq \frac{\big(1 - o(1)\big)}{2}\Bigg[p\mathbb{E}\Big(w\big(\mathsf{Alg}_2(2\omega')\big) \;\Big|\; \big\{2\omega' \geq \omega\big\}\Big) \\
&\qquad + (1-p)\mathbb{E}\Big(w\big(\mathsf{Alg}_2(2\omega')\big) \;\Big|\; \Big\{\frac{\omega}{2} \leq 2\omega' < \omega\Big\}\Big)\Bigg] && (*) \\
&\geq \frac{\big(1 - o(1)\big)}{2}\left[p\frac{(\mathrm{e}-1)\text{OPT}}{\mathrm{e}^2(2\omega'+1)} + (1-p)\frac{(\mathrm{e}-1)\text{OPT}}{\mathrm{e}^3(2\omega'+1)}\right] \\
&\geq \frac{\big(1 - o(1)\big)}{2}\left[\frac{(\mathrm{e}-1)\text{OPT}}{\mathrm{e}^3(2\omega'+1)}\right] \\
&\geq \frac{\big(1 - o(1)\big)}{2}\left[\frac{(\mathrm{e}-1)\text{OPT}}{\mathrm{e}^3(2\omega+1)}\right],
\end{aligned}$$

where $(*)$ holds for some $p \in [0, 1]$. □

3.6 Adaptive Thresholding

While POSETLABEL achieves an order-optimal competitive ratio, the proved bound on competitive ratio does not improve with increasing N and k. One might expect the achievable performance to improve as k increases, since a small number of bad decisions relative to k may not have a large effect on performance. Indeed, the $(1-5k^{-1/2})^{-1}$-competitive algorithm of [4] for the vanilla k-secretary problem supports this inclination. In this section, we present and analyze a novel algorithm for the poset k-secretary problem which achieves a competitive ratio of $\omega\big(1 + o(1)\big)$ as $k \to \infty$ under the regime where $(\log N)^2 \in o(k)$.

Before presenting the new algorithm, it is worth noting that POSETLABEL is not tight as $k \to \infty$. To see why, it suffices to consider the vanilla case, where $\omega = 1$. Recall that POSETLABEL randomly partitions the elements into k sets and selects at most one element from each set. We show below that any algorithm with this property fails to achieve a tight competitive ratio (Prop. 3.7).

Definition 3.7 (URP k-secretary algorithm)**.** A *uniform-random-partition* (URP) k-secretary algorithm is one that (1) assigns each element independently and uniformly at random to one of k sets, and (2) selects at most one element from each set.

Proposition 3.7. *Any URP k-secretary algorithm cannot achieve a competitive ratio smaller than $\left(1-\left(1-\frac{1}{k}\right)^k\right)^{-1}$. In particular, this means that no URP algorithm can achieve a competitive ratio better than $(1-1/\mathrm{e})^{-1}$ asymptotically in k.*

Proof. Let Alg^* be the offline algorithm that assigns elements to one of k sets uniformly at random, and chooses the best element in each of the sets. Now let $a_1, \dots, a_k$ be the k highest-score elements, with $w(a_1) > \cdots > w(a_k)$. Let $\varepsilon_j = w(a_1) - w(a_j)$ for each $j \le k$ and $\varepsilon = \sum_{j=1}^k \varepsilon_j$. Then

$$\begin{aligned}\mathbb{E}[w(\mathsf{Alg}^*)] &= \sum_{j=1}^k w(a_j)\mathbb{P}(a_j \in \mathsf{Alg}^*) = \sum_{j=1}^k w(a_j)\Big(1-\frac{1}{k}\Big)^{j-1} \\ &\le w(a_1)\sum_{j=1}^k \Big(1-\frac{1}{k}\Big)^{j-1} = w(a_1)\cdot\frac{1-\left(1-\frac{1}{k}\right)^k}{1/k} \\ &\le (\text{OPT}+\varepsilon)\left(1-\Big(1-\frac{1}{k}\Big)^k\right).\end{aligned}$$

Letting $\varepsilon \to 0$, we see that the performance ratio of Alg^* is no better than $\left(1-\left(1-\frac{1}{k}\right)^k\right)^{-1}$. Since Alg^* performs better than any URP algorithm, all such algorithms have competitive ratios no better than $\left(1-\left(1-\frac{1}{k}\right)^k\right)^{-1}$. Moreover, since $\left(1-\left(1-\frac{1}{k}\right)^k\right)^{-1} \to (1-1/\mathrm{e})^{-1}$, we see that URP algorithms cannot achieve a competitive ratio better than $(1-1/\mathrm{e})^{-1}$ as $k \to \infty$. □

Since URP k-secretary algorithms cannot achieve tight competitive ratios, we turn to the *adaptive thresholding* technique which is common in the secretary literature [4, 65]. In the vanilla setting, this technique involves periodically setting a new threshold, and selecting elements whose scores exceed the most recent threshold. This technique is promising for obtaining a tight competitive ratio since it allows for a small number of poor decisions to be made early on and a larger number of decisions to be made later on with more carefully chosen thresholds. The main challenges of adapting this approach to the poset setting are (1) adapting the notion of a threshold, and (2) bounding the expected scores of elements that meet the most recent threshold.

Let us first define some notions which will be useful in describing the algorithm. In what follows, $\overline{X}$ will denote the complement of a set X.

Definition 3.8. Let $\mathcal{P}$ be a poset. The *upset* $\mathcal{U}(\tau)$ of a set τ is $\{b : b \succ a \text{ for some } a \in \tau\}$. The *downset* $\mathcal{D}(\tau)$ of τ is $\{b : b \prec a \text{ for some } a \in \tau\}$. The *selection set* $\sigma(\tau)$ is the set of elements which are better than or incomparable to elements in τ, i.e., $\sigma(\tau) = \{b \mid b \not\preceq a \text{ for all } a \in \tau\} = \overline{\mathcal{D}(\tau)} \setminus \tau$.

In order to achieve a good competitive ratio, we would like as many of the highly ranked elements to be in the selection set $\sigma(\tau)$ of the threshold τ as possible. Since it is unclear from the partial order which elements have the top k scores, the threshold τ must select elements which *could possibly* be among the top k, and not only select elements which are definitely among the top k. This may, for example, involve hedging across the poset: if multiple chains in a given a chain partition contain elements that could possibly be in the top k, then it may be necessary for τ to select elements from those chains. To that end, we introduce the notion of an (m, k)-threshold.

Definition 3.9 ((m, k)-threshold)**.** Let $\mathcal{P}$ be a poset of width ω. An (m, k)-threshold $\tau \subseteq \mathcal{P}$ is an antichain with selection set $|\sigma(\tau)| \leq k$ such that the upset is large enough for each element in τ: $|\mathcal{U}(\{a\})| \geq m$ for all $a \in \tau$.

In a totally ordered set, an (m, k)-threshold must be a singleton $\tau = \{a\}$, and the selection set is the set of elements exceeding a. Generally, the selection set of an (m, k)-threshold τ is the set of elements which are not worse than any element in τ. Any (m, k)-threshold τ hedges across chains containing potentially high-scoring elements, since each element in τ needs to have a large enough upset. For the remainder of the section, we let $m_{\eta,k,\mathcal{P}} = \lfloor (1-\eta)k/\omega \rfloor$, where $\eta \in [0,1)$ and ω is the width of the poset $\mathcal{P}$. The elements that meet such a thresholding antichain τ end up having a high total score for any linear extension of the poset, as shown below.

Lemma 3.3. *Let τ be an $(m_{\eta,k,\mathcal{P}}, k)$-threshold of a scored (weighted) poset $\mathcal{P}$, and let* OPT *denote the sum of the k largest scores in $\mathcal{P}$. Then $w(\sigma(\tau)) \geq$ OPT$\lfloor (1-\eta)k/\omega \rfloor / k$.*

Proof. Let a be among the top $\lfloor (1-\eta)k/\omega \rfloor$ elements in $\mathcal{P}$ by score. If $a \notin \sigma(\tau)$, then $a \preceq b$ for some $b \in \tau$, in which case $\left\lfloor (1-\eta)\frac{k}{\omega} \right\rfloor > |\mathcal{U}(\{a\})| \geq |\mathcal{U}(\{b\})| \geq \left\lfloor (1-\eta)\frac{k}{\omega} \right\rfloor$. Thus, $a \in \sigma(\tau)$, implying that $w(\sigma(\tau)) \geq$ OPT$\lfloor (1-\eta)k/\omega \rfloor / k$. □

The above lemma shows that $(m_{\eta,k,\mathcal{P}}, k)$-thresholds are effective in selecting high-scoring elements. We next show that such $(m_{\eta,k,\mathcal{P}}, k)$-thresholds exist. Together, these two observations indicate that $(m_{\eta,k,\mathcal{P}}, k)$-thresholds could be useful in designing algorithms for poset secretary problems.

Lemma 3.4 (existence of $(m_{\eta,k,\mathcal{P}}, k)$-thresholds)**.** *Let $\mathcal{P}$ be a poset of size N and width ω, let $k \leq N$, and let $\eta \in [0,1)$. There exists an $(m_{\eta,k,\mathcal{P}}, k)$-threshold of $\mathcal{P}$.*

Proof. Let $(C_1, \ldots, C_\omega)$ be a chain decomposition of $\mathcal{P}$ and let $a_{i,1} \succ \cdots \succ a_{i,n_i}$ be the elements of C_i. For $1 \leq i \leq \omega$, define E_i to be the $(\lfloor (1-\eta)k/\omega \rfloor + 1)$th best element in C_i, if it exists, i.e.,

$$E_i = \begin{cases} \{a_{i,\lfloor (1-\eta)k/\omega \rfloor + 1}\} & \text{if } \lfloor (1-\eta)k/\omega \rfloor + 1 \leq n_i, \\ \emptyset & \text{else.} \end{cases}$$

Now let $\tau = \bigcup_{i=1}^{\omega} E_i$. Note that each element $a \in \tau$ satisfies $|\mathcal{U}(\{a\})| \geq \lfloor (1-\eta)k/\omega \rfloor$, and that $|\sigma(\tau)| \leq \omega \lfloor (1-\eta)k/\omega \rfloor \leq \lfloor (1-\eta)k \rfloor \leq k$. However, τ may not be an antichain. In this case, there is some $b_1, b_2 \in \tau$ such that $b_1 \prec b_2$. Now let $\tau' = \tau \setminus \{b_1\}$. We still have the property that $|\mathcal{U}(\{a\})| \geq \lfloor (1-\eta)k/\omega \rfloor$ for each $a \in \tau'$. Moreover, since

$\mathcal{D}(\{b_1\}) \subseteq \mathcal{D}(\{b_2\})$, we have that $|\sigma(\tau')| = |\sigma(\tau)| \leq k$. Thus we can repeat this procedure until we arrive at an antichain. □

Now that we have a good notion of thresholding, we can describe our algorithmic approach, which generalizes the vanilla k-secretary algorithm of [4]. The algorithm randomly splits the element stream into two sets, Y and Z, and recurses on Y to select at most $k/2$ elements. Then, a threshold is chosen from Y, and elements in Z who meet the threshold are selected. See Algorithm 4 for a detailed description. We next bound the competitive ratio of this algorithm when an upper bound on the width of the poset is known in advance.

Algorithm 4: ADATHRESHOLD(S, k)

input: slack $\eta = \eta(N, k)$, element stream S of size N, $k \geq 1$

if $k \geq N$ **then** select all elements

else if $k = 1$ **then** run POSETLABEL(S, k)

else

  Sample $m \sim \mathrm{Bin}(N, 1/2)$ and let Y be the first m elements

  Run ADATHRESHOLD$(Y, \lfloor k/2 \rfloor)$

  Select any $\big(m_{\eta(N,k),\lfloor k/2 \rfloor,Y}, \lfloor k/2 \rfloor\big)$-threshold $\widehat{\tau}$ of Y

  While fewer than k selections have been made, select elements in $S \setminus Y$ that are in $\sigma(\widehat{\tau})$

Proposition 3.8. *Let $\mathcal{P}$ be a poset on N elements of width ω, let $\omega' \geq \omega$ be a known upper bound on ω, and set $D = 4\sqrt{\omega'}$. Then* ADATHRESHOLD *(Algorithm 4) with $\eta = D\sqrt{\frac{\log N}{k}}$ is $\omega(1 - C\sqrt{\frac{\log N}{k}})^{-1}$-competitive, where $C = \frac{7+4D}{4-2\sqrt{2}}$.*

Proof. We proceed by induction on k, mimicking the structure of the algorithm. For simplicity, we assume that k is a power of 2. Note that the bound holds trivially if $k \leq D^2 \log N$, since $1 - C\sqrt{\log N/k} \leq 1 - C/D \leq 1 - 4/(4 - 2\sqrt{2}) < 0$, so we assume that $k > D^2 \log N$. Similarly, if $k \geq N$, then we select all elements and the competitive ratio is 1, so the bound trivially holds, since $1 - C\sqrt{\log N/k} \leq \omega$. If $N \geq 2$ and $k \leq 2$, then the bound again becomes trivial:

$$1 - C\sqrt{\frac{\log N}{k}} \leq 1 - C\sqrt{\frac{\log 2}{2}} \leq 1 - \frac{7 + 16\sqrt{1}}{4 - 2\sqrt{2}} \cdot \sqrt{\frac{\log 2}{2}} < 0.$$

Now suppose that $3 \leq k \leq N$, with $k > D^2 \log N$. We bound the performance of the algorithm on Y (the first half, on which we recurse) and $Z = S \setminus Y$ (the second half) separately. Let $\mathsf{Alg}(S)$ denote the set of elements selected by the algorithm over a set S. Letting $\textsc{opt}(Y, k/2)$ denote the total score of the top $k/2$ elements in Y,

$$\begin{aligned}
\mathbb{E}[w(\mathsf{Alg}(Y))] &\geq \Big(1 - \frac{\sqrt{2}C\sqrt{\log |Y|}}{\sqrt{k}}\Big)\mathbb{E}\Big[\textsc{opt}\Big(Y, \frac{k}{2}\Big)\Big]/\omega \qquad \text{inductive hypothesis} \\
&\overset{(*)}{\geq} \Big(1 - \frac{\sqrt{2}C\sqrt{\log |Y|}}{\sqrt{k}}\Big)\Big(\frac{1}{2} - \frac{1}{4\sqrt{k}}\Big)\frac{\textbf{OPT}}{\omega} \\
&\geq \Big(\frac{1}{2} - \frac{1 + 2\sqrt{2}C\sqrt{\log |Y|}}{4\sqrt{k}}\Big)\frac{\textbf{OPT}}{\omega} \\
&\geq \Big(\frac{1}{2} - \frac{1 + 2\sqrt{2}C\sqrt{\log N}}{4\sqrt{k}}\Big)\frac{\textbf{OPT}}{\omega},
\end{aligned}$$

where $(*)$ follows directly from a computation of [4] and is included as Lemma **??**.

We can similarly bound the performance of the algorithm on Z. To that end, note that $\eta < 1$, since $k > D^2 \log N$, so the algorithm is able to choose a threshold $\widehat{\tau}$. We next bound the probability that the threshold $\widehat{\tau}$ selected by the algorithm is good. Define $\mathcal{B} = \{\tau \in \mathfrak{A}(\mathcal{P}) : |\sigma(\tau)| > k \text{ or } |\mathcal{U}(\{b\})| < (1 - 2\eta)\frac{k}{\omega} \text{ for some } b \in \tau\}$ to be the set of "bad" thresholds (since it selects too many elements or one of its elements has a small upset), where $\mathfrak{A} = \mathfrak{A}(\mathcal{P})$ is the set of antichains of $\mathcal{P}$. Then

$$\mathbb{P}(\widehat{\tau} \in \mathcal{B}) \leq \underbrace{\mathbb{P}\big(|\sigma(\widehat{\tau})| > k\big)}_{=:A} + \underbrace{\mathbb{P}\Big(\bigcup_{b \in \widehat{\tau}} \Big\{|\mathcal{U}(\{b\})| < (1 - 2\eta)\frac{k}{\omega}\Big\}\Big)}_{=:B}.$$

We bound A and B separately, starting with A. For simplicity, we will assume that $(1 -$

$\eta)k/\omega$ and $(1-2\eta)k/\omega$ are integers. Then

$$
\begin{aligned}
A = \mathbb{P}\big(|\sigma(\widehat{\tau})| > k\big) &= \sum_{\substack{\tau \in \mathfrak{A} \\ |\sigma(\tau)| > k}} \mathbb{P}\big(\widehat{\tau} = \tau\big) \leq \sum_{T=k}^{N} \sum_{\substack{\tau \in \mathfrak{A} \\ |\sigma(\tau)| = T}} \mathbb{P}\big(\widehat{\tau} = \tau\big) \\
&\leq \sum_{T=k}^{N} \sum_{\substack{\tau \in \mathfrak{A} \\ |\sigma(\tau)| = T}} \mathbb{P}\Big(|\sigma(\tau) \cap Y| \leq (1-\eta)\frac{k}{2}\Big) \\
&\overset{(a)}{\leq} \sum_{T=k}^{N} \big|\big\{\tau \in \mathfrak{A} : |\sigma(\tau)| = T\big\}\big| \mathrm{e}^{-\eta^2 k/2} \\
&\overset{(b)}{\leq} \sum_{T=k}^{N} \binom{T+\omega-1}{\omega-1} \mathrm{e}^{-\eta^2 k/2} \leq (2N)^{\omega+1} \mathrm{e}^{-\eta^2 k/2} =: A',
\end{aligned}
$$

where (a) follows from Hoeffding's inequality (since the selection set of τ is a fixed set in the poset, and the number of sampled elements of $\sigma(\tau)$ is a binomial random variable). To show (b), I make the following claim:

Claim 3.1. Let $\mathcal{P}$ be a poset of width ω and $m \geq 0$ an integer. Then the number of antichains $\tau \subseteq \mathcal{P}$ with $|\sigma(\tau)| = m$ is at most $\binom{m+\omega-1}{\omega-1}$.

Proof of claim. First, we show that there is an injection from the set of antichains to the set of selection sets. Let τ, τ' be two antichains, and suppose that $\sigma(\tau) = \sigma(\tau')$. Let a be any element in τ, and suppose for the sake of contradiction that $a \notin \tau'$. Since $a \notin \sigma(\tau)$, it must be that $a \notin \sigma(\tau')$ as well, and so $a \prec b$ for some $b \in \tau'$. Thus $b \in \sigma(\tau) \setminus \sigma(\tau')$, a contradiction.

Now let $C_1, \ldots, C_\omega$ be a chain decomposition of $\mathcal{P}$, where $a_{i,1} \prec \cdots \prec a_{i,n_i}$ are the elements of C_i. Define $C_{i,k} = \{a_{i,1}, \ldots, a_{i,k}\}$ to be the prefix of C_i of length k.

For any τ with $|\sigma(\tau)| = m$, we must have that $\sigma(\tau) = \bigcup_{i=1}^{\omega} C_{i,x_i}$, for some $x_1 + \cdots + x_\omega = m$. The number of such sets $\sigma(\tau)$ is bounded by the number of nonnegative integral solutions to $x_1 + \cdots + x_\omega = m$, which is $\binom{m+\omega-1}{\omega-1}$. Moreover, by the above injection, the number of antichains τ with $|\sigma(\tau)| = m$ is bounded by the same. $\square_{\text{Claim}}$

Next, I bound B.

$$
\begin{aligned}
B &= \mathbb{P}\Big(|\mathcal{U}(\{b\})| < (1-2\eta)\frac{k}{\omega} \text{ for some } b \in \widehat{\tau}\Big) \\
&\leq \sum_{b:|\mathcal{U}(\{b\})|\leq(1-2\eta)\frac{k}{\omega}} \mathbb{P}(b \in \widehat{\tau}) \\
&\leq \sum_{b:|\mathcal{U}(\{b\})|\leq(1-2\eta)\frac{k}{\omega}} \mathbb{P}\Big(\underbrace{|\mathcal{U}(\{b\}) \cap Y|}_{\sim \mathrm{Bin}(\lfloor(1-2\eta)k/\omega\rfloor, 1/2)} \geq (1-\eta)\frac{k}{2\omega}\Big) \\
&\leq \sum_{b:|\mathcal{U}(\{b\})|\leq(1-2\eta)\frac{k}{\omega}} \mathbb{P}\Big(|\mathcal{U}(\{b\}) \cap Y| - \frac{1}{2}\Big\lfloor(1-2\eta)\frac{k}{\omega}\Big\rfloor \geq \frac{\eta k}{2\omega}\Big) \\
&\overset{(c)}{\leq} Ne^{-\eta^2 k/2(1-2\eta)\omega} \\
&\leq Ne^{-\eta^2 k/4\omega} =: B',
\end{aligned}
$$

where (c) follows from Hoeffding's inequality. Thus $\mathbb{P}(\widehat{\tau} \in \mathcal{B}) \leq A + B \leq A' + B'$. However, by our choice of η, we have that $A' \leq B'$. So,

$$
\mathbb{P}(\widehat{\tau} \in \mathcal{B}) \leq 2N\mathrm{e}^{-\eta^2 k/4\omega} \overset{(d)}{\leq} \frac{2}{N^3} \leq 2/\sqrt{k}, \tag{3.7}
$$

where (d) follows from the bound $\eta \geq 4\sqrt{\omega}\sqrt{\frac{\log N}{k}}$. Now that we have bounded $\mathbb{P}(\widehat{\tau} \in \mathcal{B})$, we will bound $\mathbb{E}[w(\sigma(\widehat{\tau}) \cap Z) \mid \widehat{\tau} \notin \mathcal{B}]$. To that end, define $H = \{a_1, \ldots, a_{(1-2\eta)k/\omega}\}$ to be the set of elements with the largest $(1-2\eta)k/\omega$ scores in $\mathcal{P}$, and let $\widehat{\mathrm{OPT}} = w(H)$. We define a re-scoring of the elements of $\mathcal{P}$, where all elements except those in H have score 0:

$$
\widehat{w}(a) = \begin{cases} w(a) & \text{if } a \in H \\ 0 & \text{otherwise.} \end{cases}
$$

It follows that $\mathbb{E}[\widehat{w}(\sigma(\widehat{\tau})) \mid \widehat{\tau} \notin \mathcal{B}] \geq \widehat{\mathrm{OPT}}$, since H is a subset of $\sigma(\tau)$ for any $\tau \notin \mathcal{B}$. To prove the claim, it suffices to upper bound the total score of elements of H that appear in

Y. In particular,

$$\mathbb{E}[\widehat{w}(\sigma(\widehat{\tau}) \cap Y) \mid \widehat{\tau} \notin \mathcal{B}] \leq \mathbb{E}[w(H \cap Y)] = \frac{\widehat{\text{OPT}}}{2}.$$

In sum, we have that

$$\begin{aligned}
\mathbb{E}[w(\sigma(\widehat{\tau}) \cap Z) \mid \widehat{\tau} \notin \mathcal{B}] &\geq \mathbb{E}[\widehat{w}(\sigma(\widehat{\tau}) \cap Z) \mid \widehat{\tau} \notin \mathcal{B}] \\
&= \mathbb{E}[\widehat{w}(\sigma(\widehat{\tau})) \mid \widehat{\tau} \notin \mathcal{B}] - \mathbb{E}[\widehat{w}(\sigma(\widehat{\tau}) \cap Y) \mid \widehat{\tau} \notin \mathcal{B}] \\
&\geq \widehat{\text{OPT}} - \frac{\widehat{\text{OPT}}}{2} = \frac{\widehat{\text{OPT}}}{2} \geq (1-2\eta)\frac{\text{OPT}}{2\omega}. \qquad (3.8)
\end{aligned}$$

With this in mind, we can proceed to bounding the performance of the algorithm on Z.

$$\begin{aligned}
\mathbb{E}[w(\mathsf{Alg}(Z))] &\geq \Big(1 - \frac{2}{\sqrt{k}}\Big)\mathbb{E}[w(\mathsf{Alg}(Z)) \mid \widehat{\tau} \notin \mathcal{B}] && \text{by (3.7)} \\
&\geq \Big(1 - \frac{2}{\sqrt{k}}\Big)\mathbb{E}[w(\sigma(\widehat{\tau}) \cap Z) \mid \widehat{\tau} \notin \mathcal{B}] \\
&\geq \Big(1 - \frac{2}{\sqrt{k}}\Big)\Big(1 - 2D\sqrt{\frac{\log N}{k}}\Big)\frac{\text{OPT}}{2\omega} && \text{by (3.8)} \\
&\geq \Big(1 - \frac{2 + 2D\sqrt{\log N}}{\sqrt{k}}\Big)\frac{\text{OPT}}{2\omega} = \Big(\frac{1}{2} - \frac{1 + D\sqrt{\log N}}{\sqrt{k}}\Big)\frac{\text{OPT}}{\omega}.
\end{aligned}$$

In sum,

$$\begin{aligned}
\mathbb{E}[w(\mathsf{Alg})] &\geq \Big[1 - \frac{1 + 2\sqrt{2}C\sqrt{\log n}}{4\sqrt{k}} - \frac{1 + D\sqrt{\log N}}{\sqrt{k}}\Big]\frac{\text{OPT}}{\omega} \\
&= \Big[1 - \frac{5 + (4D + 2\sqrt{2}C)\sqrt{\log N}}{4\sqrt{k}}\Big]\frac{\text{OPT}}{\omega}.
\end{aligned}$$

For the chosen $C = \frac{7+4D}{4-2\sqrt{2}}$, we have that $(4-2\sqrt{2})C - 4D = 7 \geq \frac{5}{\sqrt{\log 2}} \geq \frac{5}{\sqrt{\log N}}$. It follows that $[(4-2\sqrt{2})C - 4D]\sqrt{\log N} \geq 5$, which in turn gives that $5 + (4D + 2\sqrt{2}C)\sqrt{\log N} \leq$

$4C\sqrt{\log N}$. Continuing the above computation, we have that

$$\mathbb{E}[w(\mathsf{Alg})] \geq \left[1 - \frac{5 + (4D + 2\sqrt{2}C)\sqrt{\log N}}{4\sqrt{k}}\right]\frac{\text{OPT}}{\omega} \geq \left[1 - \frac{C\sqrt{\log N}}{\sqrt{k}}\right]\frac{\text{OPT}}{\omega},$$

as desired. □

Note that Proposition 3.8 bounds the competitive ratio in the case where a bound on the width of the poset is known. Asymptotically, however, if k grows quickly enough, this assumption can be dropped, and a tight algorithm can be achieved (see Corollary 3.2).

Corollary 3.2. *Let $\mathcal{P}$ be a poset of width $\omega \leq \log k$. Algorithm 4 run with $\eta(N,k) = \sqrt{\frac{16 \log k \log N}{k}}$ (i.e., with width bound $\omega' = \log k$) is $\omega\Big(1 - \frac{38 \log N}{\sqrt{k}}\Big)^{-1}$-competitive. Thus, if $(\mathcal{P}_N)_{N\geq 1}$ is a sequence of posets of width at most ω with $|\mathcal{P}_N| = N$ and $k = k(N)$ is such that $\log(N)/\sqrt{k} \to 0$, the competitive ratio of Algorithm 4 run with $\eta(N,k) = \sqrt{\frac{16 \log k \log N}{k}}$ is $\omega(1 + o(1))$ as $N \to \infty$.*

Proof. Using the notation from Proposition 3.8, we have that $D = 4 \log k$ and $C = \frac{7+4D}{4-2\sqrt{2}} \leq 38 \log k$. Since $k \to \infty$, we can restrict our attention to posets $\mathcal{P}_N$ for which $\log k(N) \geq \omega$. In this case, the reciprocal of the competitive ratio of the algorithm on $\mathcal{P}_N$ is at least

$$\frac{1}{\omega}\Big(1 - \frac{38\sqrt{\log k}\sqrt{\log N}}{\sqrt{k}}\Big) \geq \frac{1}{\omega}\Big(1 - \frac{38 \log N}{\sqrt{k}}\Big).$$

Thus, if $\log(N)/\sqrt{k} \to 0$ as $N \to \infty$, then the competitive ratio is $\omega(1 + o(1))$. □

3.7 Algorithms for the Special Case of Group Bias

Recall that under group bias, elements are partitioned into disjoint groups $G_1, \ldots, G_g$, and the observed scores are $\widetilde{w}(a) = w(a)/\beta_j$ for $a \in G_j$ (see Ex. 3.1). These group-specific bias factors imply comparability within each group and incomparability between groups. We can thus cast group bias as a special case of poset bias, wherein the poset is composed of disjoint chains, and the width is the number g of groups. Since g is known in advance, Corollary 3.1 implies that ProxyLabel is $(g+1)\mathrm{e}^2$-competitive, which is an improvement over the $2\mathrm{e}^3(2g+1)(1+o(1))$ bound obtained by PosetLabel. We next provide a general framework for extending vanilla k-secretary algorithms to obtain better competitive ratios under group bias (Section 3.7.1), using which we can, e.g., get a $g(1-5(k/g)^{-1/2})^{-1}$-competitive algorithm using an algorithm of [4]. In Section 3.7.2, we exhibit further improvements in performance when scores and group assignments are stochastic.

3.7.1 A framework for extending vanilla algorithms to account for group bias

As mentioned above, our methods for the poset setting can be applied to the group setting as well. However, the structure of the poset in the group setting is quite simple (it is a disjoint union of chains), so we might expect simpler algorithms and better competitive ratios compared to the poset setting. Indeed, there is room for improvement, since ProxyLabel is not tight compared to the lower bound in Proposition 3.2, and AdaThreshold is only tight when $(\log N)/\sqrt{k} \to 0$.

One natural idea is to apply a vanilla k-secretary algorithm on each group, selecting at most k/g elements from each. Since secretary algorithms take the number of elements N as input, a pure parallelization over groups would require advance knowledge of the group sizes. However, one can avoid this issue by only parallelizing phases of the algorithm that do not make use of N. To make this explicit, we define a class of *independent-sample secretary (ISS) algorithms*, which do not use knowledge of N during their selection phases.

Definition 3.10 (independent-sample secretary (ISS) algorithm)**.** A k-secretary algorithm $\mathcal{A}^{\mathcal{C}}(N,k)$ which takes as input the number of elements N and the selection capacity k, and operates on a stream $\mathcal{C}$ of elements, is an *independent-sample secretary (ISS) algorithm* if the following hold for some $p_k \in [0,1]$:

1. $\mathcal{A}$ can be temporally decomposed into a **sampling phase** $\mathcal{A}^{\mathcal{C}}_{\text{sample}}(N,k)$ which returns a sample $S \subseteq \mathcal{C}$, followed by a **selection phase** $\mathcal{A}^{\mathcal{C}\setminus S}_{\text{select}}(S,k)$ on the remaining

Algorithm 5: GROUP-AWARE PARALLELIZATION (GAP) for (adversarial) group bias.

input: ISS algorithm $\mathcal{A}(N, k)$, N, k, number of groups g, stream of elements $\mathcal{C}$

1 Produce a sample S using the sampling phase $\mathcal{A}^{\mathcal{C}}_{\text{sample}}(N, k/g)$

2 Run the selection phase $\mathcal{A}^{\mathcal{C} \cap G_j \setminus S}_{\text{select}}(S \cap G_j, k/g)$ for each $j \in [g]$

elements $\mathcal{C} \setminus S$; and

2. the sampling phase returns the first $\text{Bin}(N, p_k)$ elements and makes no selections.

Note that the sampling phase of an ISS algorithm uses the number N of elements, which prevents us from parallelizing this phase in a straightforward way. To get around this, we run the selection phase on the entire stream (unparallelized) and form group-specific samples by intersecting the unparallelized sample with each group. On the other hand, since ISS algorithms do not make use of N during their selection phases, we are free to directly parallelize these selection phases. For example, the k-secretary algorithm of [4], the matroid[18] secretary algorithm of [45], and the graphic matroid secretary algorithm of [67] can all be viewed as ISS algorithms. The resulting algorithm (Algorithm 5) preserves the competitive ratio of the vanilla algorithm up to a factor of g.

Proposition 3.9. *Let $\mathcal{A}$ be an $f(k)$-competitive ISS algorithm for the vanilla k-secretary problem satisfying ordinal fairness*[19] *with respect to the total order on elements. Then* GAP *(Alg. 5) using $\mathcal{A}$ is $g \cdot f(k/g)$-competitive for the k-secretary problem under group bias. Moreover,* GAP *satisfies ordinal fairness with respect to the group poset.*

Proof. The crux of this proof is that the GAP is *distributionally equivalent* to running the vanilla algorithm on each group separately, with a selection capacity of k/g for each group. More formally, we claim that GAP induces the same sampling and selection probabilities as running $\mathcal{A}^{\mathcal{C} \cap G_j}(|G_j|, k/g)$ independently on each group G_j.

[18] The matroid secretary setting can again be extended to include bias models; e.g., elements can be partitioned into groups. The goal is then to (fairly) select an independent set of maximum true utility.

[19] In this case, ordinal fairness reduces to (i) an increasing probability of selection with respect to increasing rank, and (ii) decisions being made only with respect to the rank of the element rather than observed utility.

To verify this, it suffices to show that sampling probabilities are preserved, since the selection phases are run independently of each other. To that end, note that $\mathbb{P}(a \in S) = p_{k/g}$ regardless of the group membership of a. Thus, the sampling distribution induced by GAP on G_j is the same as that of $\mathcal{A}^{\mathcal{C} \cap G_j}(|G_j|, k/g)$. We can thus analyze each group separately. Letting $\mathsf{Alg}^{\mathcal{C} \cap G_j}(|G_j|, k/g)$ denote the output of $\mathcal{A}^{\mathcal{C} \cap G_j}(|G_j|, k/g)$, we have

$$\mathbb{E}\big[w(\mathsf{Alg}_5)\big] = \sum_{j=1}^{g} \mathbb{E}\Big[w\big(\mathsf{Alg}^{\mathcal{C} \cap G_j}(|G_j|, k/g)\big)\Big] \geq \sum_{j=1}^{g} \frac{\text{OPT}_{k/g}(G_i)}{f(k/g)} \geq \frac{\text{OPT}_{k/g}}{f(k/g)} \geq \frac{\text{OPT}_k}{gf(k/g)} ,$$

where $\text{OPT}_m(G)$ is the sum of the top m scores in group G, and OPT_m is the sum of the top m scores overall. The parallelized algorithm therefore has competitive ratio at most $gf(k/g)$. □

Proposition 3.9 shows that a large class of secretary algorithms can be extended to the group bias setting while suffering only a factor g in competitive ratio. In particular, any tight ISS algorithm (i.e., one whose competitive ratio is $1 + o(1)$ as $k \to \infty$) such as the $(1 - 5k^{-1/2})^{-1}$-competitive algorithm of [4] can be extended to a tight algorithm (one whose competitive ratio is $g\big(1 + o(1)\big)$ as $k \to \infty$) for the group bias setting.

3.7.2 Stochastic group assignment

We end our discussion on group bias by considering a stochastic setting, where each element is assigned independently to a group according to a probability distribution $\mathbf{p} = (p_1, \ldots, p_g)$. Element scores are then sampled independently from the same nonnegative distribution. We assume that group assignments and score assignments are independent together as well.

Since the scores are stochastic, the top k elements will be proportionally distributed between the groups, on average. A natural idea would then be to apply the GAP framework, replacing the uniform quotas with proportional quotas of $p_j k$ for group G_j. To ensure

Algorithm 6: GAPCAP for the stochastic group setting

input: N, k, distribution over groups $\mathbf{p}$, quotas vector s

$S \leftarrow$ first $\lfloor \frac{N}{\mathrm{e}} \rfloor$ elements; $R_j \leftarrow$ top $s_j k$ scores in $S \cap G_j$, for each $j \in [g]$

for $a \in G_j, a \notin S$ **do**

  if $\widetilde{w}(a) \geq \min R_j$ **then**

    if *the current $s_j k$th best element from G_j was sampled* **then** Select a.

    Update R_j to be the $s_j k$ highest scores seen so far from G_j

that the selection probabilities are the same across groups,[20] we will require the probability of an element in the sample to be independent of k (otherwise, the differing selection capacities would result in different selection probabilities). We give an example of how the k-secretary algorithm of [65] can be parallelized and show that the resulting algorithm (GAPCAP) is constant-competitive.

Proposition 3.10. *Under the group bias setting, where groups are assigned according to the distribution* $\mathbf{p}$ *and scores are sampled independently from the same nonnegative distribution,* GAPCAP *on input of a proportions vector* $s = \mathbf{p}$ *is* $2\mathrm{e}(1 + o(1))$*-competitive; additionally, if* $p_j < 1 - \frac{1}{k}$ *for all* $j \in [g]$*, then the competitive ratio is bounded by* $4\mathrm{e}$*. Moreover,* GAPCAP *satisfies OF.*

The factor of e in the competitive ratio comes from bounding the probability that one of the top-scoring elements in a group is selected by the algorithm. Since groups and scores are stochastic, the expected number of the top k elements in group G_j is $p_j k$, which means that hedging is good on average. This is ultimately why we are able to avoid the factor of g in competitive ratio, as explained in more detail in the proof below. As with GAP, one drawback of this algorithm is that it imposes quotas, which limits its applicability.

Proof of Proposition 3.10. Recall that each element is assigned to a group according to the probability distributions $\mathbf{p} = (p_1, \ldots, p_g)$, and the algorithm applies proportional quotas $sk = \mathbf{p}k$. Let OPT_p denote the optimal parallelized solution; that is, the set of the top $p_j k$ elements from group G_j, for $j \in [g]$.

[20] This quality is desired because it allows for a *single* sample to be taken, which is then intersected with each group to obtain group-specific samples according to GAP.

We begin by showing that each element in $\textsc{opt}_p$ is selected with probability at least e^{-1}. To see this, suppose $a \in \textsc{opt}_p$, and without loss of generality, $a \in G_1$. Then a is selected if and only if it arrives after the sample, and the p_1kth best element from G_1 seen so far was in the sample. Note that the probability of any particular element being sampled, given an arrival time in $[1, \ell]$, is $\frac{\lfloor N/\mathrm{e} \rfloor}{\ell}$. Hence,

$$\mathbb{P}(\{a \text{ is selected}\} \mid \{a \text{ arrives at time } \ell\}) = \begin{cases} \frac{\lfloor N/\mathrm{e} \rfloor}{\ell - 1} & \text{if } p_1k\ G_1 \text{ elements have arrived,} \\ 1 & \text{otherwise.} \end{cases}$$

So, letting Alg denote the set of elements returned by GAPCAP, we have that

$$\begin{aligned} \mathbb{P}(\{a \in \mathsf{Alg}\}) &= \sum_{\ell = \lfloor N/\mathrm{e} \rfloor}^{N} \mathbb{P}(\{i \text{ arrives at time } \ell\}) \\ &\qquad \cdot \mathbb{P}(\{p_1k\text{th best } G_1 \text{ element at time } \ell \text{ was sampled}\}) \\ &= \frac{1}{N} \sum_{\ell = \lfloor N/\mathrm{e} \rfloor}^{N} \mathbb{P}(\{p_1k\text{th best } G_1 \text{ element at time } \ell \text{ was sampled}\}) \\ &\geq \frac{\lfloor N/\mathrm{e} \rfloor}{N} \sum_{\ell = \lfloor N/\mathrm{e} \rfloor}^{N} \frac{1}{\ell - 1} \\ &> \frac{\lfloor N/\mathrm{e} \rfloor}{N} \ln\left(\frac{N}{\lfloor N/\mathrm{e} \rfloor}\right) \approx \mathrm{e}^{-1}. \end{aligned}$$

It follows that $\mathbb{E}(w(\mathsf{Alg})) \geq \frac{1}{\mathrm{e}} w(\textsc{opt}_p)$. Finally, letting $\textsc{opt}$ denote the true optimal set of elements, it remains to compare $\textsc{opt}_p$ to $\textsc{opt}$. To that end, suppose $a \in \textsc{opt}$. Then

$$\begin{aligned}
\mathbb{P}(\{a \in \textsc{opt}_p\} \mid \{a \in \textsc{opt}\}) &= \sum_{j=1}^{g} \mathbb{P}(\{a \in G_j\}) \\
&\qquad \cdot \mathbb{P}(\{w(a) \geq p_j k\text{th best } G_j \text{ element}\} \mid \{a \in G_j\}) \\
&= \sum_{j=1}^{g} p_j \mathbb{P}(\{w(a) \geq p_j k\text{th best } G_j \text{ element}\} \mid \{a \in G_j\}) \\
&\geq \sum_{j=1}^{g} p_j \underbrace{\mathbb{P}(\{|G_j \cap \textsc{opt}| \leq p_j k\})}_{\xrightarrow{k\to\infty} \frac{1}{2}} \\
&= \frac{1}{2} - o(1).
\end{aligned}$$

It follows that for any $a \in \textsc{opt}$, we have

$$\begin{aligned}
\mathbb{P}(\{a \in \mathsf{Alg}\} \mid \{a \in \textsc{opt}\}) &\geq \left(\frac{1}{2} - o(1)\right) \mathbb{P}(\{a \in \mathsf{Alg}\} \mid \{a \in \textsc{opt}_p\}) \\
&\geq \left(\frac{1}{2} - o(1)\right) \mathrm{e}^{-1}.
\end{aligned}$$

Hence GAPCAP is $2\mathrm{e}(1 + o(1))$-competitive. Moreover, if $p_j < 1 - \frac{1}{k}$ for all $j \in [g]$, then $\mathbb{P}(\{|G_j \cap \textsc{opt}| \leq p_j k\}) > \frac{1}{4}$ [76], thus resulting in a competitive ratio of $4\mathrm{e}$. □

3.8 Experimental Case Study (Simulation)

In this section, we compare the impact of (1) the vanilla adaptive threshold-based k-secretary algorithm of [4], (2) the parallelized version of the same k-secretary algorithm using GAP, (3) POSETLABEL, and (4) ADATHRESHOLD on a real-world dataset. We develop a prediction model based on training data and use the resulting gender-specific errors to construct a partial ranking of the applicants. Using the model-predicted scores as a basis for the first two algorithms and the partial ranking as a basis for the second two, we compare selection rates across gender.

We use the Aspiring Minds' Employability Outcomes 2015 (AMEO 2015) dataset, which was created by Aspiring Minds, an organization which offers pre-employment assessments. We pre-process the data to only consider job seekers in computer science fields, and to only consider the following attributes: gender, standardized high school test scores, college tier, college GPA, college city tier, age at graduation, and Aspiring Minds test scores. 2212 data points remain after pre-processing, among which a random sample of 1600 is taken to be the training data. The other 612 data points comprise the test data, which includes 166 female and 446 male job seekers.[21]

We take the Aspiring Minds computer programming test score as a proxy for utility ("true scores"), and as such, develop a model for predicting computer programming scores from the other attributes. In particular, we predict computer programming using linear regression.[22]

In the Aspiring Minds dataset, we see that the female and male score distributions are qualitatively similar ($\mu_{\text{female}} = 459.27, \sigma_{\text{female}} = 80.54, \mu_{\text{male}} = 477.99, \sigma_{\text{male}} = 93.613$) and are depicted in Figure 3.6 (top left). In order to compare different algorithms under different distributional settings, we transform the Aspiring Minds dataset in two ways: decrease all female scores by 28.09 points (chosen so that the means differ by $\sigma_{\text{male}}/2$) to create a left-shifted distribution, and increase all female scores by 65.53 points (again, chosen so that the means differ by $\sigma_{\text{male}}/2$) to create a right-shifted distribution. See Figure 3.6 for a depiction of the true and predicted computer programming scores in the data, after training on unshifted (left-most) and shifted (center, right) data.

We generate score intervals for applicants (and thus partial rankings) by statistically comparing true scores and predicted scores by gender in the training data. In particular, suppose the true scores of Group j in the training data are in the vector y, and the predicted scores are $\widehat{y}$. Then observe that $\widehat{y}_{\text{transf.}} := \frac{\sigma_y}{\sigma_{\widehat{y}}}\left(\widehat{y} - \mu_{\widehat{y}}\right) + \mu_y$ has the same mean and standard deviation as y. Now let σ_j denote the standard deviation of $y - \widehat{y}_{\text{transf.}}$, which is a measure

[21] We make this dichotomy here because all candidates in the dataset were labeled as male or female.

[22] We remark that any other machine learning model could have been used to predict performance as well. This is simply an example of a predictive model that could be used in a hiring setting.

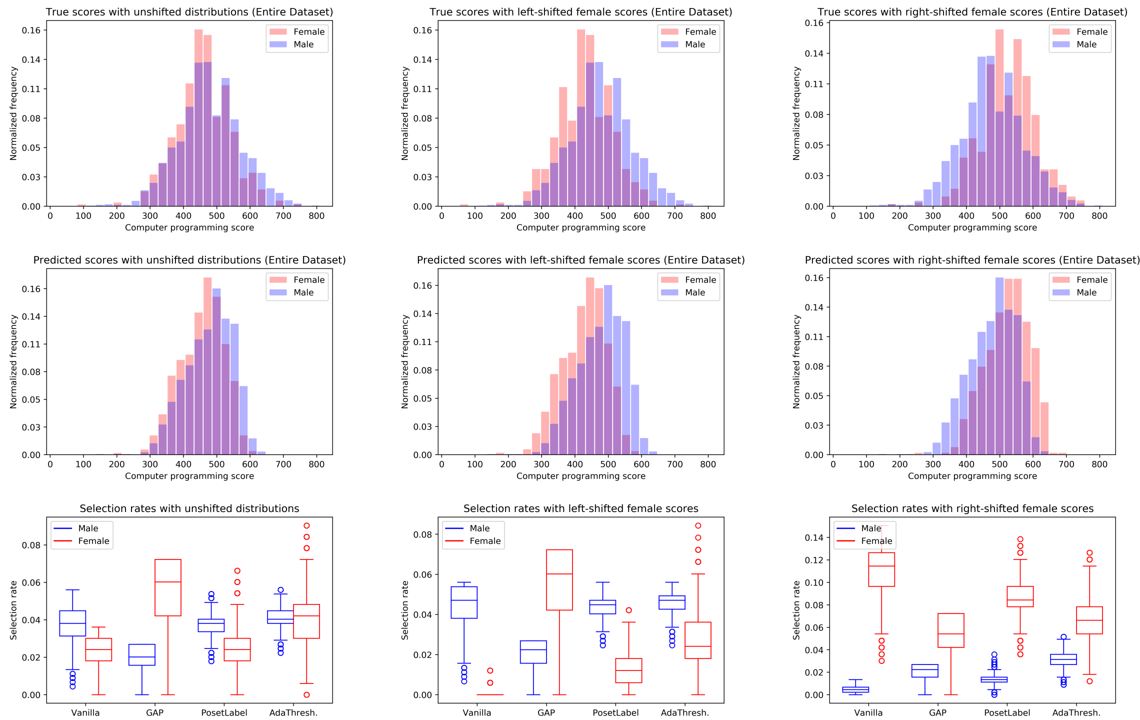


Figure 3.6: Experiment results for three "true score" distributions are shown: the unaltered programming scores (left), the distribution with left-shifted female scores (center column), and the distribution with right-shifted female scores (right). The true score distributions are shown by gender (top), as well as the predicted scores from the regression model (center row) and the selection rates of three algorithms (bottom).

of error in these transformed scores. We can form score intervals in the test data as follows: given a predicted score $\widetilde{w}(a)$ in Group j, we construct the interval

$$\begin{aligned}&\left[\frac{\sigma_y}{\sigma_{\widehat{y}}}\Big(\widetilde{w}(a)-\mu_{\widehat{y}}\Big)+\mu_y-\lambda\sigma_j,\ \frac{\sigma_y}{\sigma_{\widehat{y}}}\Big(\widetilde{w}(a)-\mu_{\widehat{y}}\Big)+\mu_y+\lambda\sigma_j\right]\\&\qquad=[\widehat{y}_{\text{transf.}}(a)-\lambda\sigma_j,\widehat{y}_{\text{transf.}}(a)+\lambda\sigma_j],\end{aligned} \tag{3.9}$$

where $\lambda > 0$ is a parameter controlling the confidence in the score range. Note that under this construction, interval lengths will be uniform within any group, but may differ between groups. We call this the *error-correction approach*.

A total of twelve experiments were performed: the four algorithms[23] were each run on the three distributions discussed above, each of which comprised the 612 applicants in the

[23]The vanilla adaptive-thresholding algorithm of [4] and the GAP algorithm which parallelizes it are both run on the ML-predicted scores. POSETLABEL and ADATHRESHOLD are run using the error-correction approach.

test data. Each of these experiments was given a selection capacity of $k = 25$ and was run 10,000 times (see Appendix H in [53] for the same experiments run with $k = 50$). Since POSETLABEL has an extensive sampling phase compared to the other three algorithms, we scaled its sample size by $1/3$; in this way, the selection rates of POSETLABEL will be large enough to make meaningful comparisons to those of the other algorithms. Finally, we set the λ parameter in (3.9) to be $1/6$, which struck a nice balance between accounting for uncertainty and retaining enough ordinal information (experiments run with $\lambda = 1/3$ and $\lambda = 1/12$ produced qualitatively similar results, which can be found in Appendix H of [53]).

The performance of the vanilla k-secretary algorithm, POSETLABEL, and ADATHRESHOLD is sensitive to the true score distributions in each of the three scenarios: when the female true score distribution is low, fewer female applicants are selected, and when the female true score distribution is high, more female applicants are selected; GAP (parallelizing the algorithm of [4]), on the other hand, uses quotas and is not sensitive to these distributional changes. Use of quotas might have been desirable if demographic parity was the goal, but its lack of sensitivity to scores can be undesirable given the legal backdrop of hiring in the U.S. and given its coarse, non-individualistic nature. POSETLABEL and ADATHRESHOLD instead produce selection rates that are closer to each other (thereby giving some benefit of the doubt to the group with higher uncertainty in their evaluation) compared to the vanilla algorithm as well as GAP. Since both these algorithms use posets derived from errors in prediction, they are better justified from a legal standpoint as well.

Biases enhanced by prediction models. The above experiments showed a natural adaptiveness in selection rates as the underlying data was altered in the three scenarios considered. This adaptiveness can be attributed to the trends in the underlying data captured by the prediction models. However, if the prediction model carries a definitive bias (e.g., a trend not seen in underlying data but picked up by the ML model) against some group, then algorithms making decisions based on the predictive model will be impacted. We

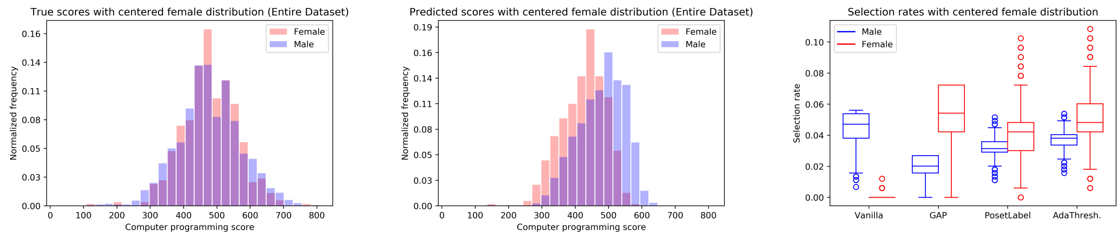


Figure 3.7: The four algorithms are run on centered data (i.e., the true scores of the female group were translated so that both groups have the same mean true score). The regression model was then modified to introduce an additive bias of 50 points against the female group. Shown above are the true score distributions (left), the predicted score distributions (center), and the resulting selection rates (right).

next show that, in fact, when score intervals are constructed as per the error-correction approach (equation (3.9)), such group-specific biases can be corrected using our proposed poset-based algorithms.

To demonstrate this, we artificially bias the prediction model as follows: we first center the underlying data so that both groups have the same mean in the training data, and then fit a linear regression model to predict CS scores. We artificially decrease the gender coefficient in the best fit linear regression model by 50 points, thus introducing a (definitive) additive bias against the female group. We ran the four algorithms on the test data using the shifted predictions.

We find that since the vanilla algorithm has no way of accounting for this additional ML-model bias, it results in a very low selection rate for female applicants. The POSETLABEL and ADATHRESHOLD algorithms, on the other hand, are able to account for this additional ML-model bias using the error-correction approach (eq. (3.9)) and therefore incorporating knowledge of the true scores in the training data. The selection rates produced by these algorithms are consequently more appropriate (given the true score distributions) than that of the vanilla algorithm, as shown in Figure 3.7. This experiment reiterates that the poset-based approaches can be sensitive to changing trends in data and can mitigate the impact of various sources of bias.

3.9 Discussion: Managerial Considerations and Open Questions

Selecting a set of qualified people is an incredibly important yet complex problem in practice. This work is the first to formally consider bias in data using partially ordered sets, and the first to consider bias within the context of the secretary problem. For the general case of partial ordinal rankings, we provided two algorithms: one with an order-optimal competitive ratio (POSETLABEL), and another using adaptive thresholds with a tight competitive ratio (ADATHRESHOLD). These algorithms make selections based solely on the partial rankings and arrival order (thus satisfying ordinal fairness, by Prop. 3.1), which means that properties of the poset (say, those designed to mitigate bias) directly affect selection decisions. We additionally provided simpler algorithms for the special case of group bias, where the poset has simpler structure and vanilla secretary algorithms can be parallelized across the groups. Our key takeaway is the following: *accounting for uncertainty and bias can improve the quality of selected applicants and produce fairer decisions.*

3.9.1 Practical and managerial implications

The necessity for bias-aware methods is born in part out of the disparate amount of errors in evaluations, and our analysis shows that poset-based interventions can provably increase the true total utility of applicants selected. Many of the ideas presented here can be adapted to other data-driven models where there is reason to believe that underlying data is biased. Our algorithmic approaches are also versatile: (1) the random partitioning technique of POSETLABEL can be adapted to a non-algorithmic setting by forming independent review committees, each of which reviews a random subset of the applicant pool. Such a process can reduce the impact of a single reviewer who may dominate a debrief; (2) enforcing diversity in selections (GAP) might be desired for certain applications such as appointment committees in academic review boards; and (3) the technique of adaptive thresholding used in ADATHRESHOLD can be applied generally in practice. For example, in a rolling decisions scenario, where there is no bound on the number of applicants over time, one can periodically update the threshold based on the available data; additionally, if score distributions shift over time, thresholds can be computed using a sliding window to ensure up-to-date selection criteria. These managerial considerations can positively impact hiring practices.

Employers naturally try to avoid litigation, and since large disparities in outcomes can trigger disparate impact lawsuits, outcomes-based constraints such as the 4/5 rule[24] are commonly used [77]. Instead of such quota-based or proportionality constraints, which are coarse and arguably non-individualistic, we propose to account for uncertainty using partial rankings of applicants, the construction of which may or may not use protected information. Our experiments show how one can construct a partial ranking using group-specific errors and how such an intervention can decrease disparities in outcomes without enforcing quotas. The fairness intervention (i.e., the construction of the partial ranking) is modular, separated from the selection process; as such, it can be audited separately from downstream decisions.

Another important managerial takeaway of our work is that the width of the partial ranking affects performance: the more ordinal comparisons one can make, the better the achievable performance is. If the width of a partial ranking is too large, then an employer can seek to reduce the width by investing more resources into (1) gathering more information on individual applicants (e.g., by offering interviews or internships), or (2) improving the evaluation metric.

3.9.2 Open questions

Here, I list some open questions related to

1. Secretary problems operate under many assumptions, such as access to a random sample of applicants and the knowledge of the total number of applicants.[25] Theo-

[24] This rule states that the selection rate of any protected group be at least 4/5 that of any other group.

[25] In some cases, these assumptions can be justified in practice; e.g., data on past applicants could act as

retically speaking though, designing algorithms which avoid rejecting elements "in sample sets" is an open question; in particular, for a fixed poset secretary algorithm, let p_i denote the probability of selecting the ith element to arrive. For fixed k,

what is the asymptotic optimal competitive ratio for an algorithm satisfying $p_1 = \cdots = p_N$, as $N \to \infty$?

This problem remains open for the poset and the classical k-secretary problems. The optimal competitive ratio for *fixed* N in the classical setting has been found using LP-based approach (see Section 2.3 for the main ideas) [47], but asymptotic results are not known.

2. Determining an optimal competitive ratio without the knowledge of N also remains an open question in both the unbiased and biased settings, although [74] showed an N'-dependent lower bound[26] on competitive ratio for the classical 1-secretary problem when N is chosen adversarially from the set $\{1, \ldots, N'\}$.

   What is the optimal competitive ratio for the poset secretary problem when N is unknown but sampled from a known distribution?

3. The capacity constraint considered in this chapter can be viewed as a *uniform matroid constraint*. The design of poset secretary algorithms with general matroidal constraints remains an open area. In particular, let $\mathcal{M}$ be an unknown matroid whose ground set is the set of applicants $\{a_1, \ldots, a_N\}$. Assume that an algorithm, having observed a subset A of applicants, can observe the matroid restricted to A. In this setting,

   find an optimal algorithm for the poset secretary problem with the additional constraint that the set of selections must be independent in $\mathcal{M}$.

the random sample, and time constraints in hiring can upper bound the number of applicants that can be reviewed.

[26] In particular, the lower bound they found was the N'-th harmonic number.

> It may be useful to restrict one's attention to a subset of posets or a subset of matroids. For example, under the group bias model, this problem is simpler: one can enforce group quotas by enforcing a partition matroid constraint; then, to incorporate the other matroid constraint, one can use Svensson and Zenklusen's framework for designing secretary algorithms on the intersection of matroids [78].

Some of our algorithms (GAPCAP and GAP) make use of group-specific selection capacities (i.e., *quotas*). Legally speaking, demographic-aware methods, such as quotas, may or may not be admissible in certain situations [79, 80, 81]. Whether such a disparate treatment can be justified in practice by the inadequacies in data/evaluations is a complex question which I discuss in detail in Chapter 5. In short, if the poset contains only justifiable comparative information amongst candidates and gives no leeway simply due to group membership, one does not have to adhere to quotas to make "fair" selections. We believe such an approach is attractive from a practical, ethical, and legal perspectives as well (e.g., see [82, 83]).

CHAPTER 4
CONSTRAINED ONLINE LEARNING

This chapter contains excerpts from [10] and [84] and includes joint work with Swati Gupta and Vijay Kamble.

In Chapter 3, I discussed an online optimization problem where an adversarially chosen set of inputs are presented to an algorithm in random order to be classified. Decisions in that setting were not associated with feedback; in other words, the information given to the algorithm did not depend on the decisions it made. In this chapter, I consider a setting where feedback is *decision-dependent*. When feedback is available, it may be possible to assess accuracy, FPRs, and other statistical performance measures, which allows for richer possibilities for formulating fairness notions.

I begin, in Section 4.1, by discussing how offline fairness constraints can be adapted to online settings. Next, in Section 4.2, I formulate a dynamic pricing problem to motivate the bandit learning framework discussed in Section 4.3. In Section 4.3, I formulate the main problem discussed in this chapter: *can one converge to the minimum of an N-dimensional function while maintaining comparative fairness between coordinates*? Finally, I end this chapter in Section 4.4 with a discussion of comparative fairness and perceptions of unfairness.

4.1 Memory and Online Decision-Making

A key question in extending static notions of fairness to dynamic settings is that of *memory*: how far back should one look? Suppose, for example, that we wanted to satisfy comparative fairness in probabilities of selection in applicant screening: similar contexts should receive similar decisions, and the same context should always receive the same decision. If our memory extends all the way to the first set of decisions made, then an applicant with context c today must be treated the same as an applicant with context c was treated on day 1. While this would unambiguously satisfy the goal of comparative fairness, it would arguably be too constraining: one bad decision early on can prevent good decisions later on. We categorize temporal fairness notions based on the amount of memory used, as we discuss next.

No memory: fairness within each period. The first and most basic possibility in this regard is to simply ignore the temporal aspect and ensure that the desired static fairness constraints are satisfied independently in each time period. That is, we require that $F_i(x_t, D_{t:t}) \leq 0$ for all $i = 1, \ldots, k$ and all times $t = 1, \ldots, T$, where $D_{t:t}$ is the data from the current time period only (e.g., contexts of the current batch of applicants, not contexts, decisions, or feedback from previous batches). In some situations, this may suffice from a fairness perspective. For instance, it may be acceptable that salaries of women fell due to some event (such as a pandemic) if men's salaries proportionally fell as well to ensure that the salaries are always equitable. From a technical perspective, such fairness constraints can often be employed in practice using standard optimization techniques. For example, assuming that the concerned functions are well-behaved (e.g., convex or concave) and the feasible region is structured (e.g., polyhedral), the theory of online convex optimization can readily handle the problem of optimizing the utility over time while ensuring that the decision rules are feasible at each time. However, in many scenarios, such time-independent notions may be insufficient; e.g., it may be difficult to justify that a small business loan application was approved today, but an applicant with an identical profile was rejected the next day. When consistency across time is desired, *full memory* or *partial memory* notions may be appropriate, as discussed next.

Full memory: fairness across all time. The second possibility lies on the opposite end of the spectrum, requiring that the fairness of a decision must be satisfied with respect to decisions across all times. That is for each $i = 1, \ldots, k$,

$$F_i\big(x_t, D_{1:t}\big) \leq 0 \text{ for all } i = 1, \ldots, k. \tag{4.1}$$

In many cases, full memory constraints are quite stringent (cf. Example 4.1 and Proposition 4.1), since they can disallow any change in decisions over time [6]. Such a requirement is impractical in scenarios where changes in the decision rule may be practically necessary across time, e.g., to learn an unknown utility function; such changes may even be acceptable from a fairness standpoint, e.g., at least *increases* in salaries over time are generally acceptable (and often necessary in view of inflation).

Example 4.1. Consider an applicant-screening scenario in which we must assign a probability of selection $x \in \mathcal{X} = [0, 1]$ to each context $c \in \mathcal{C} \subset \mathbb{R}^N$, and we want our decision rule to be Lipschitz (i.e., to satisfy individual fairness) with respect to some metric d. If we opt for full memory, then decisions made at time t are constrained by decisions made at time 1. If context c was observed at time 1 and a bad decision was made, then decisions made on c in the future must also be sub-optimal. Under modest assumptions, this can lead to poor performance, as shown in Proposition 4.1.

Partial memory. There are several possibilities between these two extremes. For example, one can impose a sliding window constraint, where the fairness constraint can be based on data from the previous m time periods. Similarly, one can impose a time-decay constraint, where recent decisions are weighted more heavily than older decisions. Importantly, this can help prevent the issues discussed in the previous example.

Example 4.2. Suppose we are making cancer diagnosis decisions at a local clinic, and we would like to approximately equalize false negative rates across the disjoint groups C_1 and C_2, where $C_1 \cup C_2 = \mathcal{C}$. However, the air quality in the city in question has degraded over the past several years, and this change is suspected to impact the relationship between contexts and cancer risk. Due to this changing socio-environmental landscape, we want to constrain decisions only based on the previous m decisions. Suppose that the available data $D_{t-m:t} = (c_{t-m}, \ldots, c_t, x_{t-m}(c_{t-m}), \ldots, x_t(c_t), y_{t-m}, \ldots, y_t)$ contains the previous m contexts, the previous m decisions, and the previous m outcomes (whether or not the individual was eventually diagnosed with cancer). In this case, the sliding-window false negative rate for group C_i at time t (i.e., after the tth decision) is

$$\mathrm{FNR}(C_i) = \frac{\sum_{\substack{j=\max\{1,t-m\},\ldots,t \\ c_j \in C_i}} \mathbb{1}[x_j(c_j) = 0,\ y_j = 1]}{|\{\max\{1, t-m\} \le j \le t : c_j \in C_i, y_j = 1\}|}.$$

To approximately enforce equal false negative rates, we require that at each time t,

$$F_1(x_t, D_{\max\{1,t-m\}:t}) = \mathrm{FNR}(C_1) - \mathrm{FNR}(C_2) - \varepsilon \le 0$$
$$F_2(x_t, D_{\max\{1,t-m\}:t}) = \mathrm{FNR}(C_2) - \mathrm{FNR}(C_1) - \varepsilon \le 0.$$

As alluded to in Example 4.1, there can be tensions between fairness and learnability in online learning problems. To quantify this tension, we need a way to quantify the performance of an online learning algorithm. We do so using the notion of *regret*; a non-contextual version of regret was defined in Section 2.2.2, but I extend it to the contextual setting here. Suppose that at time t, we observe context $c_t \in \mathcal{C}$ and choose a decision rule $x_t : \mathcal{C} \to \mathcal{X}$. Given a class of potential cost functions $\mathcal{F}$, the regret up to time T is defined as

$$\sup_{f \in \mathcal{F}} \sup_{c_1, \ldots, c_T} \mathbb{E}\left[\sum_{t=1}^{T} f(c_t, x_t(c_t)) - \inf_{x^*}\left(\sum_{t=1}^{T} f(c_t, x^*(c_t))\right)\right].$$

Simply put, the regret of an algorithm is the worst-case difference between its achieved utility and the optimal utility achieved by a fixed decision rule. Many variants of the above notion of regret exist, including those which replace the left-most supremum with an expectation and those which allow the utility function to change over time. In Example 4.1, if a sub-optimal decision is made on the first context c_1 (i.e., $f(c_1, x_1(c_1)) - f(c_1, x^*(c_1)) = \delta > 0$), then by choosing $c_1 = \cdots = c_T$, we see that the regret up to time T is at least $T\delta = \Omega(T)$, which indicates poor performance. This asymptotic lower bound holds when contexts are generated randomly as well, as long as f and the distribution over $\mathcal{C}$ are nice enough, as shown below in Proposition 4.1.

Lemma 4.1. *Consider the following problem class: the context space is $\mathcal{C} = [0,1]^N$, the decision space is $\mathcal{X} = \mathbb{R}$, and decisions are required to be L_1-Lipschitz. Assume that for each $c \in \mathcal{C}$, the cost function $f(c, \cdot) : \mathcal{X} \to \mathbb{R}$ has a unique minimizer x_c^*, and that the function $c \mapsto x_c^*$ is L_2-Lipschitz. Further, suppose that the contexts $c_1, \dots, c_T$ are sampled independently from a distribution with positive density over $\mathcal{C}$. Now let $|x(c_1) - x_{c_1}^*| = \delta$. Then there is some $\rho = \rho(N, \delta) > 0$ such that for each $t > 1$, $\mathbb{P}(|x(c_t) - x_{c_t}^*| \geq \delta/2) \geq \rho$.*

Proof. First, note that the statement holds trivially if $\delta = 0$, so we may assume that $\delta > 0$. Let $\varepsilon = \frac{\delta}{2(L_1+L_2)} > 0$. Since the distribution over $\mathcal{C}$ has positive density, we have that $\mathbb{P}(\|c_t - c_1\|_2 \leq \varepsilon) \geq \rho > 0$ for all $t > 1$, where ρ depends only on δ and N.

Now suppose that $\|c_t - c_1\| \leq \varepsilon$, for some $t > 1$. Then, since $|x(c_1) - x_{c_1}^*| = \delta$ and $|x(c_t) - x(c_1)| \leq L_1\varepsilon$, we have that $|x(c_t) - x_{c_1}^*| \geq \delta - L_1\varepsilon$. Moreover, since $|x_{c_t}^* - x_{c_1}^*| \leq L_2\varepsilon$, we have that

$$|x(c_t) - x_{c_t}^*| \geq \delta - (L_1 + L_2)\varepsilon = \frac{\delta}{2}.$$

Thus, $\mathbb{P}(|x(c_t) - x_{c_t}^*| \geq \delta/2) \geq \mathbb{P}(\|c_t - c_1\| \leq \varepsilon) \geq \rho$, as desired. □

An immediate consequence of this lemma is that linear regret is unavoidable when enforcing individual fairness with full memory in this setting.

Proposition 4.1 (Full memory IF incurs linear regret)**.** *Consider the following problem class: the context space is $\mathcal{C} = [0,1]^N$, the decision space is $\mathcal{X} = \mathbb{R}$, and decisions are required to be L_1-Lipschitz. Assume that for each $c \in \mathcal{C}$, the cost function $f(c, \cdot) : \mathcal{X} \to \mathbb{R}$ is α-strongly convex, has a unique minimizer x_c^*, and that the function $c \mapsto x_c^*$ is L_2-Lipschitz. Further, suppose that the contexts $c_1, \dots, c_T$ are sampled independently from a distribution with positive density over $\mathcal{C}$. In this setting, any algorithm incurs linear regret.*

Proof. Consider the $N = 1$ where the cost functions are either $f(c, x) = (x - c)^2$ for all $c \in \mathcal{C}$ or $f(c, x) = \big(x - (c + 2)\big)^2$ for all $c \in \mathcal{C}$. In this case, regardless of c_1 and regardless of the decision $x(c_1)$, there is some choice of cost functions for which $|x(c_1) - x^*_{c_1}| \geq 1$. In this case, Lemma 4.1 shows that $\mathbb{P}(|x(c_t) - x^*_{c_t}| \geq 1/2) \geq \rho > 0$ for some constant ρ.

Next, note that by α-strong convexity (Definition 2.7), if $|x(c_t) - x^*_{c_t}| \geq 1/2$, then $f(c_t, x(c_t)) - f(c_t, x^*_{c_t}) \geq \frac{\alpha}{8}$. Thus the regret is at least $\frac{\alpha\rho}{8}(T - 1) = \Omega(T)$. □

There are two ways around this tension between online fairness and learning: first, one can *reduce the memory* of the fairness constraint, thus allowing for more flexibility in adjusting decisions over time. Second, one can *relax the fairness constraint*. Weakening the fairness constraint may allow for learning, even in the full-memory framework.

Example 4.3. Recall from Proposition 4.1 that enforcing individual fairness with full memory can lead to poor performance. In this example, we show how such notions of comparative fairness can be *temporally relaxed*, potentially avoiding this issue. In particular, suppose we have a finite context space $\mathcal{C} = \{c_1, \ldots, c_N\}$ and are tasked with assigning probabilities of selection in $\mathcal{X} = [0, 1]$ for applicant-screening.

If we were to impose individual fairness with full memory, as in Example 4.1, then the decision $x_t(c_i)$ for context c_i at time t is constrained from *above and below* by decisions made at times $1, \ldots, t - 1$. However, a job applicant will only feel mistreated if their decision is worse than expected with respect to decisions made in the past (i.e., if they are rejected and a similar applicant was selected in the past). Motivated by this, we present a one-sided relaxation of comparative fairness which we call *comparative fairness at the time of decision* (CFTD). In particular, given slacks $s(i, j) \in \mathbb{R}$ for $i, j \in [N]$, the constraints imposed at time t are

$$x_t(c_i) \geq x_{t'}(c_j) - s(i, j) \;\text{ for all } t' \leq t \text{ and } i, j \in [N].$$

Importantly, these constraints allow decisions to become more conducive to applicants (i.e., to increase) over time, while ensuring that no context is treated unfairly *relative to decisions made in the past*. On the other hand, since decisions are allowed to increase arbitrarily under this constraint, a decision made on context c_i at time t may be viewed as unfair compared to a decision made at time $t + 10$. The benefit of this weaker form of comparative fairness is that this ability to increase decisions over time allows for learning in some scenarios where the unrelaxed constraint would not, as we illustrate in the next section.

As discussed above, there are important normative questions about the meaning of fairness in dynamic decision-making and technical questions about the trade-offs between temporal notions of fairness and performance. In a given decision-making setting, how can one balance memory and stringency of a fairness constraint with the flexibility required to learn a good decision? These questions pose technically new challenges in online optimization and require new design tools and techniques. In Section 4.3, I illustrate techniques and challenges in the design of temporally fair algorithms through the lens of bandit convex optimization.

4.2 Motivating Example: Multi-Segment Pricing

In this section, I will introduce a dynamic pricing problem which fits in the online learning framework. This problem directly motivates the problem formulated in Section 4.3.1. Suppose that at each time period $t \in [T]$, a good must be priced for N customer segments. Let $x_t = (x_{1,t}, \ldots, x_{N,t}) \in [0, 1]^N$ denote the prices given to the segments at time t.

Upon choosing a price of $x_{i,t} \in [0, 1]$ for Segment i, the decision-maker observes a (noisy) demand of $D_i(x_{i,t}) + \varepsilon$. The corresponding revenue function for segment i is $R_i(x_{i,t}) = x_{i,t} D_i(x_{i,t})$. The total revenue at time t is therefore $R(x_t) = R_1(x_{1,t}) + \cdots + R_N(x_{N,t})$. The goal in most pricing literature is to either estimate the demand function(s) or maximize revenue. We will focus on the goal of maximizing revenue [85, 86, 87].

If the revenue is concave and perfect gradient information is available, then standard optimization techniques (e.g., gradient descent) can be used to maximize revenue. However, we are only given noisy observations of the demand $D_i(x)$, so we can only glean noisy observations of the revenue $R_i(x)$ for each segment i. Thus, when the revenue function is concave, this problem can be framed as a *bandit convex optimization* problem [41], where the function to be minimized is $-R_1(x_1) - \cdots - R_N(x_N)$.

The simplest demand model in economics is the linear model: $D_i(x) = a_i x + b_i$ for some $a_i, b_i \in \mathbb{R}$ with $a_i < 0$. Under this model, the negative revenue function $-R(x)$ is smooth and strongly convex.[1] Thus, a modified gradient descent method, such as that of Hazan and Levy [41], can be used to attain $\widetilde{\mathcal{O}}(\sqrt{T})$ regret.

However, in some cases, one may want to enforce some regularity in decisions across segments. For example, in movie ticket pricing, Segment 1 might be youth, Segment 2 might be adult students, and Segment 3 might be everyone else. In this case, we may want to ensure that $x_1 \leq x_2 \leq x_3$ and $x_3 \leq x_2 + 1$. This would guarantee that the youth price is the lowest, the general adult price is highest, and all adult prices are similar. Such constraints have been studied in the offline setting by Cohen et al. [88]. However, as discussed in Section 4.1, enforcing such constraints in online settings with full memory can lead to high regret. Thus, in the next section, I discuss algorithm design for a temporal relaxation of such constraints.

4.3 Convex Optimization with Bandit Feedback under CFTD

Motivated by the multi-segment pricing problem of Section 4.2, this section is devoted to the design and analysis of bandit convex optimization algorithms satisfying comparative fairness at the time of decision (Definition 4.1)). The problem is formulated in Section 4.3.1, and related work is given in Section 4.3.2. In Section 4.3.3, I present algorithms for one and two groups which use a novel technique which we call *lagged gradient descent*. Finally, in Section 4.3.4, I present a simpler algorithm for the general N-group

[1] In fact, even some demand models whose negative revenue curves are non-convex have the property that the negative revenue is smooth and strongly convex in some ball around the optimum. See Appendix D of [84] for an explanation of this.

setting which yields a worse (but still sublinear) regret bound.

4.3.1 Problem formulation

We now formulate the problem of stochastic convex optimization with bandit feedback under the constraint of ensuring *comparative fairness at the time of decision*. We assume a finite context space $\mathcal{C} = \{c_1, \ldots, c_N\}$. At each time $t \in [T]$, the decision-maker chooses a point $x_t = (x_{1,t}, \ldots, x_{N,t}) \in [x_{\min}, \infty)^N$, where $x_{i,t} = x_t(c_i)$ is the decision on context c_i. The cost incurred on x_t is $f(x_t) = \sum_{i=1}^N f_i(x_{i,t})$.

In this setting, the cost functions f_i are unknown to the principal. The principal hopes to learn to administer the solution to $\min_x f(x)$ under a temporal fairness constraint (described below) over a discrete time horizon T. We make the following assumptions on the cost functions f_i.

Assumption 4.1. We assume that for all $i \in [N]$, the function f_i is

1. α-*strongly convex*, i.e., $f_i(y) \geq f_i(x) + \nabla f_i(x)(y-x) + \frac{\alpha}{2}(y-x)^2$ for all $x, y \in \mathcal{X}$, and

2. β-*smooth*, i.e., $f_i(y) \leq f_i(x) + \nabla f_i(x)(y-x) + \frac{\beta}{2}(y-x)^2$ for all $x, y \in \mathcal{X}$,

for some $\alpha >$ and $\beta > 0$ known to the principal.

These assumptions are commonly made in the convex optimization literature and they allow us to focus on the difficulties that arise in adapting gradient-driven online optimization procedures to satisfy the CFTD constraint.[2]

Comparative fairness at the time of decision (CFTD). As noted in Section 4.1, imposing a comparative fairness constraint can lead to high regret. For example, if we imposed individual fairness (see Section 2.1) in full memory, then we would be forced to choose $x_1 = \cdots = x_T$, since $|x_{i,t} - x_{i,t-1}| \leq L \cdot d(c_i, c_i) = 0$ for each $i \in [N]$, thus preventing

[2] In pricing, the corresponding assumption is that the revenue functions are strongly concave and smooth. This assumption is satisfied under the popular linear demand model that is commonly assumed in the pricing literature [89, 90].

learning. However, a relaxation such as *comparative fairness at the time of decision* allows more room for learning. For convenience, I rephrase this constraint below.

Definition 4.1. A sequence of decisions is said to satisfy *comparative fairness at the time of the decision* (CFTD) if

$$x_{i,t} \geq x_{j,t'} - s(i,j) \text{ for all } i,\ j \in [N],\ \text{and } 1 \leq t' \leq t \leq T. \tag{4.2}$$

The constraints $x_i \geq x_j - s(i,j)$ for all $i, j \in [N]$, in addition to the constraints $x \geq x_{\min}$, form the CFTD polytope $\mathcal{X}_F^N$.

Dynamics and feedback. At each time period t, the algorithm produces a decision $x_t \in \mathcal{X}_F^N$ and observes *bandit* feedback, i.e., the function values for each i at the chosen decision $x_{i,t}$, corrupted with noise. In particular, we assume that the feedback observed is a random variable $Y_{i,t} = f_i(x_{i,t}) + \varepsilon_{i,t}$, where $(\varepsilon_{i,t})_{1 \leq t \leq T}$ for each $i \in [N]$ is a sequence of random variables representing the noise in the feedback.[3] Note that the distribution of $\varepsilon_{i,t}$ can potentially depend on $x_{i,t}$. We make the following commonly made assumption on this sequence.

Assumption 4.2. For each i, and a sequence of decisions $x_{i,1}, \ldots, x_{i,T}$, the random variables $\varepsilon_{i,1}, \ldots, \varepsilon_{i,T}$ are independent, have zero mean, and are sub-Gaussian: there exists a constant $c > 0$ such that $\mathbb{P}(|\varepsilon_{i,t}| \geq s) \leq 2e^{-cs^2}$ for all s. We also assume that they have bounded norm $\max_{i,t} \|\varepsilon_{i,t}\|_{\psi_2} \leq E_{\max}$, where $\|\varepsilon_{i,t}\|_{\psi_2} = \inf\left\{s > 0 : \mathbb{E}[\exp(\varepsilon_{i,t}^2/s^2)] \leq 2\right\}$.

Objective and constraints. The problem is an online decision-making problem, in that the decision x_t at time t is made using the historic information $\mathcal{H}_{t-1} = (x_1, y_1, \ldots, x_{t-1}, y_{t-1})$. These decisions are further constrained to respect the CFTD constraint: $x_{i,t} \geq x_{j,t'} - s(i,j)$ for all $i,\ j \in [N],$ and $1 \leq t' \leq t \leq T$. The regret incurred by an algorithm is

[3] In fact, the algorithms and analysis presented in this section can also accommodate feedback of the form $\sum_{i=1}^{N} Y_{i,t}$.

defined to be

$$\sup_{f\in\mathcal{F}} \sum_{t=1}^{T} \sum_{i\in[N]} \mathbb{E}(f_i(x_{i,t})) - T \min_{x\in\mathcal{X}_F^N} \sum_{i\in[N]} f_i(x_i), \tag{4.3}$$

where the expectation is over the randomness in $(x_t)_{1\le t\le T}$, $\mathcal{F}$ is the set of all α-strongly convex and β-smooth separable functions on $\mathbb{R}^N$, and $\mathcal{X}_F^N$ is the set of decision vectors x that satisfy the CFTD constraints.

4.3.2 Related work

Our work makes fundamental contributions to decision-constrained stochastic convex optimization under bandit feedback. In general, none of the existing algorithms from the convex optimization toolkit can be easily modified to satisfy CFTD. For example, CFTD may require that decisions are monotonically increasing in each dimension as we see in Section 4.3.3. Even this requirement cannot be met by simple modifications of existing algorithms. In the noiseless bandit feedback setting, Kiefer gave an $\mathcal{O}(1)$-regret algorithm, now well known as GOLDEN-SECTION SEARCH, for minimizing a one-dimensional convex function [91]. This algorithm iteratively uses three-point function evaluations to "zoom in" to the optimum, by eliminating a point and sampling a new point in each round. Its mechanics render it infeasible to implement it in a fashion that respects the monotonicity of decisions. For higher dimensions, a $\mathcal{O}(1)$-regret algorithm has been designed by [92]. This algorithm is based on gradient-descent using a one-point gradient estimate constructed by sampling uniformly in a ball around the current point. This key idea recurringly appears in several works on convex optimization with bandit feedback (e.g., [93], [94], [41]). However, due to the randomness in the direction chosen to estimate the gradient, such an approach does not satisfy monotonicity.

For the case of noisy bandit feedback, [95] has shown a lower bound of $\Omega(\sqrt{T})$ in the single-dimensional setting that holds for smooth and strongly convex functions and showed that an appropriately tuned version of the well-known Kiefer-Wolfowitz [96] stochastic approximation algorithm achieves this rate. This algorithm uses two-point function evaluations to construct gradient estimates, which are then utilized to perform gradient-descent. Our near-optimal algorithms in Section 4.3.3 are the most related to this approach, where we tackle the significant additional challenge that the two-point evaluations need to consistently satisfy monotonicity over time while attaining the same regret bounds (up to logarithmic factors). For the case of convex functions, Agrawal et al. designed an algorithm that achieves the $\widetilde{\mathcal{O}}(\sqrt{T})$ bound under bandit feedback [97]. In the one-dimensional case, their approach is the most related to the golden-section search procedure of [91], and as such, is infeasible to implement in a monotonic fashion.

Though not motivated by fairness concerns, two recent parallel works have considered the problem of ensuring monotonicity of decisions in stochastic optimization under bandit feedback ([98] and [99]).[4] Assuming that the functions are only known to be Lipschitz and unimodal they find that the optimal achievable regret is $\widetilde{\Theta}(T^{3/4})$, which is higher than the $\widetilde{\Theta}(T^{2/3})$ regret achievable without the monotonicity requirement. In contrast, Theorem 4.2 implies that under the assumption that the cost functions are smooth and strongly convex, the unconstrained optimal regret bound of $\widetilde{\mathcal{O}}(\sqrt{T})$ is attainable while ensuring monotonicity of decisions (up to logarithmic terms). We also note that, because the settings considered by these other works are more pessimistic in their view of the possible cost functions, algorithm-design for optimizing the worst-case turns out to be simpler. In particular, both [98] and [99] show that we cannot do better than the simple approach of sequentially traversing the decision space in a fixed direction and using a fixed step size until the utility keeps increasing (as determined by a sequence of two-point hypothesis tests). In contrast, the problem of leveraging gradient information while maintaining monotonicity, while also ensuring negligible impact on regret, results in several novel and non-trivial aspects of our algorithm design.

[4] It may be worth mentioning that the first public drafts of both these works appeared after the first public draft of an earlier version of our paper [100], although the research was conducted in parallel.

4.3.3 Algorithm design and analysis for $N = 1, 2$ groups

Ensuring CFTD imposes complex constraints across decisions over time. In this section, we tackle the challenges that arise in designing a near-optimal regret algorithm for the simplest case of two population groups. First, let's examine what the CFTD constraints require in this setting. These constraints can be broken down into two parts:

1. **Coordinate-wise monotonicity.** We first require that $x_{i,t} \geq x_{i,t'}$ for $1 \leq t' \leq t \leq T$ and for $i = 1, 2$, which means that the decisions on each coordinate must (weakly) increase over time.

2. **Cross-coordinate constraints.** We also have that $x_{1,t} \geq x_{2,t'} - s(1,2)$ and $x_{2,t} \geq x_{1,t'} - s(2,1)$ for all $1 \leq t' \leq t \leq T$, which means that the decision for group i at any time can be at most $s(i,-i)$ less than the highest decision seen by group $-i$ so far.[5]

Because several ideas will be necessary for designing a near-optimal regret algorithm that satisfies these constraints with just bandit feedback, we break down this problem into parts. First, we will consider a single-group online stochastic optimization problem where the only constraint is to satisfy weak monotonicity. The algorithmic approach we develop for that case will then be used as a component in the two-group case where we additionally have to tackle the cross-coordinate constraints.

Single group monotonic stochastic optimization with bandit feedback.

In this section, we consider the problem of minimizing a smooth and strongly convex function $f(\cdot)$ over the feasible set $\mathcal{X}$ under noisy bandit feedback, while ensuring that the decisions monotonically (weakly) *increase* over time and while ensuring low regret. We denote $x^* = \arg\max_{x \in \mathcal{X}} f(x)$. As we noted earlier in Section 4.3.2, none of the existing stochas-

[5] We adopt the convention of "Group $-i$" indicating the group other than Group i. So, Group -1 refers to Group 2, and Group -2 refers to Group 1.

Algorithm 7: Lagged Gradient Descent (LGD) (noiseless bandit)

input: strong convexity parameter α, smoothness parameter β, time horizon T, $x_{\min}$

Set $x'_1 \leftarrow x_{\min}$ and $x_1 \leftarrow x'_1 + \delta$, and observe $f(x'_1)$ and $f(x_1)$

for $t = 1, \ldots, T/2$ **do**

  Let $\widetilde{\nabla}_t \leftarrow \frac{f(x_t) - f(x'_t)}{x_t - x'_t}$

  if $-\frac{1}{\beta}\widetilde{\nabla}_t \geq (1+\gamma)\delta$ **then**

    Sample $f(\cdot)$ at $x'_{t+1} = x'_t - \frac{1}{\beta}\widetilde{\nabla}_t - \delta_{t+1}$ (lagged iterate)

    Sample $f(\cdot)$ at $x_{(t+1)} = x'_{t+1} + \delta_{t+1}$ (non-lagged iterate)

  else

    Exit from loop and stabilize at x_t

tic optimization algorithms designed for this setting satisfy monotonicity.

Algorithmic approach. At their core, our algorithm relies on gradient estimates constructed from monotonic two-point function evaluations (i.e., "secant" information) to improve decisions as in the well-known Keifer-Wolfowitz algorithm for convex optimization with noisy bandit feedback [96]. Due to the monotonicity constraint, the challenge is to ensure that sufficient progress is continually made towards reaching the optimal decision using gradient information while avoiding excessively overshooting the optimum (since backtracking is disallowed). There are two key algorithmic ideas we develop to tackle this challenge: (a) taking gradient steps from a "lagged" point to avoid overshooting the optimum, and (b) adapting the lag size to local gradient estimates to tailor the degree of caution to the distance from the optimum point, while ensuring monotonicity.

Warm-up: the case of noiseless bandit feedback. We first consider algorithm design in the noiseless bandit setting: a decision $x_t \in \mathcal{X}_F^1 = [x_{\min}, \infty)$ is made at each time $t \in [T]$, and for each decision x_t, the algorithm observes $f(x_t)$. The decisions made by the algorithm are constrained to be monotone: $x_1 \leq x_2 \leq \cdots \leq x_T$.

We present a monotonic procedure called LAGGED GRADIENT DESCENT (LGD) (Algorithm 7) for this setting, which is a variation on classical gradient descent. At each round, we use two queries (one at x_t and one at the "lagged" point $x'_t = x_t - \delta$) to estimate the gradient. Since we need to ensure monotonicity of iterates, we need to sample first at the lagged point x'_t and next at x_t to get an estimate of the gradient. The size of δ will depend on the time horizon T and be optimized for minimizing the regret of the overall scheme. We then move by an amount proportional to the estimated gradient.

While increasing decisions, even in this non-noisy case, we need to be careful about not excessively overshooting the optimum point, which could result in high regret due to the monotonicity constraint that disallows backtracking. To avoid overshooting, we move proportional to the gradient *from the lagged point x'_t instead of from x_t*. Since the estimated gradient $\widetilde{\nabla}_t$ in LAGGED GRADIENT DESCENT is less steep than the true gradient at x'_t, the smoothness of f allows us to ensure that we never overshoot. However, since we jump from x'_t instead of x_t, a small jump (of a magnitude smaller than $|x_t - x'_t|$) may violate monotonicity. To avoid this, we jump forward only if the gradient is steep enough; in particular, if the magnitude of the estimated gradient is at least $\beta(1+\gamma)\delta$, for some $\gamma > 1$, defined in Theorem 4.1.

Lemma 4.2 below shows that the decisions resulting from LAGGED GRADIENT DESCENT (LGD) are monotonic, they avoid overshooting, and their convergence rate to the optimum is exponential.

Lemma 4.2. *Let $f : \mathcal{X}_F^1 \to \mathbf{R}$ be an α-strongly convex and β-smooth function. Let $x_1, \ldots, x_{T/2}$ be the non-lagged points generated by* LAGGED GRADIENT DESCENT *(Algorithm 7), and assume that $x_1 \le x^* = \arg\min_{x\in\mathbb{R}} f(x)$. Then, for $\gamma > 1$, the following hold:*

1. *Decisions increase monotonically toward the optimum: $x_1 \le x_2 \le x_3 \le \cdots \le x_{T/2} \le x^*$;*
2. *The convergence rate is exponential up to halting: $h_{t+1} \le h_1 \exp\big(-2\alpha c t\big)$, where $h_t = f(x_t) - f(x^*)$ and $c = \frac{1}{2\beta} - \frac{1}{(1+\gamma)\beta}$.*

Proof. We begin by proving "1." To show that we never overshoot, we will exploit smoothness. In particular, for any t such that $x_t \leq x^*$, we have

$$\begin{aligned}
x_{t+1} - x_t' &= -\frac{1}{\beta}\widetilde{\nabla}_t = -\frac{1}{\beta}\nabla f(\overline{x_t}) && \text{for some } \overline{x_t} \in [x_t', x_t] \\
&= \frac{1}{\beta}\|\nabla f(\overline{x_t})\| && \text{since } x_t \leq x^* \\
&\leq x^* - \overline{x_t} && \text{assuming } \nabla f(x^*) = 0 \\
&\leq x^* - x_t' \text{ .This proves “1.”}
\end{aligned}$$

To show the convergence (“2”), first note that by α-strong convexity, we have that $f(y) \geq f(x) - \frac{1}{2\alpha}\|\nabla f(x)\|^2$. For $y = x^*$, this becomes:

$$\|\nabla f(x)\|^2 \geq 2\alpha\big[f(x) - f(x^*)\big] \;. \tag{4.4}$$

Now we wish to bound the gap $h_t = f(x_t) - f(x^*)$. For any $t \geq 2$, we have

$$\begin{aligned}
h_{t+1} - h_t = f(x_{t+1}) - f(x_t)& \\
&\leq \nabla_t^\top(x_{t+1} - x_t) + \frac{\beta}{2}(x_{t+1} - x_t)^2 && \beta\text{-smooth} \\
&\leq (\widetilde{\nabla}_t^\top + \beta\delta)(x_{t+1} - x_t) + \frac{\beta}{2}(x_{t+1} - x_t)^2 && \text{Lemma 2.1} \\
&= -\frac{1}{2\beta}\|\widetilde{\nabla}_t\|^2 - \delta\widetilde{\nabla}_t - \frac{\beta}{2}\delta^2 && \text{definiton of } x_{t+1} \\
&\leq -\frac{1}{2\beta}\|\widetilde{\nabla}_t\|^2 - \left(-\frac{\widetilde{\nabla}_t}{(1+\gamma)\beta}\right)\widetilde{\nabla}_t - \frac{\beta}{2}\delta^2 && \text{since } -\tfrac{1}{\beta}\widetilde{\nabla}_t \geq (1+\gamma)\delta \\
&= -\left(\frac{1}{2\beta} - \frac{1}{(1+\gamma)\beta}\right)\|\widetilde{\nabla}_t\|^2 - \frac{\beta}{2}\delta^2 \\
&\leq -\underbrace{\left(\frac{1}{2\beta} - \frac{1}{(1+\gamma)\beta}\right)}_{=:c}\|\widetilde{\nabla}_t\|^2.
\end{aligned}$$

By the mean value theorem, there is some $\overline{x_t} \in [x_t', x_t] \subset [x_{t-1}, x_t]$ such that $\nabla f(\overline{x_t}) = \widetilde{\nabla}_t$. Using 4.4, we get: $h_{t+1} - h_t \leq -c\|\widetilde{\nabla}_t\|^2 = -c\|\nabla f(\overline{x_t})\|^2 \overset{4.4}{\leq} -2\alpha c[f(\overline{x_t}) - f(x^*)] \leq$

$-2\alpha c h_t$. Note that for $\gamma > 1$ as specified in the lemma, $2\alpha c \in (0,1)$. So,

$$h_{t+1} \leq \big(1 - 2\alpha c\big) h_t \leq \cdots \leq \big(1 - 2\alpha c\big)^t h_1 \leq h_1 \exp\big(-2\alpha c t\big) .$$

□

With this convergence rate established, we can calculate a regret bound for LGD. We can break the regret of this procedure into three categories: regret during exploration, regret due to stopping (i.e., regret incurred after the "for loop" has ended), and regret due to potential overshooting (which is 0 by Lemma 4.2). We balance these three to obtain an $\mathcal{O}(1)$ regret bound.

Theorem 4.1. *Assume that* $x^* = \arg\min_{x \in \mathbf{R}} f(x) \in (x_{\min}, \infty)$*, and fix* $\delta = T^{-1/2}$ *and* $\gamma = 1 + \frac{1}{\log T}$*. Then* LAGGED GRADIENT DESCENT *(Alg. 7) is a* $\mathcal{O}(1)$*-regret CFTD algorithm for optimizing an* α*-smooth and* β*-strongly convex function in the noiseless bandit setting.*

Proof. As stated in the theorem, fix $\delta = T^{-1/2}$ as the lag size. Since overshooting never occurs (by Lemma 4.2), we need only calculate the exploration and stopping regret. By Lemma 4.2, the exploration regret is bounded by $2\sum_{t=1}^{\infty} h_1 \exp\big(-2\alpha c(t-1)\big)$, which is constant.

Now we analyze the stopping regret. If the algorithm stops at some time t, then it must be that

$$-\frac{1}{\beta}\widetilde{\nabla}_t \leq (1+\gamma)\delta ,$$

which allows us to bound the gradient: $\|\nabla_t\| \leq \|\widetilde{\nabla}_t\| \leq (1+\gamma)\delta$. In other words, if we stop at time t, then $\|\nabla_t\| \in \mathcal{O}(\delta)$, for $\gamma = 1 + \frac{1}{\log T}$. When $\|\nabla_t\| \leq 3\delta$, Lemma 2.1 tells us

Algorithm 8: Adaptive Lagged Gradient Descent (ADA-LGD)

input: convexity parameters α, β, time horizon T, $x_{\min}$, initial lag size δ_1, stopping parameter $\gamma > 1$, $q \in (0,1)$, noise sub-Gaussian norm bound $E_{\max}$, Hoeffding constant C

$\delta_i \leftarrow q^{i-1}\delta_1$ for $i \geq 2$, $\xi \leftarrow 1-q$, $n(d) = \frac{64E_{\max}^2 \log\frac{2}{p}}{C\alpha^2 d^4}$ for any d, $x_1 \leftarrow x_{\min} + \delta_1$, $t \leftarrow 1$, $i \leftarrow 0$

repeat

  $i \leftarrow i - 1$

  repeat

    $i \leftarrow i + 1$

    $\overline{f}(x_t - \delta_i) \leftarrow$ average of $n(\xi\delta_i)$ samples at $x_t - \delta_i$ `// estimate` $f(x_t - \delta_i)$

    $\overline{f}(x_t - \delta_{i+1}) \leftarrow$ average of $n(\xi\delta_i)$ samples at $x_t - \delta_{i+1}$ `// estimate` $f(x_t - \delta_{i+1})$

    $g_t^{(i)} \leftarrow \frac{\overline{f}(x_t-\delta_{i+1}) - \overline{f}(x_t-\delta_i) + \alpha\xi^2\delta_i^2/4}{\xi\delta_i}$ `// compute the approximate secant`

  until $-\frac{1}{\beta}g_t^{(i)} \geq (2+\gamma)\delta_i$

  $\overline{f}(x_t) \leftarrow$ average of $n(\delta_i)$ samples at x_t `// estimate` $f(x_t)$

  $\widetilde{\widetilde{\nabla}}_t \leftarrow \frac{\overline{f}(x_t) - \overline{f}(x_t-\delta_i) + \alpha\delta_i^2/4}{\delta_i}$ `// compute the approximate secant`

  Compute $x_{t+1} \leftarrow x_t - \frac{1}{\beta}\widetilde{\widetilde{\nabla}}_t - \delta_i$

  $t \leftarrow t + 1$

until *T samples have been taken*

that $\|x_t - x^*\| \leq 3\delta/\alpha$, and so $f(x_t) - f(x^*) \in \mathcal{O}(\delta^2)$ by β-smoothness. Hence,

$$\text{regret} \leq \underbrace{2\sum_{t=1}^{\infty} h_1 \exp\big(-2\alpha c(t-1)\big)}_{\in\mathcal{O}(1)} + \underbrace{T\delta^2}_{\in\mathcal{O}(1)} .$$

So, we get a regret of $\mathcal{O}(1)$. □

Theorem 4.1 shows that imposing monotonicity has no (asymptotic) effect on the hardness of the noiseless setting: a non-monotonic constant-regret procedure is already known [91], and LGD is a *monotonic* constant-regret procedure.

One important aspect of the noiseless setting is that gradient estimation is easy: one simply requires two samples, and additionally, the gap δ between the samples can be arbitrarily small, resulting in arbitrarily accurate gradients. As we will discuss in the next section, when noise is introduced, there is a trade-off between the magnitude of δ and the number of samples required to accurately estimate the gradient. This tension ultimately results in a higher regret bound.

The challenges posed by noisy bandit feedback. Recall that in the noisy bandit setting, upon querying the t^{th} point x_t, the algorithm observes $f(x_t) + \varepsilon_t$, where the noise ε_t is assumed to be independent, mean zero, and sub-Gaussian of bounded sub-Gaussian norm (Assumption 4.2). To appreciate the complication added by the noise, let's suppose we try to replicate the approach in LAGGED GRADIENT DESCENT, in which we utilize secant calculations with a fixed lag size δ to estimate the gradient step. In the absence of noise, the secant is always sandwiched between the true gradients at x_t and the lagged point $x_t - \delta$ by the mean value theorem. Thus moving from the lagged point using the secant ensures that the algorithm never overshoots the optimum. But when the function evaluations are noisy, multiple function evaluations are necessary to evaluate the secant accurately enough for this sandwich property to hold. In particular, one can show that $\Theta(1/\delta^4)$ function evaluations are necessary at the two points to get such an accurate secant estimate (Lemma 2.3). So, if one uses $\delta = T^{-1/2}$ as in the analysis of noiseless LGD (Theorem 4.1), the number of samples required at each point is $\Omega(T^2)$. Such high sampling rates may be acceptable when the algorithm iterates are very close to the optimum, but will lead to a high (linear) regret when the iterates are farther from the optimum. Thus, clearly, δ cannot be set to be that small.

At the same time if δ is too large then the algorithm may stop far from the optimum, since the local gradient may prematurely become small enough relative to the lag size that jumping from a lagged point may violate monotonicity (i.e., the condition $-\frac{1}{\beta}g < (1+\gamma)\delta$ is satisfied). Such premature stopping will again cause high regret. Later in Section 4.3.4, we will see that one can set a fixed δ that optimizes this tradeoff and obtain a regret of $\widetilde{\mathcal{O}}(T^{2/3})$ in the multi-group setting.

However, it turns out that we can do strictly better and obtain the near-optimal regret rate of $\widetilde{\mathcal{O}}(\sqrt{T})$ by choosing the lag sizes *adaptively*. The key idea is that if the algorithm stops moving with a particular lag size δ, then we reduce the lag size so that the algorithm can continue to proceed. This ensures that smaller lag sizes and correspondingly higher sampling rates are utilized only when the iterates are closer to the optimum when they do not result in high regret.

This approach, however, presents a crucial challenge: when estimating the secant at a point x_t and a lagged point $x_t - \delta$, the decision of whether the lag size must be reduced from δ to some smaller quantity must be made *before* we sample at x_t to insure monotonicity of the iterates. To address this challenge, we design a novel algorithm that respects monotonicity while searching for the correct lag size.

Adaptive Lagged Gradient Descent. We develop a novel procedure called ADAPTIVE LAGGED GRADIENT DESCENT (ADA-LGD) (Alg. 8) in this section. In this procedure, there are "non-lagged" iterates, denoted as $(x_t)_{t\in\mathbb{N}}$, and "lagged" iterates denoted in relation to the non-lagged iterates, e.g., $x_t - \delta_i$ for some specified i. For any non-lagged iterate x_t such that $x_{t+1} = x_t - \frac{1}{\beta}\tilde{\tilde{\nabla}}_t - \delta_i$, we say that δ_i is the lag size of x_t.

We now describe how ADA-LGD reduces the lag sizes in a monotonic manner. Suppose the current lag size is δ_i, and we are sampling at $x_t - \delta_i$. Right after sampling at $x_t - \delta_i$, we sample at $x_t - \delta_{i+1}$ (where $\delta_{i+1} = q\delta_i$ for some $q < 1$). This has the benefit of providing a gradient estimate at $x_t - \delta_i$. This estimate in turn gives us an estimate of the gradient at x_t, which can be used in deciding whether or not the lag size should indeed be reduced to δ_{i+1} or lower. If yes, then we continue to sample at $x_t - \delta_{i+2}$ and continue the search for the right lag size; else, we finally sample at x_t and continue the secant descent procedure. Such pre-emptive sampling to search for the correct lag size thus guarantees monotonicity.

In view of these dynamics, it is useful to think of the iterates with the same lag size as forming a *phase*, where phases are numbered chronologically: Phase 1 is the phase containing x_1, Phase 2 is the next phase, and so on. The lag size associated to Phase j is denoted δ_{n_j}. Since multiple lag sizes can be skipped between x_t and x_{t+1}, it may be the case that $n_{i+1} > n_i + 1$. For example, see Figure 4.1, where the first jump is taken with a lag size of $\delta_2 = q\delta_1$. While sampling at $x_2 - \delta_2$ and $x_2 - \delta_3$, since the estimated gradient is not steep enough, the algorithm begins sampling at $x_2 - \delta_4$. The algorithm decreases the

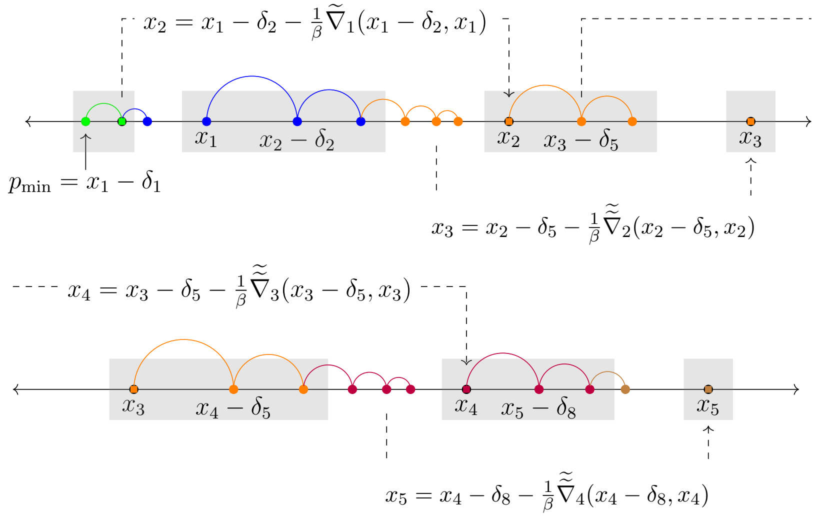


Figure 4.1: Illustration of the points: the algorithm starts exploring at $p_{\min} = x_1 - \delta_1$ followed by $x_1 - \delta_2$. In this case, Phase 1 consists of x_1, Phase 2 consists of x_2 and x_3, Phase 3 consists of x_4, and Phase 4 consists of x_5; the step-size indices are $n_1 = 2, n_2 = 5, n_3 = 8$, and $n_4 = 9$. The computation of x_{t+1} is given by approximate gradient from the *chosen* lagged point, as depicted by the dotted lines, using the estimate $\tilde{\tilde{\nabla}}_t(x_t - \delta_i, x_t)$ obtained by sampling at $x_t - \delta_i$ and x_t.

lag size twice more before deciding that δ_5 is an appropriate lag size. In this case, $x_2 - \delta_5$ is the "chosen" lagged point. Theorem 4.2 formalizes the regret guarantee achieved:

Theorem 4.2. *Assume that $x^* = \arg\min_{x\in\mathbb{R}} f(x) \in (x_{\min}, \infty)$, and assume the noise is mean zero, independent, and sub-Gaussian of bounded sub-Gaussian norm (Assumption 4.2). Then* ADA-LGD *(Algorithm 8) satisfies CFTD and, on input of $\delta_1 = 1/\log T$, $\gamma = 1 + \frac{1}{\log T}$, any $q \in (0, 1)$, and $p = T^{-2}$, incurs regret of order $\mathcal{O}\left((\log T)^2 T^{1/2}\right)$.*

There are several steps in the proof of Theorem 4.2, which is presented below. We first show that all the secant estimates are accurate enough with high probability so that the sandwich property holds, i.e., the secant is indeed sandwiched between the gradients at the two points (Claim 4.1 in the proof). With this established, we show that overshooting does not occur (Claim 4.2), that the iterates are monotonic (Claim 4.3), and that we achieve exponential convergence to the optimum (Claim 4.4) (again, with high probability) across all the iterates, regardless of lag size. The challenging part is to bound the regret from non-lagged and lagged iterates, which we address separately. For each source of regret, we derive a key phase-dependent regret bound that is inversely proportional to the local gradient in that phase; this bounds the regret from the early phases (Claims 4.5 and 4.6). We finally bound the regret from the later phases by leveraging the fact that the iterates are closer to the optimum. Balancing regret between early and late phases yields the $\widetilde{\mathcal{O}}(T^{1/2})$ regret bound (Claim 4.7).

Proof of Theorem 4.2. For this proof, we introduce some additional notation for ease of exposition. Following our notion of *phases*, we can uniquely associate to each x_t a pair (s, i) such that x_t is the sth iterate in the ith phase; we denote such an iterate as $x_t = y_s^{(i)}$. So, e.g., in Figure 4.1, we see that $x_1 = y_1^{(1)}, x_2 = y_1^{(2)}, x_3 = y_2^{(2)}, x_4 = y_1^{(3)}$ and $x_5 = y_1^{(4)}$. We now proceed with the proof.

We break the proof into several claims. We refer to the estimated gradient at the δ_i-lagged point as $g_t^{(i)} = \frac{\overline{f}(x_t-\delta_{i+1})-\overline{f}(x_t-\delta_i)+\varepsilon(\xi\delta_i)}{\xi\delta_i}$ and the probability parameter of the algorithm as $p = T^{-2}$.

Claim 4.1 (gradient accuracy). Let $\widetilde{\widetilde{\nabla}}_t = \frac{\overline{f}(x_t)-\overline{f}(x_t-\delta_i)+\varepsilon(\delta_i)}{\delta_i}$ be the estimated secant at epoch t, and let $g_t^{(i)} = \frac{\overline{f}(x_t-\delta_{i+1})-\overline{f}(x_t-\delta_i)+\varepsilon(\xi\delta_i)}{\xi\delta_i}$, where $\delta_i = q^{i-1}\delta_1$ is the i^{th} lag size and $\xi = 1 - q$. Then these gradient estimates of the algorithm satisfy

$$\widetilde{\widetilde{\nabla}}_t \in [\nabla f(x_t - \delta_i), \nabla f(x_t)] \quad \text{and} \quad g_t^{(i)} \in [\nabla f(x_t - \delta_i), \nabla f(x_t - \delta_{i+1})]. \tag{4.5}$$

each with probability at least $(1-p)^2$, where $p = T^{-2}$.

Note that for this choice of p, we have that $(1-p)^T \to 1$. This claim follows immediately from Lemma 2.3 since each of the estimates is constructed by sampling $\frac{4E_{\max}^2 \log \frac{2}{p}}{C\varepsilon(d)^2}$ times, where $\varepsilon(d) = \alpha d^2/4$ and d is the gap between the two points at which we are sampling (i.e., $d = \delta_i$ in the case of $\widetilde{\widetilde{\nabla}}_t$ and $d = \xi\delta_i$ in the case of $g_t^{(i)}$).

Claim 4.2 (overshooting). Assuming (4.5) holds for all estimated gradients,[6] and that $x_1 = x_{\min} + \delta_1 < x^*$, then all the iterates $x_1, x_2, \ldots x_k$ (for $k \leq T$) generated by the lagged

[6]This happens with probability at least $(1-p)^T$.

secant movements in the outer loop of the algorithm do not overshoot the optimum; that is, $x_t \leq x^*$ for all $t \leq k$. Note that this implies that all lagged points sampled by the algorithm (i.e., those sampled *between* x_t and x_{t+1}, for some t) also do not overshoot.

PROOF OF CLAIM 4.2. We show this by induction, where the base case follows from assumption that $x_1 \leq x^*$. Suppose $x_t \leq x^*$, and that x_{t+1} was chosen based on a lag size of δ_i. In other words, $x_{t+1} = x_t - \frac{1}{\beta}\widetilde{\widetilde{\nabla}}_t - \delta_i$, where $\widetilde{\widetilde{\nabla}}_t = \frac{\overline{f}(x_t) - \overline{f}(x_t - \delta_i) + \varepsilon(\delta_i)}{\delta_i}$ and $\varepsilon(\delta_i) = \alpha\delta_i^2/4$. By (4.5) and the mean value theorem, $\widetilde{\widetilde{\nabla}}_t = \nabla f(\overline{x}_t)$ for some $\overline{x}_t \in [x_t - \delta_i, x_t]$. So,

$$
\begin{aligned}
x_{t+1} - (x_t - \delta_i) &= -\frac{1}{\beta}\widetilde{\widetilde{\nabla}}_t = -\frac{1}{\beta}\nabla f(\overline{x}_t) && \text{for some } \overline{x}_t \in [x_t - \delta_i, x_t] \\
&= \frac{1}{\beta}|\nabla f(\overline{x}_t)| && \text{since } x_t \leq x^* \\
&\leq x^* - \overline{x}_t && \text{since } \nabla f(x^*) = 0 \text{ and } f \text{ is smooth} \\
&\leq x^* - (x_t - \delta_i).
\end{aligned}
$$

This proves Claim 4.2. $\square_{\text{Claim 4.2}}$

Claim 4.3 (monotonicity). Assuming (4.5) holds for all estimated gradients, and that $x_1 < x^*$, samples taken by the algorithm (including lagged and non-lagged iterates) are non-decreasing.

PROOF OF CLAIM 4.3. Again, suppose that x_{t+1} was chosen based on a lag size of δ_i. In other words, $x_{t+1} = x_t - \frac{1}{\beta}\widetilde{\widetilde{\nabla}}_t - \delta_i$, where $\widetilde{\widetilde{\nabla}}_t \leftarrow \frac{\overline{f}(x_t) - \overline{f}(x_t - \delta_i) + \varepsilon(\delta_i)}{\delta_i}$ and $g_t^{(i)} = \frac{\overline{f}(x_t - \delta_{i+1}) - \overline{f}(x_t - \delta_i) + \varepsilon\left(\xi\delta_i\right)}{\xi\delta_i}$, where $\delta_i = q^{i-1}\delta_1$ is the ith lag sizes and $\xi = 1 - q$. Note that $\xi\delta_i = (x_t - \delta_{i+1}) - (x_t - \delta_i)$ is the domain gap between $x_t - \delta_i$ and $x_t - \delta_{i+1}$. To show monotonicity, we need to show that the next *lagged* point exceeds the current point; i.e., we must show that $x_t \leq x_{t+1} - \delta_i$.

Note that the lagged step is taken only for the first δ_i that achieves $-\frac{1}{\beta}g_t^{(i)} \geq (2 + \gamma)\delta_i$.

So,

$$
\begin{aligned}
-\frac{1}{\beta}\tilde{\tilde{\nabla}}_t &\geq -\frac{1}{\beta}\nabla f(x_t) && \text{by grad. bounds (4.5)} \\
&\geq -\frac{1}{\beta}\left(\nabla f(x_t - \delta_i) + \beta\delta_i\right) && \beta\text{-smoothness} \\
&\geq -\frac{1}{\beta}\left(g_t^{(i)} + \beta\delta_i\right) && \text{by grad. bounds (4.5)} \\
&\geq (1+\gamma)\delta_i && \text{by assumption.}
\end{aligned}
$$

Since $\gamma > 1$ by assumption, we have that $(x_{t+1}-\delta_i)-x_t = -\frac{1}{\beta}\tilde{\tilde{\nabla}}_t - 2\delta_i \geq (1+\gamma)\delta_i - 2\delta_i > 0$. In other words, we do not break monotonicity. This proves Claim 4.3. $\square_{\text{Claim 4.3}}$

Claim 4.4 (convergence rate). Assume (4.5) holds for all estimated gradients, and that $x_1 < x^*$. Now let $h_t = f(x_t) - f(x^*)$ be the instantaneous regret at x_t. Then $h_{t+1} \leq h_1 \exp(-2\alpha c t)$, where $c = \frac{1}{2\beta} - \frac{1}{(1+\gamma)\beta}$. In particular, letting $h_t^{(i)} = f(y_t^{(i)}) - f(x^*)$ be the instantaneous regret at the tth point of the ith phase, we have that

$$h_{t+1}^{(i)} \leq h_1^{(i)} \exp\big(-2\alpha c t\big) \quad \text{(Phase } i \text{ convergence)}, \tag{4.6}$$

when at least $t+1$ distinct, non-lagged points are sampled in Phase i. Moreover, across non-trivial phases, we get a cumulative contraction:

$$h_1^{(i)} \leq h_1^{(1)} \exp\big(-2\alpha c k_i\big) \quad \text{when } k_i \text{ distinct non-lagged points are sampled up to Phase } i. \tag{4.7}$$

Consequently, since each phase contains at least one non-lagged point,

$$h_1^{(i)} \leq h_1^{(1)} \exp\big(-2\alpha c(i-1)\big) \quad \text{(inter-phase convergence)}. \tag{4.8}$$

PROOF OF CLAIM 4.4. To show the exponential convergence rate in Claim 4.4, we will show that the improvement $h_t - h_{t+1}$ at time t is of order at least (approximately)

$|\nabla f(x_t)|^2$. This will imply, by the strong convexity assumption, a contraction in h_t, which gives us an exponential rate of convergence. To that end, suppose we jump from x_t using a lag size of δ_i; in other words, suppose $x_{t+1} = x_t - \frac{1}{\beta}\tilde{\tilde{\nabla}}_t - \delta_i$. Then the following holds:

$$
\begin{aligned}
h_{t+1} - h_t &= f(x_{t+1}) - f(x_t) && (4.9)\\
&\leq \nabla_t^\top (x_{t+1} - x_t) + \frac{\beta}{2}(x_{t+1} - x_t)^2 \qquad \beta\text{-smooth} && (4.10)\\
&\leq (\tilde{\tilde{\nabla}}_t^\top + \beta\delta_i)(x_{t+1} - x_t) + \frac{\beta}{2}(x_{t+1} - x_t)^2 \qquad \text{by (4.5)} && (4.11)\\
&= (\tilde{\tilde{\nabla}}_t^\top + \beta\delta_i)(-\tilde{\tilde{\nabla}}_t - \delta_i) + \frac{\beta}{2}(-\tilde{\tilde{\nabla}}_t - \delta_i)^2 && (4.12)\\
&= -\frac{1}{2\beta}\|\tilde{\tilde{\nabla}}_t\|^2 - \delta_i\tilde{\tilde{\nabla}}_t - \frac{\beta}{2}\delta_i^2 && (4.13)\\
&\leq -\frac{1}{2\beta}\|\tilde{\tilde{\nabla}}_t\|^2 - \left(-\frac{\tilde{\tilde{\nabla}}_t}{(1+\gamma)\beta}\right)\tilde{\tilde{\nabla}}_t - \frac{\beta}{2}\delta_i^2 && (4.14)\\
&= -\left(\frac{1}{2\beta} - \frac{1}{(1+\gamma)\beta}\right)\|\tilde{\tilde{\nabla}}_t\|^2 - \frac{\beta}{2}\delta_i^2 && (4.15)\\
&\leq -\underbrace{\left(\frac{1}{2\beta} - \frac{1}{(1+\gamma)\beta}\right)}_{=:c}\|\tilde{\tilde{\nabla}}_t\|^2. && (4.16)
\end{aligned}
$$

By (4.5) and the mean value theorem, there is some $\overline{x}_t \in [x_t - \delta_i, x_t] \subset [x_{t-1}, x_t]$ such that $\nabla f(\overline{x}_t) = \tilde{\tilde{\nabla}}_t$, which gives:

$$
\begin{aligned}
h_{t+1} - h_t &\leq -c\|\tilde{\tilde{\nabla}}_t\|^2 = -c\|\nabla f(\overline{x}_t)\|^2 \\
&\leq -2\alpha c[f(\overline{x}_t) - f(x^*)] && \text{by (4.4)}\\
&\leq -2\alpha c h_t && \text{by Claim 4.2.}
\end{aligned}
$$

For this to constitute a contraction, we need that $(1 - 2\alpha c) \in [0, 1)$. To that end, note that $2\alpha c = \left[1 - \frac{2}{1+\gamma}\right]\frac{\alpha}{\beta}$. Since $\gamma > 1$ (as specified in the algorithm), $1 - \frac{2}{1+\gamma} \in (0, 1)$, and since $0 < \alpha \leq \beta$, we have that $\alpha/\beta \in (0, 1]$. It follows that $2\alpha c \in (0, 1) \subset [0, 1)$, as desired. This immediately gives a linear convergence rate:

$$h_{t+1} \le \big(1 - 2\alpha c\big) h_t \le \cdots \le \big(1 - 2\alpha c\big)^t h_1 \le h_1 \exp\big(-2\alpha c t\big)\,.$$

The contractions in (4.6)-(4.7) follow immediately, and (4.8) follows from the fact that if $y_1^{(i)} = x_t$ and $y_1^{(i+1)} = x_s$, then $t < s$. $\square_{\text{Claim 4.4}}$

Claim 4.5 (regret from gradient descent jumps). Assume (4.5) holds for all estimated gradients, and that $x_1 < x^*$. Then, let $\mathrm{regret}_{\mathrm{LGD}}(k)$ be the total regret incurred during the first k phases on the points x_t, as well as the two lagged points just after x_t (see the boxed points in Figure 4.1); i.e., $\mathrm{regret}_{\mathrm{LGD}}(k)$ is the regret incurred during the following samplings:

- the $\frac{64E_{\max}^2 \log \frac{2}{p}}{C\alpha^2\delta_{n_i}^4}$ samples at $x_s = y_t^{(i)}$,
- the $\frac{64E_{\max}^2 \log \frac{2}{p}}{C\alpha^2\xi^4\delta_{n_i}^4}$ samples at $x_{s+1} - \delta_{n_i}$, and
- the first $\frac{64E_{\max}^2 \log \frac{2}{p}}{C\alpha^2\xi^4\delta_{n_i}^4}$ samples at $x_{s+1} - \delta_{n_i+1}$,

for phases $i \in [k]$ (for any k) and their respective t^{th} points. Then $\mathrm{regret}_{\mathrm{LGD}}(k)$ during the first k phases is of order $\mathcal{O}\left(\frac{h \log \frac{1}{p}}{\delta_1^4} + \log \frac{1}{p} \sum_{i=1}^k \frac{1}{|\nabla f(y_1^{(i+1)})|^2}\right)$.

PROOF OF CLAIM 4.5. As we progess toward the optimum, we potentially use smaller and smaller lag sizes. This results in cumbersome sampling when close to the optimum. To prove Claim 4.5, we will use the exponential decay of instantaneous regret to allay some of this regret. Additionally, we will show that the instantaneous regret during Phase i is of order $\delta_{n_i}^2$; since the gradient at $y_1^{(i)}$ is of order δ_{n_i}, this fact will ultimately ultimately allow us to bound $\mathrm{regret}_{\mathrm{LGD}}$ in terms of gradients.

Recall that we are bounding regret incurred during the first k phases only, as stated in the claim. Let q be the contraction in the lag sizes, i.e., $\delta_{i+1} = q\delta_i$. We will bound the regret at $x_t, x_{t+1} - \delta_i, x_{t+1} - \delta_{i+1}$; among these three points, the instantaneous regret is highest at x_t, so we simply bound the instantaneous regret at each of these points by h_t. For any i, let $T^{(i)}$ be the index of the last point in Phase i; i.e., $y_{T^{(i)}}^{(i)}$ is the last point in Phase i. The number of samples at each of these three points is bounded by $\frac{64E_{\max}^2 \log\frac{2}{p}}{C\alpha^2\xi^4\delta_{n_i}^4}$, so this in turn gives us the following upper bound, where $h = f(x_{\min}) - f(x^*)$:

$$\begin{aligned}
\text{regret}_{\text{LGD}} &\le \frac{128E_{\max}^2 \log\frac{2}{p}}{C\alpha^2\xi^4\delta_1^4} h + \sum_{i=1}^{k} \frac{192E_{\max}^2 \log\frac{2}{p}}{C\alpha^2\xi^4\delta_{n_i}^4} \sum_{t=1}^{T^{(i)}} h_t^{(i)} \\
&= \frac{128E_{\max}^2 \log\frac{2}{p}}{C\alpha^2\xi^4\delta_1^4} h + \sum_{i=1}^{k} \frac{192E_{\max}^2 \log\frac{2}{p}}{C\alpha^2\xi^4 q^{4n_i}\delta_1^4} \sum_{t=1}^{T^{(i)}} h_t^{(i)} \\
&= \frac{128E_{\max}^2 \log\frac{2}{p}}{C\alpha^2\xi^4\delta_1^4} h + \frac{192E_{\max}^2 \log\frac{2}{p}}{C\alpha^2\xi^4\delta_1^4} \sum_{i=1}^{k} \frac{1}{q^{4n_i}} \sum_{t=1}^{T^{(i)}} h_t^{(i)} \\
&\le \frac{128E_{\max}^2 \log\frac{2}{p}}{C\alpha^2\xi^4\delta_1^4} h + \frac{192E_{\max}^2 \log\frac{2}{p}}{C\alpha^2\xi^4\delta_1^4} \sum_{i=1}^{k} \frac{1}{q^{4n_i}} \sum_{t=1}^{T^{(i)}} h_1^{(i)} \mathrm{e}^{-2\alpha c(t-1)} \qquad \text{by (4.6)} \\
&= \frac{128E_{\max}^2 \log\frac{2}{p}}{C\alpha^2\xi^4\delta_1^4} h + \frac{192E_{\max}^2 \log\frac{2}{p}}{C\alpha^2\xi^4\delta_1^4} \sum_{i=1}^{k} \frac{1}{q^{4n_i}} h_1^{(i)} \sum_{t=1}^{T^{(i)}} \mathrm{e}^{-2\alpha c(t-1)} \\
&\le \frac{128E_{\max}^2 \log\frac{2}{p}}{C\alpha^2\xi^4\delta_1^4} h + \frac{192E_{\max}^2 \log\frac{2}{p}}{C\alpha^2\xi^4\delta_1^4\left(1 - e^{-2\alpha c}\right)} \sum_{i=1}^{k} \frac{1}{q^{4n_i}} h_1^{(i)}.
\end{aligned}$$

To continue this analysis, we bound $h_1^{(i)}$. To that end note that $|\nabla f(y_1^{(i)})| < \beta(2+\gamma)\delta_{n_i-1}$, since a previous secant g must have satisfied $-\frac{1}{\beta}g < (2+\gamma)\delta_{n_i-1}$. So,

$$\beta(2+\gamma)\delta_{n_i-1} \overset{(a)}{>} |\nabla f(y_1^{(i)})| = |\nabla f(y_1^{(i)}) - \nabla f(x^*)| \overset{(b)}{\ge} \alpha|x^* - y_1^{(i)}|, \tag{4.17}$$

where (a) follows from the fact that $|\nabla f(y_1^{(i)})| < |\nabla f(y_1^{(i)} - \delta_j)| \le |g|$ for some j (by (4.5)), and (b) follows from strong-convexity. Finally, by smoothness, we have

$$h_1^{(i)} \overset{(c)}{\le} \frac{\beta}{2}(x^* - y_1^{(n_i)})^2 \overset{(4.17)}{<} \frac{\beta^3(2+\gamma)^2}{2\alpha^2}\delta_{n_i-1}^2 = \frac{\beta^3(2+\gamma)^2}{2q^2\alpha^2}\delta_{n_i}^2, \tag{4.18}$$

again where (c) follows from smoothness, since $\nabla f(x^*) = 0$.

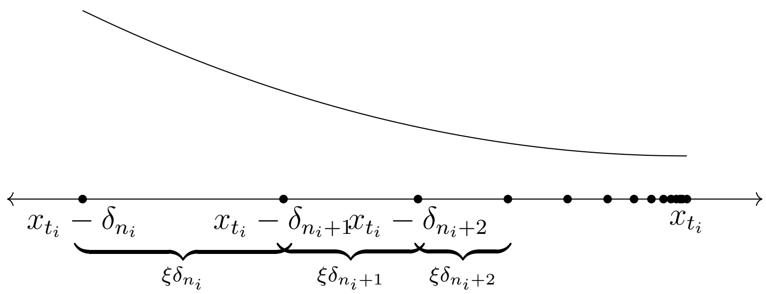


Figure 4.2: Points sampled while transitioning from lag size δ_{n_i}.

Continuing the analysis from above, we have

$$\begin{aligned}
\text{regret}_{\text{LGD}} &\le \frac{128E_{\max}^2 \log\frac{2}{p}}{C\alpha^2\xi^4\delta_1^4}h + \frac{192E_{\max}^2 \log\frac{2}{p}}{C\alpha^2\xi^4\delta_1^4\big(1-e^{-2\alpha c}\big)}\sum_{i=1}^{k}\frac{1}{q^{4n_i}}h_1^{(i)} \\
&\le \frac{128E_{\max}^2 \log\frac{2}{p}}{C\alpha^2\xi^4\delta_1^4}h + \frac{96E_{\max}^2\beta^3(2+\gamma)^2 \log\frac{2}{p}}{C\alpha^4\xi^4\delta_1^4\big(1-e^{-2\alpha c}\big)q^2}\sum_{i=1}^{k}\frac{\delta_{n_i}^2}{q^{4n_i}} \\
&= \frac{128E_{\max}^2 \log\frac{2}{p}}{C\alpha^2\xi^4\delta_1^4}h + \frac{96E_{\max}^2\beta^3(2+\gamma)^2 \log\frac{2}{p}}{C\alpha^4\xi^4\delta_1^2\big(1-e^{-2\alpha c}\big)q^2}\sum_{i=1}^{k}\frac{q^{2n_i}}{q^{4n_i}} \\
&= \frac{128E_{\max}^2 \log\frac{2}{p}}{C\alpha^2\xi^4\delta_1^4}h + \frac{96E_{\max}^2\beta^3(2+\gamma)^2 \log\frac{2}{p}}{C\alpha^4\xi^4\delta_1^2\big(1-e^{-2\alpha c}\big)q^2}\sum_{i=1}^{k}\frac{1}{q^{2n_i}}.
\end{aligned}$$

At this point, for convenience, we will replace $\frac{1}{q^{2n_i}}$ with a gradient estimate. In particular, let $x_{t_i} = y_1^{(i+1)}$ be the iterate at which we jump using $\delta_{n_{i+1}}$ for the first time. Then it must be that $|\nabla f(x_{t_i})| < \beta(2+\gamma)\delta_{n_i}$, or in other words, $q^{n_i} > \frac{|\nabla f(x_{t_i})|}{\beta(2+\gamma)\delta_1}$. So, continuing the inequalities above,

$$\begin{aligned}
\text{regret}_{\text{LGD}} &\le \frac{128E_{\max}^2 \log\frac{2}{p}}{C\alpha^2\xi^4\delta_1^4}h + \frac{96E_{\max}^2\beta^3(2+\gamma)^2 \log\frac{2}{p}}{C\alpha^4\xi^4\delta_1^2\big(1-e^{-2\alpha c}\big)q^2}\sum_{i=1}^{k}\frac{1}{q^{2n_i}} && (4.19) \\
&\le \frac{128E_{\max}^2 \log\frac{2}{p}}{C\alpha^2\xi^4\delta_1^4}h + \frac{96E_{\max}^2\beta^5(2+\gamma)^4 \log\frac{2}{p}}{C\alpha^4\xi^4\big(1-e^{-2\alpha c}\big)q^2}\sum_{i=1}^{k}\frac{1}{|\nabla f(x_{t_i})|^2}. && (4.20)
\end{aligned}$$

$\square_{\text{Claim 4.5}}$

Claim 4.6 (regret from transitioning lag sizes). Assume (4.5) holds for all estimated gradients, and that $x_1 < x^*$. Here we bound the regret which is unaccounted for in Claim 4.5; i.e., we bound the regret incurred at exploratory lagged points. More precisely, let

$\text{regret}_{\delta\text{-transition}}(k)$ be the regret incurred from sampling at $y_1^{(i+1)} - \delta_{n_i+1}, \ldots, y_1^{(i+1)} - \delta_{n_{i+1}+1}$ for $1 \le i \le k$. Then, $\text{regret}_{\delta\text{-transition}}(k) \in \mathcal{O}\left(\log \frac{1}{p} \sum_{i=1}^{k} \frac{1}{|\nabla f(y_1^{(i+1)})|^2}\right)$, for any $k \le T$.

PROOF OF CLAIM 4.6. Here we bound the regret incurred while transitioning lag sizes (e.g., the regret from the samples taken at points which do *not* appear in boxes in Figure 4.1). In order to bound the regret resulting from δ-transitions, we argue that whenever we transition from phase n_i, the current gradient is small in magnitude compared to δ_{n_i}. This allows us to bound the number of lagged points sampled during any transitions from δ_{n_i} to $\delta_{n_{i+1}}$, which in turn gives us a bound on the total regret resulting from these transitions.

Let $t_1, \ldots, t_k$ be such that $x_{t_i} = y_1^{(i+1)}$. To bound the regret from transitioning lag sizes, let us consider the regret incurred transitioning from δ_{n_i} to $\delta_{n_{i+1}}$, which happens at time t_i (note that n_i and n_{i+1} are not consecutive if the algorithm goes down multiple lag sizes during the same round). The regret incurred from transitioning to $\delta_{n_{i+1}}$ is the regret incurred at $x_{t_i} - \delta_{n_i+1}, x_{t_i} - \delta_{n_i+2}, \ldots x_{t_i} - \delta_{n_{i+1}+1}$ (see Figure 4.2). Let us begin by bounding the number of lag sizes that can be passed in one round. Suppose we sample at $x_{t_i} - \delta_{n_i+j}$ and decide to decrease the lag size to δ_{n_i+j}. In other words, we observe that

$$-\frac{1}{\beta} \cdot \underbrace{\frac{\overline{f}(x_{t_i} - \delta_{n_i+j}) - \overline{f}(x_{t_i} - \delta_{n_i+j-1}) + \varepsilon(\xi\delta_{n_i+j-1})}{\xi\delta_{n_i+j-1}}}_{=:g} < (2+\gamma)\delta_{n_i+j-1} \tag{4.21}$$

and therefore need to begin sampling at $x_{t_i} - \delta_{n_i+j+1}$. In this case, observe that

$$|\nabla f(x_{t_i})| \overset{(a)}{\le} |g| \overset{(b)}{\le} \beta(2+\gamma)\delta_{n_i+j-1} = \beta(2+\gamma)q^{j-1}\delta_{n_i}.$$

where (a) follows from (4.5) and (b) from (4.21). Thus for

$$J_i = \max\left\{2,\ \left\lceil \frac{1}{\log q} \log\left(\frac{|\nabla f(x_{t_i})|}{\beta(2+\gamma)\delta_{n_i}}\right)\right\rceil + 1\right\},$$

we will not transition; i.e., $n_{i+1} < n_i + J_i$ and we will not sample at $x_{t_i} - \delta_{n_i+J_i+1}$.

Next, we will bound the instantaneous regret incurred by sampling at $x_{t_i} - \delta_{n_i+j}$. Suppose we have just transitioned to δ_{n_i+j}, and must now sample at $x_{t_i} - \delta_{n_i+j}$ and $x_{t_i} - \delta_{n_i+j+1}$ in order to determine whether or not to reduce the lag size further. Since we have just transitioned to δ_{n_i+j}, we know that (4.21) holds. This in turn implies that

$$|\nabla f(x_{t_i} - \delta_{n_i+j})| \leq |g| < \beta(2+\gamma)q^{j-1}\delta_{n_i}.$$

So, the regret we get from each sample (whether at $x_{t_i} - \delta_{n_i+j}$ or $x_{t_i} - \delta_{n_i+j+1}$), is bounded by

$$f(x_{t_i} - \delta_{n_i+j}) - f(x^*) \leq \frac{\beta}{2\alpha}\left[\beta(2+\gamma)q^{j-1}\delta_{n_i}\right]^2 = \frac{\beta^3(2+\gamma)^2 q^{2(j-1)}\delta_{n_i}^2}{2\alpha}. \tag{4.22}$$

We can now calculate our regret from lag transitions as follows:

$$
\begin{aligned}
\text{regret}_{\delta\text{-transition}} &\le \sum_{i=1}^{k}\sum_{j=1}^{J_i} \frac{128E_{\max}^2 \log\frac{2}{p}}{C\alpha^2\xi^4\delta_{n_i+j}^4} \cdot \left[f(x_{t_i}-\delta_{n_i+j}) - f(x^*)\right] \\
&\le \sum_{i=1}^{k}\sum_{j=1}^{J_i} \frac{128E_{\max}^2 \log\frac{2}{p}}{C\alpha^2\xi^4\delta_{n_i+j}^4} \cdot \frac{\beta^3(2+\gamma)^2 q^{2(j-1)}\delta_{n_i}^2}{2\alpha} \qquad \text{by (4.22)} \\
&= \frac{64E_{\max}^2\beta^3(2+\gamma)^2\log\frac{2}{p}}{C\alpha^3\xi^4 q^2} \sum_{i=1}^{k} \frac{1}{\delta_{n_i}^2} \sum_{j=1}^{J_i} \frac{1}{q^{2j}} \\
&\le \frac{64E_{\max}^2\beta^3(2+\gamma)^2\log\frac{2}{p}}{C\alpha^3\xi^4 q^2\left(\frac{1}{q^2}-1\right)} \sum_{i=1}^{k} \frac{1}{\delta_{n_i}^2} \frac{1}{q^{2(J_i+1)}} \\
&= \frac{64E_{\max}^2\beta^3(2+\gamma)^2\log\frac{2}{p}}{C\alpha^3\xi^4 q^8\left(\frac{1}{q^2}-1\right)} \sum_{i=1}^{k} \frac{1}{\delta_{n_i}^2} \frac{1}{q^{2(J_i-2)}} \\
&= \frac{64E_{\max}^2\beta^3(2+\gamma)^2\log\frac{2}{p}}{C\alpha^3\xi^4 q^8\left(\frac{1}{q^2}-1\right)} \sum_{i=1}^{k} \frac{1}{\delta_{n_i}^2} e^{(\log q^{-2})(J_i-2)} \\
&= \frac{64E_{\max}^2\beta^3(2+\gamma)^2\log\frac{2}{p}}{C\alpha^3\xi^4 q^8\left(\frac{1}{q^2}-1\right)} \sum_{i=1}^{k} \frac{1}{\delta_{n_i}^2} e^{(\log q^{-2})\left(\left\lceil \frac{1}{\log q}\log\left(\frac{|\nabla f(x_{t_i})|}{\beta(2+\gamma)\delta_{n_i}}\right)\right\rceil - 1\right)} \\
&\le \frac{64E_{\max}^2\beta^3(2+\gamma)^2\log\frac{2}{p}}{C\alpha^3\xi^4 q^8\left(\frac{1}{q^2}-1\right)} \sum_{i=1}^{k} \frac{1}{\delta_{n_i}^2} e^{(\log q^{-2})\left(\frac{1}{\log q}\log\left(\frac{|\nabla f(x_{t_i})|}{\beta(2+\gamma)\delta_{n_i}}\right)\right)} \\
&\le \frac{64E_{\max}^2\beta^3(2+\gamma)^2\log\frac{2}{p}}{C\alpha^3\xi^4 q^8\left(\frac{1}{q^2}-1\right)} \sum_{i=1}^{k} \frac{1}{\delta_{n_i}^2} e^{2\log\left(\frac{\beta(2+\gamma)\delta_{n_i}}{|\nabla f(x_{t_i})|}\right)} \\
&= \frac{64E_{\max}^2\beta^3(2+\gamma)^2\log\frac{2}{p}}{C\alpha^3\xi^4 q^8\left(\frac{1}{q^2}-1\right)} \sum_{i=1}^{k} \frac{1}{\delta_{n_i}^2} \left(\frac{\beta(2+\gamma)\delta_{n_i}}{|\nabla f(x_{t_i})|}\right)^2 \\
&= \frac{64E_{\max}^2\beta^5(2+\gamma)^4\log\frac{2}{p}}{C\alpha^3\xi^4 q^8\left(\frac{1}{q^2}-1\right)} \sum_{i=1}^{k} \frac{1}{|\nabla f(x_{t_i})|^2}.
\end{aligned}
$$

$\square_{\text{Claim 4.6}}$

Claim 4.7 (total regret). Assume (4.5) holds for all estimated gradients, and that $x_1 < x^*$. Then the regret is of order $(\log T)^2 T^{1/2}$, for $\delta_1 = 1/\log T$, $p = T^{-\lambda}$ for $\lambda > 3/2$, and $\gamma = 1 + \frac{1}{\log T}$.

PROOF OF CLAIM 4.7. The expressions of regret in Claims 4.5 and 4.6 grow rapidly as k increases, so we only use these expressions to bound regret in early phases. For later phases, we argue that the gradient is small, which gives us a bound on the late-phase regret. Balancing the regret between early and late stages ultimately achieves the claimed $\widetilde{\mathcal{O}}(T^{1/2})$ regret bound.

In particular, we will break the regret from the δ-transition into two categories: the regret when the gradient is small ($|\nabla f(x_{t_i})| \leq T^{-d}$) and the regret when the gradient is large ($|\nabla f(x_{t_i})| > T^{-d}$). If k_d is the number of δ-transitions until $|\nabla f(x_{t_i})| \leq T^{-d}$, then k_d is of order $\log T$. To show this precisely, observe that

$$\begin{aligned} |\nabla f(x_{t_i})|^2 &\leq \frac{2\beta^2}{\alpha}\big(f(x_{t_i}) - f(x^*)\big) && \text{smoothnes and str. convexity} \\ &\leq \frac{2\beta^2 h_1^{(1)}}{\alpha} \mathrm{e}^{-2\alpha c(i-1)}. \end{aligned}$$

So, if $i > \frac{1}{\alpha c}\left(\frac{1}{2}\log\frac{2\beta^2 h_1^{(1)}}{\alpha} + d\log T\right) + 1$, then $|\nabla f(x_{t_i})| \leq T^{-d}$. In turn, if $|\nabla f(x_{t_i})| \leq T^{-d}$, then

$$f(x_{t_i}) - f(x^*) \leq \frac{\beta}{2\alpha^2}T^{-2d},$$

by smoothness and strong convexity. Now, define $\mathrm{regret}_{\delta\text{-transition}}$ to be the total regret from transitioning lag sizes over all iterations; that is, if m is the last phase of the algorithm (which is well-defined since number of iterations is bounded by T), then $\mathrm{regret}_{\delta\text{-transition}} = \mathrm{regret}_{\delta\text{-transition}}(m)$. Then we can bound $\mathrm{regret}_{\delta\text{-transition}}$ as

$$\begin{aligned} \mathrm{regret}_{\delta\text{-transition}} &= \mathrm{regret}_{\delta\text{-transition}}(k_d) + \left[\mathrm{regret}_{\delta\text{-transition}} - \mathrm{regret}_{\delta\text{-transition}}(k_d)\right] \\ &\leq \frac{64E_{\max}^2\beta^5(2+\gamma)^4\log\frac{2}{p}}{C\alpha^3\xi^4 q^8\left(\frac{1}{q^2}-1\right)}k_d T^{2d} + \frac{\beta}{2\alpha^2}T^{1-2d}. \end{aligned}$$

Setting $d = 1/4$ and bounding k_d by $\left\lceil \frac{1}{\alpha c}\left(\frac{1}{2}\log\frac{2\beta^2 h_1^{(1)}}{\alpha} + d\log T\right)\right\rceil + 2$, we get that

$$\text{regret}_{\delta\text{-transition}} \in \mathcal{O}\Big((\log T)^2 T^{1/2}\Big).$$

Since $\text{regret}_{\text{LGD}}$ is of the same order as $\text{regret}_{\delta\text{-transition}}$ (ignoring the lower-order term $\frac{h\log\frac{1}{p}}{\delta_1^4}$),

$$\begin{aligned}
\text{regret} &\le (1-p)^T\Big[\text{regret}_{\text{LGD}} + \text{regret}_{\delta\text{-transition}}\Big] + \Big(1-(1-p)^T\Big)T \\
&\in \mathcal{O}\left((1-p)^T(\log T)^2 T^{1/2} + \Big(1-(1-p)^T\Big)T\right) \\
&= \mathcal{O}\left(\big(1-o(1)\big)(\log T)^2 T^{1/2} + o(T^{1/2})\right),
\end{aligned}$$

for $p = T^{-\lambda}$, $\lambda > 3/2$. This gives us a regret bound of $\mathcal{O}\left((\log T)^2 T^{1/2}\right)$. □

The case of two groups.

Having developed a procedure that achieves near-optimal-regret for single-dimensional monotonic stochastic convex optimization with bandit feedback, we now return to the problem of two groups.

Algorithmic Approach. Assume without loss of generality that $\max\{s(1,2), s(2,1)\} > 0$ (otherwise if both are 0, then we get the single group case).[7] Recall that we have to ensure the constraint $x_{1,t} \ge x_{2,t'} - s(1,2)$ and $x_{1,t} \ge x_{2,t'} - s(2,1)$ for all $1 \le t' \le t \le T$. Our overall approach is simply described as a continuous-time procedure in the case where we have access to perfect gradient feedback, i.e., $\nabla f_i(x_{i,t})$:

1. **Coordinate descent phase (continuous-time).** Starting with $x_{1,0} = x_{2,0} = x_{\min}$, pick an arbitrary coordinate, say i, and increase it while keeping the other coordinate $-i$ fixed until either (a) $\nabla f_i(x_{i,t}) = 0$, or (b) $x_{i,t} = x_{-i,t} + s(-i,i)$. Switch the

[7] While we require slacks to be nonnegative, the algorithm allows for negative slacks as well with a minor adjustment. In particular, assuming that the feasible set of decisions is non-empty, we compute the minimum decision on each dimension that is feasible. The algorithm then proceeds from this point.

coordinate $i \leftarrow -i$ and repeat until, for both $i = 1, 2$, either $x_{i,t} = x_{-i,t} + s(-i, i)$ or $\nabla f_i(x_{i,t}) = 0$. Once that happens, go to step 2.

2. **Combined descent phase (continuous-time).** If $\nabla f_i(x_{i,t}) = 0$ for both $i = 1, 2$, then we are done – the CFTD constraint doesn't bind and the unconstrained optimum is the same as the constrained optimum. Else, there is some $i^* \in \{1, 2\}$ such that $\nabla f_i(x_{i^*,t}) = 0$ and $x_{-i^*,t} = x_{i^*,t} + s(i^*, -i^*)$ (since we assumed at least one slack is non-zero). At that point, we can deduce that the corresponding CFTD constraint will be tight at the optimum (which means we can again reduce to the single group case). Then define $g(x) = f_{i^*}(x) + f_{-i^*}(x + s(i^*, -i^*))$, and continue reducing $x_{i^*,t}$ and $x_{-i^*,t} = x_{i^*,t} + s(i^*, -i^*)$ jointly until $\nabla g(x_{i^*,t}) = 0$.

It is clear that this continuous-time dynamic finds the constrained optimum while satisfying CFTD. The challenge then is to convert this process into a practical discrete-time procedure with only noisy bandit feedback on each dimension, while ensuring optimal overall regret. To adapt the above continuous-time dynamic to a discrete-time setting with noisy bandit feedback, we use a similar discretization as ADA-LGD, where for each coordinate, we compute *non-lagged iterates* and slowly move towards the non-lagged iterates using *lagged iterates* while searching for the right lag size to estimate the gradient. However, due to the two-group setting, a number of methodological adaptations must be made to overcome the following challenges:

(1) the unconstrained optimality for a particular group, i.e., $\nabla f_i(x_{i,t}) = 0$, can only be approximately detected, which means that more care must be taken before entering the combined phase;

(2) similarly, since the unconstrained optimality for a particular group can only be approximately detected, the trigger for switching between groups in the coordinate descent phase must be adapted as well (recall that reaching optimality for a group triggered the switch to the other group in the continuous-time method);

Algorithm 9: Switch-then-Combine Adaptive Lagged Gradient Descent (SCADA-LGD)

input: α strongly convex and β-smooth $f(x_1, x_2) = f_1(x_1) + f_2(x_2)$, horizon T, slacks $s(1,2), s(2,1), x_{\min}, \gamma,$ $q \in (0,1), \xi = 1-q$, initial lag $\delta = \delta^{(1)} = \delta^{(2)}$, $\text{GRAD}(x,y)$ = gradient computed using average of samples at x and y (Lem. 2.3), $n(d) = 64E_{\max}^2 \log\frac{2}{p}/(C\alpha^2 d^4)$, where $E_{\max}$ is a bound on the noise, and C is the Hoeffding constant.

1 Initialize queues to maintain tuples of (sampling point, number of samples, and the type of sampling point (S)):
$Q_1 = Q_2 = \{(x_{\min}, n(\xi\delta), S=1), (x_{\min} + \xi\delta, n(\xi q\delta), S=2), (x_{\min} + \delta, n(\delta), S=0)\}$
`// ` $S = k > 0$ `represents the` k`th lagged point,` $S = 0$ `represents a non-lagged point, and` $S \in \{-1,-2\}$ `represents the two feasibility iterates.`

2 Initialize $x_1 = x_2 = x_{\min}$

3 **Coordinate Descent Phase:**

4 **while** *fewer than T samples have been taken* **do**

5 $\quad \mathcal{X}_i = [x_i, x_{-i} + s(i,-i)]$ for $i = 1,2$ `// current feasible points for Group` $i = 1,2$

6 $\quad$ **if** $\exists j : Q_j \cap \mathcal{X}_j \neq \emptyset$ **then** Set $i \in \{j \mid Q_j \cap \mathcal{X}_j \neq \emptyset\}$; **else** go to **Combined Phase** (line 10)

7 $\quad$ set `switch` = False

8 $\quad$ **while** $Q_i \cap \mathcal{X}_i \neq \emptyset$ **do**

9 $\quad\quad$ **Sample:** Let (x_i, n_x, S_i) be the lowest point to sample in $Q_i \cap \mathcal{X}_i$ by first coordinate (with ties broken arbitrarily) and obtain n_x samples of $f_i(x_i)$; Update $Q_i \leftarrow Q_i \setminus (x_i, n_x, S_i)$

10 $\quad\quad$ **Gradient checks based on type (S_i) of x_i:**

(a) **if** x_i is a jth lagged point (i.e., $S_i = j$) and $j > 1$ **then**

if $-\frac{1}{\beta}\text{GRAD}((j-1)\text{st lagged point}, x_i) < (1+\gamma)\delta^{(i)}$:

Plan to sample next lagged point: $Q_i \leftarrow Q_i \cup \{(x_i + \xi q\delta^{(i)}, n(\xi q\delta^{(i)}), S = j+1)\}$

$\delta^{(i)} \leftarrow q\delta^{(i)}$ and adjust the sample size for the next non-lagged iterate in Q_i to $n(\delta^{(i)})$

(b) **if** x_i is a non-lagged point (i.e., $S_i = 0$) **then**

Let $g = \text{GRAD}(x_i - \delta^{(i)}, x_i)$

if $-\frac{1}{\beta}g < T^{-1/4}$ **then** set $Q_i \leftarrow \emptyset$ (never move group i in coordinate descent again)

else populate queue with the next non-lagged and lagged points:

Let $y = x_i - \delta^{(i)} - \frac{1}{\beta}g$, and

$Q_i \leftarrow Q_i \cup \{(y - \delta^{(i)}, n(\xi\delta^{(i)}), S=1),\ (y - q\delta^{(i)}, n(\xi q\delta^{(i)}), S=2), (y, n(\delta^{(i)}), S=0)\}$

if lagged size has dropped enough so that $\delta^{(i)} < \delta^{(-i)}$, then switch group: `switch` = True

(c) **if** x_i is the second feasibility iterate (i.e., $S_i = -2$) **then**

Let $g = \text{GRAD}(\text{previous feasibility iterate}, x_i)$

if $-\frac{1}{\beta}g \geq \left(\frac{(2+\gamma)\beta}{q\alpha} + 1\right)\delta^{(-i)}$ **then**

set $Q_1, Q_2 \leftarrow \emptyset$ (i.e., enter the combined phase)

else if $Q_{-i} = \emptyset$ **then** sample at (x_1, x_2) for the remaining time until T

Add feasibility iterates: if (`switch` = True or $Q_i \cap \mathcal{X}_i = \emptyset$) AND $\delta^{(i)} < \delta^{(-i)}$ **then**
set $\mathcal{X}_{-i} \leftarrow [x_{-i}, x_i + s(-i,i)]$

$Q_{-i} \leftarrow Q_{-i} \cup \{(\max\mathcal{X}_{-i} - \delta^{(i)}, n(\xi\delta^{(i)}), S=-1), (\max\mathcal{X}_{-i} - q\delta^{(i)}, n(\xi q\delta^{(i)}), S=-2)\}$
switch groups: $i \leftarrow -i$

11 **Combined phase:**

12 **while** *fewer than T samples have been taken* **do**

13 $\quad$ Run ADA-LGD with previously sampled lagged points (with lags $\delta^{(-i)}$ and $q\delta^{(-i)}$) on function h:

14 $\quad\quad$ **if** Group 2 is tight, set $h(x) = f_1(x) + f_2(x + s(2,1))$; **else** set $h(x) = f_1(x + s(1,2)) + f_2(x)$

(3) recall that the analysis of ADA-LGD depended on controlling the lag sizes and resulting gradient accuracy as the sampling process approached the optimum. Since

the two groups may have different derivatives at the current point, their lag sizes may differ significantly as well. This raises an issue as the group with the smaller derivative will spend more time sampling, thus causing the other group to incur regret in the meantime;

(4) moreover, if the two groups have different lag sizes, this disparity must be reconciled when entering the combined phase, where the groups must proceed jointly with a single lag size.

We explain our novel discrete-time bandit algorithm SCADA-LGD below, which addresses these concerns.

Coordinate descent phase (SCADA-LGD)**.** In the coordinate descent phase of SCADA-LGD, each group maintains a queue of points to sample denoted as Q_i for group i, and the iterate with the minimum value for coordinate i is sampled first. As in ADA-LGD, we call iterates computed using gradient jumps "non-lagged iterates" (i.e, $x_i \leftarrow x_i - \delta^{(i)} - g_i(x_i)/\beta$ where $g_i(x_i)$ denotes the estimate of the gradient at x_i), and we move towards non-lagged iterates using "lagged iterates" (defined by $x_i - \delta^{(i)}$) to control the accuracy of the gradients. We additionally introduce a new set of iterates, which we refer to as "feasibility iterates," which are used to detect whether or not the combined phase should be initiated. These iterates are used to provide the group with larger lag size (and thus a lower-accuracy gradient estimate) with a higher-accuracy gradient estimate, thus alleviating concern (4) if the combined phase is initiated.

The two groups are optimized in turn, one at a time, and the following values of gradients are used to detect various conditions:

(a) **Low gradient accuracy:** If the point sampled is the kth lagged point[8] ($k > 1$), and the gradient estimate is less than $\mathcal{O}(\delta^{(i)})$, then we know that we need to sample the

[8] We do not compute the gradient for the first lagged point, since there is no "previous" lagged point with which a difference quotient can be calculated. The samples taken at the first lagged point are only used to estimate the gradients at the next lagged point.

next lagged point (with a lower lag size) as well. As was the case for ADA-LGD, this check allows us to continue converging to the optimum despite the low-gradient condition being met.

(b) **Low gradient or Switch to other group**: If the point sampled is a non-lagged iterate in group i, and the gradient estimated is small enough (i.e., less than $\mathcal{O}(T^{-1/4})$), then we permanently switch to sampling group $-i$ and never optimize group i in the coordinate descent phase again. This is to ensure that group $-i$ does not incur excessive regret while group i is already close to its optimum and thus further improvements to group i would take a large number of steps. Otherwise, the next non-lagged and lagged iterates are computed (as in ADA-LGD) and added to the queue to sample. If the lag size of group i is smaller than the lag size of group $-i$, then a switch to optimizing group $-i$ is triggered. This is to prevent the algorithm from devoting too much time to one group without optimizing the other. Intuitively, if the slacks were high enough that the constraints are never tight, then this would ensure that the lag sizes of the two groups differ by at most a factor of q, in the coordinate descent phase.

(c) **Combined Phase Trigger:** When the next points to sample for group i are the two feasibility iterates, then a gradient estimate is computed using the function values at these two points. At this stage, we know that group $-i$ is close to its optimum, since feasibility iterates are only added in such a scenario. So, if the gradient is large enough (i.e., greater than $\Omega(\delta^{(-i)})$), then we can infer that group i is far from its optimum, and this indicates that the CFTD constraint is tight at the joint optimum; hence, in this case, the combined descent phase is triggered.

After these checks, it may be the case that (i) a switch to the other group was triggered, or that (ii) there are no more feasible points for group i to sample. If either of these situations occurs, then the algorithm will switch groups, adding feasibility iterates to Q_{-i} if applicable (that is, if the lag size of group i is smaller than the lag size of group $-i$). If

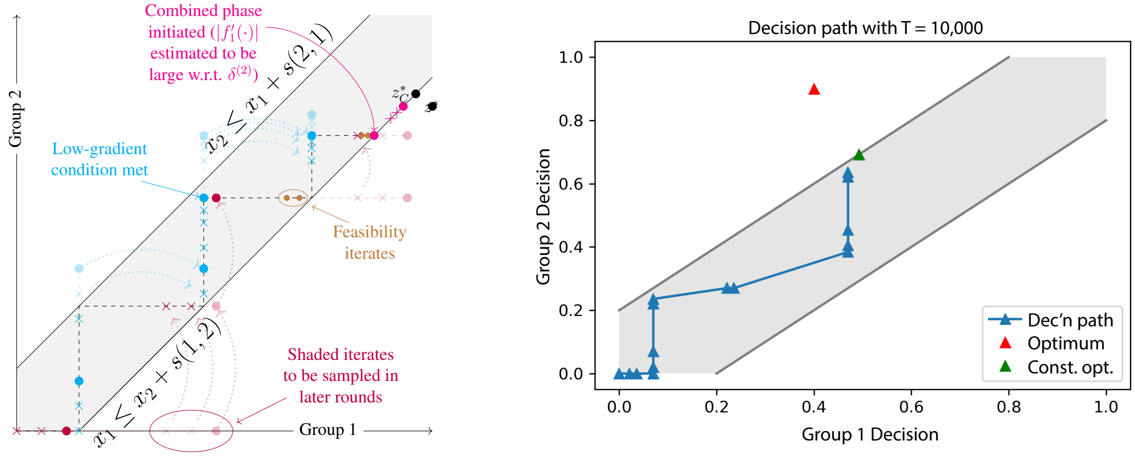


Figure 4.3: (**left**) An illustration of a potential decision trajectory of Algorithm 9, drawn over a shaded decision space $\mathcal{X}_F^2$. The red iterates are sampled to estimate f_1', the blue iterates are sampled to estimate f_2', points marked with an "x" are lagged iterates, brown vertices are the "feasibility check iterates" described in the algorithm, and pink iterates in the combined phase. Points which are infeasible when calculated are shaded and will be sampled if and when they become feasible. (**right**) An actual decision path generated by Alg. 9, as discussed in Section 14.

neither (i) nor (ii) occur, then we remain on group i and repeat this process.

Combined descent phase (SCADA-LGD)**.** The decisions for the two groups are locked together once the combined phase is initiated. This means that the function $f(x_1, x_2)$ can be expressed as a single-variable function $h(x) = f_1(x) + f_2(x + r)$ for some $r \in \mathbb{R}$. Since feasibility iterates were sampled in the group with larger lag size, both groups enter the combined phase with similar-accuracy gradients, which provide the first gradient estimate of h. At this point, ADA-LGD is run on h for the remainder of the time horizon.

The following result shows that SCADA-LGD achieves an order-optimal (up to poly-logarithmic factors) regret guarantee.

Theorem 4.3. *Assume that $f(x_1, x_2) = f_1(x_1) + f_2(x_2)$ is α-strongly convex and β-smooth function, $(x_1^*, x_2^*) = \arg\min_{(x_1,x_2)\in\mathbf{R}^2} f(x_1, x_2) \in (x_{\min}, \infty)^2$, and that the noise is mean zero, independent, and sub-Gaussian of bounded sub-Gaussian norm (Assumption 4.2). Then* SCADA-LGD *(Alg. 9) satisfies CFTD and, on input of $\delta = 1/\log T$, $\gamma = 1 + \frac{1}{\log T}$, any $q \in (0, 1)$, and $p = T^{-2}$, incurs $\widetilde{\mathcal{O}}\left(T^{1/2}\right)$ regret.*

Proof. Algorithm 9 is composed of two phases: (1) a coordinate descent phase, in which

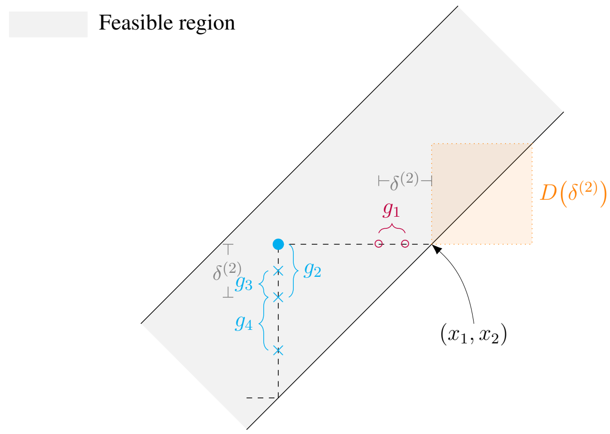


Figure 4.4: Decision space $\mathcal{X}_F^N$ with the following scenario in Alg. 9: Group 2 has undergone at least one lag size transition and Group 1 is at a tight constraint. X's represent lagged points, filled-in circles the non-lagged points, and empty circles the feasibility iterates; gradients of f_1 are estimated at red nodes, and gradients of f_2 are estimated at blue nodes. $g_1, \ldots, g_4$ denote gradient estimates obtained from the two nodes indicated by the brackets. The orange square of side length $D\big(\delta^{(2)}\big)$ is relevant to Claims 4.8-4.9.

ADA-LGD (Alg. 8) is run separately on the two groups, and the group being optimized changes each time a lag size is contracted; and (2) a combined phase, where the decisions for one group are locked with respect to the decisions for the other, and the two are optimized simultaneously. Much of the analysis is identical to that of Algorithm 8; as such, we focus on the differences, and break up the argument into claims.

Claim 4.8. Suppose $\delta^{(2)} < \delta$ is the lag size for Group 2, and Group 1 has hit the boundary of the feasible region (see Fig. 4.4) at $(x^{(1)}, x^{(2)})$. Then the following hold with probability at least $(1-p)^4$:

- if $x_1^* - x_1 \leq D(\delta^{(2)})$, then the algorithm does not enter the combined phase; and
- if $x_1^* - x_1 \geq D(\delta^{(2)}) + H(\delta^{(2)})$, then the algorithm enters the combined phase.

Proof of Claim 4.8. First suppose that $x_1^* - x_1 \leq D(\delta^{(2)})$. Then by smoothness, we have that $|f_1'(x_1)| \leq \beta D(\delta^{(2)}) = (2+\gamma)\frac{\beta^2}{\alpha} \cdot \frac{\delta^{(2)}}{q}$. Now let g_1 be the gradient estimate obtained from points $x_1 - \delta^{(2)}$ and $x_1 - q\delta^{(2)}$, as shown in Figure 4.4. Then

$$\begin{aligned}
-\frac{1}{\beta} g_1 &= \frac{1}{\beta}|g_1| \le \frac{1}{\beta}|f_1'(x_1 - \delta^{(2)})| \\
&\le \frac{1}{\beta}|f_1'(x_1)| + \delta \le D(\delta^{(2)}) + \delta^{(2)} \\
&= (2+\gamma)\frac{\beta}{\alpha} \cdot \frac{\delta^{(2)}}{q} + \delta^{(2)}
\end{aligned}$$

with probability at least $(1-p)^2$. In this case, the algorithm does not enter the combined phase.

Now suppose that $x_1^* - x_1 \ge D(\delta^{(2)}) + H(\delta^{(2)})$. Expanding this expression, we get that $x_1^* - x_1 \ge (2+\gamma)\frac{\beta^2}{\alpha^2} \cdot \frac{\delta^{(2)}}{q} + \frac{\beta\delta^{(2)}}{\alpha}$. This implies by strong convexity that

$$|f_1'(x)| \ge (2+\gamma)\frac{\beta^2}{\alpha} \cdot \frac{\delta^{(2)}}{q} + \beta\delta^{(2)}.$$

Since $|g_1| > |f_1'(x)|$ with probability at least $(1-p)^2$, it follows that $-\frac{1}{\beta}g_1 \ge (2+\gamma)\frac{\beta}{\alpha} \cdot \frac{\delta^{(2)}}{q} + \delta^{(2)}$ with the same probability. Hence, in this case, the algorithm will enter the combined phase. $\square_{\text{Claim 4.8}}$

Claim 4.9. Let g_1 be the gradient estimated at the feasibility iterates (see Figure 4.4), and let $z^* = (x_1^*, x_2^*)$ be the unconstrained optimum.

- If $z^* \in \mathcal{X}_F^2$, then the algorithm never enters the combined phase.
- If the algorithm enters the combined phase based on the gradient estimate g_1 (as in Fig. 4.4), then $-\frac{1}{\beta}g_1 \ge (2+\gamma)\delta^{(2)}$.

Proof of Claim 4.9. We begin with the first claim. It is enough to show that Group 1 sampling will not induce a transition to the combined phase, given that Group 2 has already undergone at least one lag size transition. As in Claim 4.9, let $\delta^{(2)} < \delta$ be the current lag size of Group 2. We will argue that the Group 2 gradient must be small, since this group has transitioned from the previous lag size $\delta^{(2)}/q$; thus, the optimum x_2^* of f_2 must be close to the current point x_2.

To that end, let g_4 be the gradient which induced the transition to $\delta^{(2)}$ (see, e.g., Fig. 4.4). It follows that $-\frac{1}{\beta}g_4 < (2+\gamma)\frac{\delta^{(2)}}{q}$. Let y be the non-lagged iterate corresponding to g_4 (so, in Fig. 4.4, $y = x_2$). Then $\alpha|x_2^* - y| \leq |f_2'(y)| < |g_4| < (2+\gamma)\beta\delta^{(2)}/q$. It follows that

$$|x_2^* - x_2| \leq |x_2^* - y| < (2+\gamma)\frac{\beta}{\alpha} \cdot \frac{\delta^{(2)}}{q} = D\big(\delta^{(2)}\big). \tag{4.23}$$

Since we are assuming that $z^* \in \mathcal{X}_F^2$, it must be that $|x_1^* - x_1| \leq D\big(\delta^{(2)}\big)$ as well. Thus, by Claim 4.8, the algorithm will not enter the combined phase with high probability.

To prove the second statement, let g_1 be the gradient estimate (in the Group 1 direction) which induced the transition to the combined phase (see, e.g., Fig. 4.4). Then, since $\alpha \leq \beta$ and $0 < q < 1$,

$$-\frac{1}{\beta}g_1 \geq \left(\frac{(2+\gamma)\beta}{q\alpha} + 1\right)\delta^{(2)} > (2+\gamma)\delta.$$

In other terms, since the algorithm transitions to the combined phase *only when* the gradient in the Group 1 direction is large enough, a lag size transition will not be triggered. $\square_{\text{Claim 4.9}}$

Claim 4.10. The feasibility check iterates (e.g., the brown nodes in Figure 4.3) increase the overall regret during the coordinate phase by at most a factor of two.

Proof of Claim 4.10. Feasibility check iterates are sampled $\widetilde{\mathcal{O}}(1/\delta_i^4)$ times, where δ_i is the current lag size of the other group. Thus the regret incurred at the two feasibility check iterates is strictly less than the regret incurred at the previous two lagged iterates in the other group. $\square_{\text{Claim 4.10}}$

Claim 4.11. Let $\delta_{n_1} > \cdots > \delta_{n_m}$ be the distinct non-trivial lag sizes of Group j (excluding δ if δ is a non-trivial lag size), where $\delta_i = q^i\delta$. Defining $n_0 = 0$, we have that $n_{i+1} - n_i \in \mathcal{O}(1)$ for $0 \leq i < m$. Moreover, the number of gradient-scaled jumps taken with lag size δ_{n_i} is constant as well, for $1 \leq i \leq m$.

Proof of Claim 4.11. Let us begin by bounding $n_{i+1} - n_i$, for $i \geq 1$. Let $x_{t_{i+1}}$ denote the first non-lagged iterate which admits a lag size of $\delta_{n_{i+1}}$. In this case, the previous non-lagged iterate x admitted a lag size of δ_{n_i}. We can show that the gradient estimate g at x is an underestimate (in magnitude) by $\mathcal{O}(\delta_{n_i})$:

$$g \geq \frac{f_j(x) - f_j(x - \delta_{n_i})}{\delta_{n_i}} \geq \nabla f_j(x - \delta_{n_i}) + \frac{\alpha}{2}\delta_{n_i}$$

by the Sandwich Lemma (Lem. 2.3) and strong convexity. Thus $x_{t_{i+1}}$ must be shy of the optimum by at least $\Omega(\delta_{n_i})$, and thus has gradient of order $\Omega(\delta_{n_i})$. Since the gradient changes by at most a constant factor from x to $x_{t_{i+1}}$, the number $n_{i+1} - n_i$ of transient lag sizes must also be constant.

The above argument can be extended to bound n_1 as well. If the initial lag size of δ was non-trivial (i.e., some gradient-scaled jump was made using a lag size of δ), then the above argument hold verbatim. Otherwise, the initial non-lagged iterate $(x_{\min} + \delta)$ has a gradient of magnitude $\mathcal{O}(\delta)$. However, since $\delta = 1/\log T$ and the optimum is assumed to be strictly greater than zero, we may assume that T is large enough so that the gradient at $x_{\min} + \delta$ is large with respect to δ.

Finally, we argue the "moreover" claim. While δ_{n_i} is the lag size, the gradient has magnitude $\Theta(\delta_{n_i})$. Once the gradient drops below $\mathcal{O}(q\delta_{n_i})$, a lag transition is initiated. Since the gradient-scaled jumps yield linear convergence, t jumps with lag size δ_{n_i} would yield instantaneous regret of order $e^{-t}\delta_{n_i}^2$. For $t \in \mathcal{O}(1)$, this would result in an instantaneous regret of order $(q\delta_{n_i})^2$, which would trigger a lag size transition. $\square_{\text{Claim 4.11}}$

Claim 4.12. The waiting regret (i.e., the regret incurred by Group 1 over the blue nodes and by Group 2 over the red nodes in Figure 4.3) is $\widetilde{\mathcal{O}}(T^{1/2})$.

Proof of Claim 4.12. Note that smaller lag sizes require more sampling: if $\delta^{(2)} < \delta^{(1)}$, then more time will be spend sampling f_2 at $x_2 - \delta^{(2)}$ and $x_2 - q\delta^{(2)}$ than will be spent sampling f_1 at $x_1 - \delta^{(1)}$ and $x_1 - q\delta^{(1)}$. In order to capture worst-case waiting regret for a group, we will consider the case where $\delta^{(2)} \leq \delta^{(1)}$ at all times and bound the waiting regret of Group 1.

Let $\delta_{n_1} > \cdots > \delta_{n_m}$ be the non-trivial lag sizes used by Group 2 (excluding δ, if δ is a non-trivial lag size). Note that δ_{n_1} is not necessarily δ, and $\delta_{n_{i+1}}/\delta_{n_i}$ is not necessarily q. For each i, let T_i denote the number of samples taken where δ_{n_i} is the lag size *or* in transitioning to $\delta_{n_{i+1}}$. Thus, letting T_0 denote the number of samples prior to δ_{n_1}, we have that $\sum_{i=0}^{m} T_i = T$.

We will now bound the waiting regret of Group 1 during these $m+1$ phases. First, consider the time period associated with T_i, for any $1 \le i \le m$. During this phase, Group 2 has already transition to a lag size of δ_{n_i} from a lag size of δ_{n_i}/q. Note that once this transition is made, the algorithm switches to optimizing over Group 1. Since the algorithm did not enter the combined phase in the previous round, it must be that the gradient in the Group 1 direction is of order δ_{n_i}. Thus, the total waiting regret for Group 1 is of order

$$T_0 + \sum_{i=1}^{m} T_i \delta_{n_i}^2 \ .$$

Now let us consider bounds on T_i, for $0 \le i \le m$. By Claim 4.11, we know that the number of gradient-scaled jumps that Group 2 takes during this phase is constant, and similarly the number of lags transitions (i.e., $n_{i+1} - n_i$) is also constant. It follows that $T_i \in \widetilde{\mathcal{O}}(\delta_{n_i}^{-4})$.

We can similarly bound T_0. As in the previous calculations, we will ignore constants for simplicity. To bound T_0, we first bound the number of gradient-scaled jumps before the current gradient is of order $q\delta$ (as this would trigger a lag size transition). To that end, the linear convergence rate of ADA-LGD (Alg. 8) implies that the gradient after t gradient-scaled jumps is of order $e^{-t}M^2$, where $M = \max_{(x,y)\in\mathcal{X}_F^2} |f_2'(y)|$. It follows that the gradient is of order $q\delta$ after $\ln \frac{M}{q\delta}$ jumps; since $\delta = 1/\log T$, the total number of gradient-scaled jumps before the first lag size transition is of order $\log\log T$. Finally, by Claim 4.11, n_1 is constant as well, which implies that $T_0 \in \mathcal{O}\big((\log T)^5 \log\log T\big)$.

Putting all of this together, the waiting regret for Group 1 is of soft order $\sum_{i=1}^{m} \delta_{n_i}^{-2}$. Letting x_{2,t_i} denote the first non-lagged iterates for Group 2 to admit a lag size of δ_{n_i}, we have that $|\nabla f_2(x_{2,t_i})| \in \Theta(\delta_{n_i})$. Thus we can rewrite the waiting regret for Group 1 as $\sum_{i=1}^{m} 1/|\nabla f_2(x_{2,t_i})|^2$. Finally, since we stop optimizing over Group 2 when its gradient is bounded in magnitude by $T^{-1/4}$, we can bound this further by $mT^{1/2}$. Since there can be at most logarithmically many gradient-scaled jumps before the gradient is of order $T^{-1/4}$, the waiting regret is $\widetilde{\mathcal{O}}(T^{1/2})$. $\square_{\text{Claim 4.12}}$

Since the combined function h is α-strongly convex and β-smooth, we have the same convergence rate for the combined phase as before. Moreover, since the algorithm does not erroneously enter the combined phase with high probability, the optimum of h is the same as the constrained optimum with high probability. Thus, the regret analysis of Algorithm 8 carries over. $\square$

Numerical validation

To validate the CFTD behavior of SCADA-LGD, we run it on a synthetic example. The functions being optimized are $f_1(x) = (x - .6)^2/.36$ and $f_2(x) = (x_2 - .1)^2/.81$, which are chosen so that

$$\min_{x\in[0,1]} f_1(x) = \min_{x\in[0,1]} f_2(x) = 0 \text{ and } \max_{x\in[0,1]} f_1(x) = \max_{x\in[0,1]} f_2(x) = 1.$$

A sample decision path for this input over $T = 10,000$ time periods is shown in Figure 4.3 (right), where the initial decision is $(0, 0)$, and the decisions increase in both coordinates over time. Note that Group 1 overshot its optimum slightly in this run, but the decision path did not overshoot its joint constrained optimum.

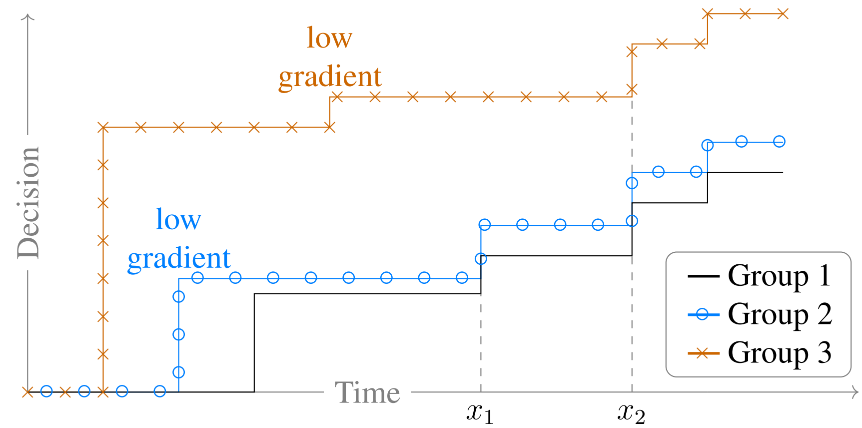


Figure 4.5: Illustrative decision paths of three groups under C^2-LGD (Algorithm 10). Points at which groups hit the low-gradient condition are marked. At point x_1, Groups 1 and 2 are combined, and at point x_2, Group 3 is combined with the other two groups.

4.3.4 Algorithm design and analysis for N groups

We now consider the multi-group setting beyond two groups. In this case, an overall approach similar to the two-group case can be used when perfect gradient feedback is available. The groups can take turns ascending along their coordinate until they either hit a constraint boundary or meet the local optimality condition $\nabla f_i = 0$. When all groups have stopped moving, each group that has stopped because it hit a constraint boundary can be associated with a group that has stopped because $\nabla f_i = 0$, by tracing the path of tight CFTD constraints leading up to this group (there could be more than one such group). We can thus partition the groups into disjoint clusters, where each cluster is associated with a set of groups that have stopped because $\nabla f_i = 0$. Each such cluster moves together from that point onwards, with the relative positions of the different groups in the cluster locked. We then continue the procedure with these clusters as the new groups and so on, until all current clusters are at their optimum.

However, in the case where only noisy bandit feedback is available, similar challenges arise as in the two-group setting in converting this high-level approach to a practical algorithm that incurs near-optimal regret. Since we anticipate the details of such an algorithm to be quite cumbersome, we instead consider a simpler approach where we use static lags instead of dynamic lags. In particular, unlike the two-group setting, where a decision for a group is allowed to increase despite meeting the low gradient condition by adaptively choosing a lower lag size, with a static lag, each group (or a cluster) meets the low gradient condition at most once. The algorithm initially considers each group to be its own cluster. Each cluster is optimized in turn sampling at the next feasible point or moving to the boundary if no feasible points exist. If no clusters can move and not all clusters are at a low gradient, then one of the high-gradient clusters is combined with one of its constraining low-gradient clusters, and the process continues. This process, which we call CYCLE-THEN-COMBINE LAGGED GRADIENT DESCENT (C^2-LGD), is described formally in Algorithm 10. A sample decision path of this algorithm for the case of three groups is shown in Figure 4.5.

We show below that this algorithm attains $\widetilde{\mathcal{O}}(N^3T^{2/3})$ regret. That said, we conjecture that an asymptotically better regret bound of $\widetilde{\mathcal{O}}(\sqrt{T})$ can be obtained using a dynamic-lag approach, where the latter $\widetilde{\mathcal{O}}$ hides dependence on N.

First, I introduce some notation that will be used in the multi-group algorithm (Algorithm 10), starting with the *succession function*, which cycles between the different clusters of groups.

Definition 4.2 (succession function)**.** Let $N \in \mathbb{N}$ and let Π be a partition of $[N]$. For any $A \in \Pi$, define the successor

$$\varphi(A;\Pi) := \begin{cases} \underset{B\in\Pi:\min B>\min A}{\arg\min} \min B & \text{if } \exists B \in \Pi \text{ with } \min B > \min A \\ \underset{B\in\Pi}{\arg\min} \min B & \text{else.} \end{cases}$$

Note that $\varphi(\,\cdot\,;\Pi)$ corresponds to the linear ordering of Π by the minimum elements of its blocks.

Next, we define the *feasibility distance*, which quantifies the extent to which a cluster A can move without violating any of the constraints.

Definition 4.3 (feasibility distance)**.** Let $N \in \mathbb{N}$, let $\emptyset \neq A \subsetneq [N]$, let $s : [N] \times [N] \to \mathbb{R}^{\geq 0}$, and let $x \in \mathbb{R}^N$ satisfy. Then the *feasibility distance* of A is $d(A; x) = \min_{i \in A, j \notin A} x_j + s(i, j) - x_i$.

With these definitions in mind, we outline our N-group algorithm, CYCLE-THEN-COMBINE LAGGED GRADIENT DESCENT (C^2-LGD) (Alg. 10). The high-level idea of the algorithm is to maintain a clustering (i.e., a partition) of the groups, optimize each cluster separately in coordinate-descent fashion, and combine clusters whenever none can move and doing so would loosen a tight constraint. These ideas are described in more detail below.

Making gradient-scaled jumps. As with all other algorithms presented in this paper, gradients are estimated using a lagged and non-lagged point. However, while ADA-LGD and SCADA-LGD used adaptive lag size, C^2-LGD uses a fixed lag size δ, which greatly simplifies algorithm design and analysis. In C^2-LGD, whenever a cluster's gradient becomes smaller than $\mathcal{O}(\delta)$, it can no longer move without risking breaking monotonicity, so it is no longer optimized in isolation (although it may be optimized further after being combined with another cluster).

Switching between clusters. Since all clusters have the same lag size, each cluster takes the same number of samples to make a gradient-scaled jump. Thus, there is no concern over one cluster obtaining copious amounts of regret while another cluster is sampling at a single point. We can therefore employ a simple trigger: switch to the next cluster after (1) a point has been sampled, or (2) the boundary has been hit.

Combining clusters. Suppose we have reached a point where no cluster can move. If all clusters are at a low gradient, then we have reached an approximate optimum, and no further movements are made. Otherwise, there are clusters at tight constraints. In this can, we argue that there must be a low-gradient cluster which is preventing a tight cluster from moving without violated the CFTD constraint. In this case, we combine two such clusters and continue.

Theorem 4.4. *Assume that $f(x) = \sum_{i=1}^{N} f_i(x_i)$ is α-strongly convex and β-smooth,* $\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathbb{R}^N} f(\mathbf{x}) \in (x_{\min}, \infty)^N$*, and the noise is mean zero, independent, and sub-Gaussian of bounded norm (assumption 4.2). Then* C^2-LGD *(Alg. 10) satisfies CFTD and, on input of $\delta = T^{-1/6}$, $\gamma = 1 + \frac{1}{\log T}$, and $p = T^{-2}$ is $\widetilde{\mathcal{O}}(N^3 T^{2/3})$-regret.*

Proof of Theorem 4.4. First, we argue that Algorithm 10 *run on a single group* would result in $\widetilde{\mathcal{O}}(T^{2/3})$ regret. To that end, we bound the pre-stopping regret and the post-stopping regret separately. Since the convergence rate is exponential, the number of queried points is bounded by a constant. Since each point is queried $\widetilde{\mathcal{O}}(\delta^{-4})$ times, the total regret from exploration is $\widetilde{\mathcal{O}}(\delta^{-4})$. Next, since the algorithm stops when the gradient is of order δ, the instantaneous stopping regret is of order δ^2. Since overshooting is avoided with high probability, the total regret is $\widetilde{\mathcal{O}}(\delta^{-4} + T\delta^2) = \widetilde{\mathcal{O}}(T^{2/3})$. Next, we return to the N-group setting. Before bounding the regret, we prove that the algorithm never encounters a situation where all clusters are constrained by each other.

Claim 4.13. Suppose the condition at line 6 (that no clusters can move and not all clusters are at a low gradient) is satisfied. Then either (1) some subset of clusters is locked in place with respect to each other (i.e., their joint feasible decisions form a line), or (2) some cluster must be at a low-gradient condition and constraining the movement of another cluster.

To prove this claim, suppose (1) does not hold, and suppose $C_1, \ldots, C_k$ are the current clusters. For each $j \in [k]$, let $i_j = \min C_j$ be a representative group from cluster C_j. We can therefore represent the constraints placed on C_j by linear inequalities on x_{i_j}; i.e., for each $j \neq m \in [k]$, there exist $b_{j,m} \in \mathbb{R}$ such that

$$x_{i_j} \leq x_{i_m} + b_{j,m}$$

is the constraint placed on C_{i_j} by C_{i_m}. Moreover, since the slacks $s(\cdot, \cdot)$ are nonnegative, $b_{j,m} \geq 0$.

Now consider the directed graph $G = (\{C_1, \ldots, C_k\}, A)$, where the arc (C_{i_j}, C_{i_m})

means that C_{i_j} is constrained by C_{i_m} (i.e., $x_{i_j} = x_{i_m} + b_{j,m}$). We now argue that this graph must be acyclic. For the sake of contradiction, suppose that there is a cycle $(j_1, \ldots, j_\ell)$; i.e.,

$$\begin{aligned} x_{i_{j_1}} &= x_{i_{j_2}} + b_{j_1,j_2} \\ &\vdots \\ x_{i_{j_{\ell-1}}} &= x_{i_{j_\ell}} + b_{j_{\ell-1},j_\ell} \\ x_{i_{j_\ell}} &= x_{i_{j_1}} + b_{j_\ell,j_1}. \end{aligned}$$

This implies that $x_{i_{j_1}} \geq \cdots \geq x_{i_{j_\ell}} \geq x_{i_{j_1}}$. If these inequalities are all equalities, this contradicts our assumption that (1) does not hold; otherwise, if one of the inequalities is strict, this also yields a contradiction. Thus, G is acyclic.

Now, choose any cluster C which has at least one out-going arc (which must exist, since not all clusters are at a low gradient). Find a maximal directed path beginning at C, and let (C_{i_j}, C_{i_m}) be the terminal arc. Since G is acyclic, there are no out-going arcs from C_{i_m}, which means that C_{i_m} is at a low gradient and is constraining the movement of C_{i+j}. This proves the claim.

The previous claim shows that the algorithm operates without getting stuck. Next, we bound the cluster-combining regret and stopping regret separately.

Regret from erroneously combining clusters

Claim 4.14. The total regret incurred due to erroneously combining clusters (i.e., combining clusters whose joint unconstrained optimum is in the interior of the feasible region) is of order $\widetilde{\mathcal{O}}(N^3 T \delta^2)$.

To justify Claim 4.14, first note that clusters are combined at most $N-1$ times throughout a run of the algorithm. We define $f_i^1 = f_i$ to be the initial group-specific functions. Each time clusters are combined, we define a new set of functions which are "close" to the previous set of functions, and for which the clusters were combined correctly.

Suppose that clusters have been combined k times, and the current partition of the groups is $\mathcal{C}_k = \{C_1^k, \ldots, C_{N-k}^k\}$. For any cluster C, we define $\min C$ to the the representative group of that cluster. Thus, there exists a sequence of gaps $\eta \in \mathbb{R}^N$ such that the function evaluation at a point $x \in \mathbb{R}$ for cluster C can be expressed as

$$f_C^k(x, \eta^k) := \sum_{j \in C} f_j^k(x + \eta_j^k).$$

Suppose we are combining clusters C and D at the representative points (x_t, y_t). Without loss of generality, suppose C is tight and D is at a low gradient. Then the slacks s and the current lockings induce linear constraints $x \leq y + b_{C,D}$ and $y \leq x + b_{D,C}$ for some $b_{C,D}, b_{D,C} \geq 0$, where x is the representative decision for cluster C, and y for cluster D. Since C is tight, we have that $x_t = y_t + b_{C,D}$, and since D is at a low gradient condition, $|(f_D^k)'(y_t, \eta^k)| \in \mathcal{O}(\delta)$ with high probability.

Let $\mathcal{X}_{C,D}$ be the polyhedron in $\mathbb{R}^2$ defined by the constraint imposed on C by D and vice versa (i.e., the gray region in Figure 4.6). Now let $(x', y') = \arg\min_{(x,y)} f_C^k(x, \eta^k) + f_D^k(y, \eta^k)$. With high probability, we have not overshot in either the C or the D direction. So, if $(x', y') \notin \mathcal{X}_{C,D}$, then we have correctly combined the clusters. Otherwise, $(x', y') \in \mathcal{X}_{C,D}$ (see Figure 4.6 for a depiction of this scenario). In this case, $|y' - y_t| \in \mathcal{O}(\delta)$ and $|x' - x_t| \in \mathcal{O}(\delta)$. So, letting (w_1, w_2) be the projection of (x', y') onto the facet $x = y + b_{C,D}$, and letting $(\rho, \rho) = (x', y') - (w_1, w_2)$, we have that $\rho \in \mathcal{O}(\delta)$.

Now let η^{k+1} be the new gaps after combining clusters C and D. To account for the $\mathcal{O}(\delta)$ error from combining C and D, we consider a new function

$$f_{C \cup D}^{k+1}(x, \eta^{k+1}) = \sum_{i \in C \cup D} f_i^{k+1}(x + \eta_j),$$

where $f_i^{k+1}(x) = f_i^k(x + \rho)$. For all other clusters E, we define $f_E^{k+1} = f_E^k$. Thus, with high probability, according to the functions $f_1^{k+1}, \ldots, f_N^{k+1}$, we have not erroneously combined clusters. Moreover, for any i and any $m \leq N-1$, we have that $f_i^m(x) = f_i(x+r)$ for some $r \in \mathcal{O}(N\delta)$. This will allow us to bound the regret of the original function f with respect to the modified function f^m.

To that end, suppose we arrive at a point where $\mathbf{x}_t$ is within $N\delta$ of the constrained optimizer of f^m. Then, letting z denote the constrained optimizer of f, we have that $\|\mathbf{x}_t - z\|_2 \in \mathcal{O}(N^{3/2}\delta)$, since the optima differ by at most $N\delta$ in each coordinate. So, the worst-case stopping regret in the case of erroneously combining lags is $TN^3\delta^2$.

Stopping regret

Claim 4.15. Suppose C^2-LGD has reached a point where each cluster has met the low-gradient condition. Then the regret incurred during the remainder of the time horizon is of order $\widetilde{\mathcal{O}}(NT\delta^2)$.

To justify Claim 4.15, suppose that the algorithm reaches a point where all the clusters $C_1, \ldots, C_k$ have reached the low-gradient condition. So, for all $i \in [k]$, letting $g_{C_i} = \sum_{j \in C_i} g^{(j)}$, we have that $|g_{C_i}| \in \mathcal{O}(\delta)$ with high probability. Let $f_{C_i}(x) = \sum_{j \in C_i} f_j(x+\eta_j)$ for the appropriate shifts η, and let $z^*_{C_i} = \arg\min f_{C_i}(x)$. Note that g_{C_i} is the gradient of f_{C_i}. Then, letting x_t denote the current point for C_i, we have that $\|x_t - z^*_{C_i}\| \leq \frac{\delta}{\alpha}$, and so

$$f_{C_i}(x_t) - f_{C_i}(z^*_{C_i}) \leq \nabla f_{C_i}(z^*_{C_i})^\top (x_t - z^*_{C_i}) + \frac{\beta}{2}\|x_t - z^*_{C_i}\|^2 = \frac{\beta}{2}\|x_t - z^*_{C_i}\|^2 \in \mathcal{O}(\delta^2)$$

Since this is true of all clusters, we have that $f(\mathbf{x}) - f(z) \in \mathcal{O}(N\delta^2)$, where $\mathbf{x}$ is the current point, and z is the optimum subject to the current lockings. By the above bound on regret due to erroneously combining groups, we have that

$$f(\mathbf{x}) - f(z^*) \in \mathcal{O}(N^3\delta^2),$$

where z^* is the constrained optimum.

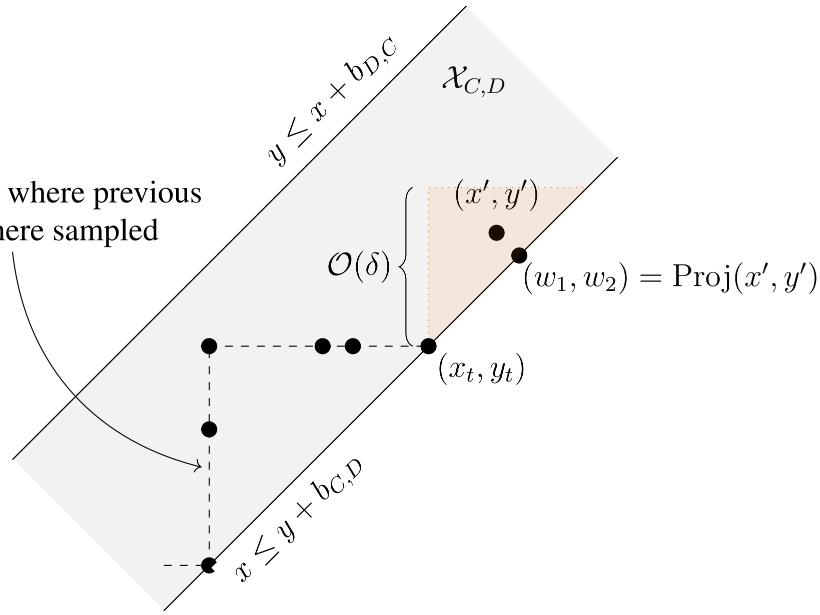


Figure 4.6: Illustration of an erroneous entering of the combined phase, relevant to the $\mathcal{O}(\delta^2)$ regret bound of Theorem 4.4. Here, $\mathcal{X}_{C,D}$ is the set of feasible points constrained by EFTD slacks on clusters C and D. In this scenario, the combined phase for clusters C and D is entered at (x_t, y_t) (thereafter, the iterates will lie on the tight constraint), despite the fact that the unconstrained joint optimum (x', y') for these two clusters is in the interior of $\mathcal{X}_{C,D}$. As discussed in the proof, one can ensure with high probability that (x', y') is $\mathcal{O}(\delta)$ away from (x_t, y_t) in both coordinates, implying that it is $\mathcal{O}(\delta)$ away from its projection (w_1, w_2) onto the facet $x = y + b_{C,D}$.

Total regret

In addition to the regret described above, we have the exploration regret of $\widetilde{\mathcal{O}}(N\delta^{-4})$ and the waiting regret of $\widetilde{\mathcal{O}}(N\delta^{-4})$. In sum, the total regret is of order $N\delta^{-4}+N^3T\delta^2$. Choosing $\delta = T^{-1/6}$, we get a regret bound of $\widetilde{\mathcal{O}}(N^3T^{2/3})$. □

4.4 Discussion: Unfairness and Perceptions thereof

Notions of comparative fairness, such as individual fairness (Section 2.1), are based on the idea that similar individuals should receive similar decisions. In online settings, where outside factors can change over time, this is arguably not necessarily the case. For instance, someone of context c_1 may be offered a price of \$1 today; the next day, there may be a supply shortage that causes someone of context c_2 to receive a price of \$3. From the perspective of c_2, this may seem unfair, but the actual unfairness of this situation is debatable. In general, there may be a difference between *perceptions of unfairness* and unfairness itself, since perceptions of unfairness are often relative [101, 102, 103, 26, 104].

In deciding on a temporal fairness constraint, one can try to account for these differences. For example, if one wishes to allow for price increases to account for inflation, then the monotonicity constraints could be relaxed to the following, for some $s(i, i) < 1$.

$$x_{i,t+1} \geq s(i, i)x_{i,t} \text{ for all } i \in [N] \text{ and } 1 \leq t < T.$$

It is important to identify fairness goals in every decision-making scenario and not blindly apply a popular fairness constraint.

4.5 Open Questions

I end this chapter with several open questions stemming from Section 4.3.

1. Given the gap between the lower bound of $\widetilde{\mathcal{O}}(\sqrt{T})$ and the upper bound of $\widetilde{\mathcal{O}}(T^{2/3})$ for a general number of groups, can this gap be closed? In particular,

   Does there exist a $\widetilde{\mathcal{O}}(\sqrt{T})$-regret algorithm for bandit convex optimization of a smooth, strongly convex, separable, N-dimensional function satisfying CFTD?

2. **Multiplicative slacks.** In many scenarios, multiplicative allowable disparities in treatment could be more appropriate than additive disparities. For example, in the context of pricing, a firm may want to price an item while ensuring that the price disparity between the youth and the general population is not more than 25%. In particular, consider slack functions $\bar{s}(\cdot, \cdot) : [N] \times [N] \to \mathbb{R}_{\geq 0}$ defined on ordered pairs of groups, and consider the following notion of comparative fairness with multiplicative slacks.

Definition 4.4 (CFTD with Multiplicative Slacks))**.** We say that a decision sequence $(x_1, \ldots, x_T) \in \mathcal{X}^{TN}$ satisfies *CFTD with multiplicative slacks* if the following inequalities hold:

$$x_{i,t} \geq \bar{s}(i,j)x_{j,t'} \text{ for all } i,j \in [N] \text{ and } 1 \leq t' \leq t \leq T. \tag{4.24}$$

The following question is open:

For any N, does there exist a $\widetilde{\mathcal{O}}(\sqrt{T})$-regret algorithm for bandit convex optimization of a smooth, strongly convex, separable, N-dimensional function satisfying CFTD with multiplicative slacks?

One setting where our current results extend is the case where there is a function $m : [N] \to \mathbb{R}_{>0}$ such that $\bar{s}(i,j) = m(j)/m(i)$. In this case, (4.24) effectively requires that $x_i m_i \geq x_j m_j$. In this case, one can redefine the decision space for group i as $y_i = x_i m_i$. Our algorithms then readily apply to optimizing $\mathbf{y}$ in this setting.

3. **Time decay.** Certain fairness goals may require consistency with recent decisions, but not on decisions made long ago; as such, there is room for adapting CFTD to better fit specific objectives. For example, one may wish to incorporate a time-decay element to the comparative fairness constraint, which can be achieved by having the slack functions s be dependent on the difference in the times at which the decisions were received by the two groups. In particular, we may have that $x_{i,t} \geq x_{j,t'} - s(i,j,t-t')$ for all $t \geq t'$, where the slack function now also takes the time difference $t-t'$ as input. If s is assumed to be increasing in the time difference, this would allow for greater changes in decisions over time.

   What is the optimal regret for an algorithm for bandit convex optimization of an N-dimensional function satisfying these time decay constraints?

Note that with these time decay constraints, it may be easier to drop the smoothness and strong convexity assumptions and still achieve sublinear regret, since overshooting can potentially be undone.

4. **Beyond strong convexity.** The analysis (and algorithms) of this chapter relied on the objective function being strongly convex.

   Can the results of this chapter be extended beyond the strong convexity assumption?

   As discussed earlier, for the simplest case of single-dimensional optimization, [98] and [99] have shown that a significant impact on the achievable rate of regret is inevitable under the monotonicity constraint if only unimodality and Lipschitzness of the cost function are assumed. While it seems that smoothness is necessary to obtain the near-optimal regret rate of $\widetilde{\mathcal{O}}(\sqrt{T})$ in our setting (since smoothness is fundamental to controlling overshooting of the optimum), we are hopeful that the strong convexity assumption can be relaxed (to, e.g., just assuming convexity) without impacting the regret guarantee.

Algorithm 10: Cycle-then-Combine Lagged Gradient Descent (C^2-LGD)

input: Number of groups N, sub-Gaussian norm bound $E_{\max}$, $n(d) = \frac{64E_{\max}^2 \log \frac{2}{p}}{C\alpha^2 d^4}$, smoothness parameter β and strong convexity parameter α of $f(x_1, \ldots, x_N) = \sum_{i=1}^{N} f_i(x_i)$, time horizon T, non-negative CFTD slacks s, $x_{\min}$, γ, lag size δ

Initialize queue $Q_i \leftarrow \emptyset$ for $i \in [N]$ `// will contain points to query`

Initialize the partition $\Pi = \{\{1\}, \ldots, \{N\}\}$ of groups, and set active$_A$ = Yes and $g_A = -\infty$ for each $A \in \Pi$

For any $A \subseteq [N]$, define $f_A(x) = \sum_{i \in A} f_i(x)$; initialize $A = \{1\}$

Add $x_{\min}$, $x_{\min} + \delta$ to Q_i for $i \in [N]$

while *fewer than T samples have been taken* **do**

- **if** active$_B$ = No *for each cluster $B \in \Pi$ (i.e., no clusters can move)* **then**
  - **if** *each cluster is constrained by some other cluster* **then**
    - Find clusters C, D which are locked in place with respect to each other (see Claim 4.13)
  - **else**
    - Find clusters $C \neq D \in \Pi$ such that C is at a tight constraint imposed by some group in D, and D has met the small gradient condition (see Claim 4.13)
  - Update $\Pi \leftarrow \Pi \cup \{C \cup D\} \setminus \{C, D\}$
  - Let $A = C \cup D$, and set $g_A = -\infty$ and active$_A$ = Yes
  - Let $i = \min A$, and add $x_i, x_i + \delta$ to Q_i `// add points to the representative group's queue`
- Let $i = \min A$ `// choose a representative group`
- $\mathcal{X}_A = [x_i, x_i + d(A; x)]$ `// current feasible points for cluster A`
- **if** *Q_i contains any elements in $\mathcal{X}_A$* **then**
  - Let $x_i \leftarrow \min Q_i$, remove x_i from Q_i, and update x_j (for $j \in A \setminus \{i\}$) accordingly (i.e., letting $d = \min Q_i - x_i$, we set $x_j \leftarrow x_j + d$ for all $j \in A$)
  - Update active$_B$ for $B \in \Pi$ as necessary (formerly tight clusters may now be active; in particular, if B is at a tight constraint imposed by some group in A and B is not at any tight constraints imposed by any group in any other cluster, then B will no longer be at a tight constraint)
  - Sample $n(\delta)$ times at x_j for $j \in A$
  - **if** *x_i is a non-lagged iterate for cluster A* **then**
    - Update the gradients $g^{(i)}$ for $i \in A$ and set $g_A = \sum_{i \in A} g^{(i)}$
    - **if** $-\frac{1}{\beta} g_A \geq (1 + \gamma)\delta$ **then**
      - Add $x_i - 2\delta - \frac{1}{\beta} g_A$ and $x_i - \delta - \frac{1}{\beta} g_A$ to Q_i
    - **else**
      - Set active$_A$ = No
- **if** *Q_i contains no elements in $\mathcal{X}_A$* **then**
  - Set $x_i = \max \mathcal{X}_A$ and update x_j (for $j \in A$) accordingly
  - Set active$_A$ = No and update active$_B$ for $B \in \Pi$ (formerly tight clusters may now be active)
- **if** $-\frac{1}{\beta} g_B < (1 + \gamma)\delta$ *for every cluster $B \in \Pi$* **then**
  - Exit the while loop and remain at point x for the remaining iterations
- $A \leftarrow \varphi(A; \Pi)$ `// move to the next group`

CHAPTER 5
ALGORITHMS AND THE LAW

This chapter contains excerpts from [82] and includes joint work with Swati Gupta and Deven Desai.

Many important decision-making processes are subject to law and policy. That said, technology often evolves faster than the law, and it is often unclear how existing laws affect new technologies. In Section 5.1, I provide a legal analysis of the algorithms presented in Chapter 3 with respect to U.S. law and caselaw. Next, I highlight open questions regarding pricing algorithms and the law in Section 5.2.

5.1 Hiring

For example, decisions made in the hiring process are subject to anti-discrimination law in the U.S. This is the case regardless of whether an algorithm was used to make the decisions. However, whether or not the decision-making process is automated affects the way in which the law interacts with decision-making. For instance, when using algorithms to make decisions, the decision-making process is explicit and documented, and thus can potentially be audited. Unautomated decision-making, on the other hand, is often a nebulous process in which it is difficult to identify intent. The ways in which algorithmic decision-making is affected by—and is affecting—the law is an emerging issue of great importance. In this chapter, I partially answer the following question:

Are the poset secretary algorithms of Chapter 3 legal in the context of applicant screening?

5.1.1 Motivation from Industry

How employers identify whom to interview and then hire has important effects across society. Employment significantly affects access to healthcare, continuing education and, therefore, quality of life. The benefits of employment are not, however, evenly distributed across race and gender categories in the United States. After George Floyd's death, companies acted to address racial injustice by making public statements, donations to support racial equality, and Juneteenth a company holiday [105]. Several companies went further. Microsoft announced a $150 million investment to improve diversity including setting a goal of doubling the number of "Black and African American people managers, senior individual contributors and senior leaders" in the United States by 2025 [80]. Wells Fargo made a commitment to "double Black Leadership" by 2025 and "will evaluate senior leaders based on their progress in improving diversity and inclusion in their areas of responsibility, in addition to other efforts" [80]. Google has set a goal of having 30% of its leadership from "under represented groups" by 2025 [106]. Boeing seeks to increase representation of "Black employees by 20% while boosting other underrepresented groups over the next three years" [106]. Adidas announced plans to fill at least "30% of new positions with black or Latinx people" [107]. Yet, both Microsoft and Wells Fargo received letters from the Labor Department's Office of Federal Contract Compliance Programs (OFCCP) due to concern that the plans may discriminate based on race [105]. At the same time, the OFCCP announced a settlement with Microsoft in September 2020 for $3 million back pay and interest to address hiring disparities "against Asian applicants" for several positions from December 2015 to November 2018 [108]. The two OFCCP positions clash and appear to create a world where inaction opens the company to litigation, if not breaking the law, and corrective action creates the same risks. One might argue that the recent OFCCP inquiries were peculiar to the Trump administration's approach to this area of law and not something the current administration would pursue. Administrations, however, change and a new one might follow the Trump approach. Regardless of who is in the White House, legal activism to challenge steps taken to address diversity or challenge discriminatory results are not likely to go away.[1]

The reason this challenge is not likely to go away is that a company may be pursuing diversity goals and/or be addressing affirmative action plans; but the two are not the same, and the difference matters [109]. As the Equal Employment Opportunity Commission explains in "Section 15 Race and Color Discrimination" of its Compliance Manual, diversity can be understood as "a business management concept under which employers voluntarily promote an inclusive workplace" [110]. Companies have pursued diversity to attract talent and gain "a competitive advantage" [110]. In contrast, affirmative action refers to "those actions appropriate to overcome the effects of past or present practices, policies, or other barriers to equal employment opportunity" [111]. Such steps may occur because of a court order, negotiated settlement, or government regulation [110]. Employers may also use a voluntary affirmative action plan "in appropriate circumstances, such as to eliminate a manifest imbalance in a traditionally segregated job category" [110]. There is a conceptual and practical link between diversity goals and affirmative action. A company may pursue diversity "for competitive reasons rather than in response to discrimination" and "such initiatives may also help to avoid discrimination" [110]. As the legal status of diversity plans is unclear, methods to support both options are needed.

As another motivation, companies may want to see whether they are missing hiring and talent opportunities. Companies can be stuck in an equilibrium because they rely on, or exploit "old certainties," rather than explore "new possibilities" [112]. This exploration/exploitation trade-off began in organizational business literature but has become a significant part of how the machine learning community thinks about understanding information [113]. As a matter of best organizational and ML practices, companies need ways to explore new candidate pools.

[1] As Primus has explained, "If equal protection requires the law to be thoroughly colorblind, then a statutory doctrine that requires racial classification and makes liability turn on the status of groups considered collectively is an equal protection problem."

Regardless of the motivation behind a company plan, there is a steady drumbeat for algorithmic transparency, especially in employment and admissions contexts [114]. Thus an entity may have to or wish to reveal the process at some point. In either case the entity would want to show that their process is sound from both a mathematical and a legal view. These issues could push any company to avoid steps to address diversity because of litigation risks, both real and perceived. Although some scholars argue that the use of machine learning would constitute a valid business necessity claim so long as the target variable is job-related, thus rendering the question of equality of outcomes irrelevant, debates about which actions are and are not allowed to address diversity persist, especially when using an algorithmic approach [115]. Simply put, when entities wish to be proactive regarding diversity, potential discrimination, or wish to explore whether they have missed opportunities in hiring talent [69], they will need a path that passes muster against a range of challenges.

This paper thus seeks to offer techniques and legal analysis to enable companies to pursue legal and ethical hiring goals and face this question: *How to improve equal opportunity and employment practices without crossing into arguably illegal discriminatory practices?* The ideas discussed here are general and key takeaways can be applied to several stages in the hiring pipeline. That said, this paper uses the screening stage of employment to exemplify methods and analysis and offer one way to attack the general problem.

5.1.2 Algorithms and the Hiring Process

Employers want to hire a great workforce, but reaching and assessing the full viable range of potential employees poses problems. Many parts of the hiring process use algorithms as a way to manage and sort candidates. The practice can be traced back at least 40 years [116, 117]. The problem is that there are a number of junctures in the hiring pipeline at which bias can affect decisions, as depicted in Figure 5.1. Job advertisements on various platforms can be targeted at specific audiences [118, 119]. Application rates can differ across groups due to presumed employer bias [120]. Data-driven tools for evaluating résumés can be

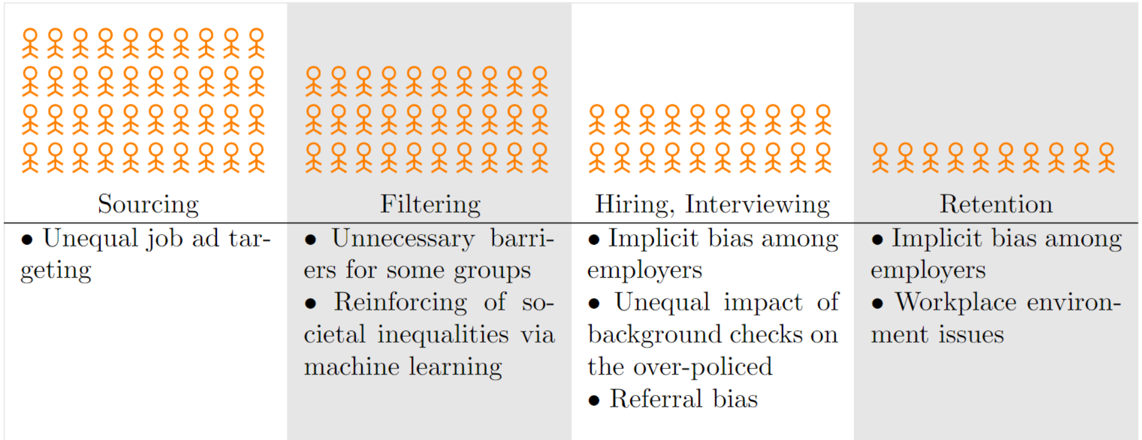


Figure 5.1: Some ethical concerns in various stages of the employment pipeline.

biased due to inequalities in training data [121], imbalance in data [122], or differences in false positive/negative error rates in prediction algorithms leading to bias as a *downstream effect* [123]. Referral hiring can lead to favoritism [124]. Customer evaluations of freelancers can adversely impact certain groups [125]. Final hiring decisions can be influenced by human biases of the hiring committee [126]. After going through the hiring pipeline, candidates also see a significant difference in salaries offered [58], and retention rates can differ dependent on the work environment [127]. Indeed, societal biases are pervasive and can affect decisions made by experts [128].

In addition, when automated systems are used at any stage, missed opportunity (false negatives) with respect to minority candidates is often shrugged off as an artifact of the prediction model, necessary for overall accuracy [129]. These models often train on historic data, which can depict imbalanced selection rates across different groups of candidates, and these trends can be learnt by automated methods [130, 131]. History can dictate future actions. In short, existing pipeline practices can reiterate and increase disparity in opportunity and outcomes. Although the hiring pipeline can be improved in many places, we find the screening stage to be particularly ripe for improvement, and we therefore focus the article on this stage for the reasons outlined below.

First, data-driven methods, by their nature, can pose a problem. Seemingly objective methods interact with real-world data, and so automated decisions can *reflect* and therefore, *reinforce* societal inequalities [132, 133]. Even when there is no intent to discriminate, and the decision system uses the same data and applies the same rule to all, there may be a disproportionate effect on a protected class (i.e., groups protected by law from discrimination, such as those defined by sex, race, age, etc.) [134, 135, 136]. The problems in screening map to the more general ones present when using data-driven decision-making in hiring. So, screening is a good lens through which to investigate the concerns around using algorithms and data in the employment context in general.

Second, algorithms are already used for screening applications. Such automated methods offer numerous advantages: speed, cost-effectiveness, potential objectivity, and uniformity in process. These properties may seem desirable at first glance from an ethical and fairness perspective; consistency in decisions is often a good thing, and a lack of human involvement would seem to minimize the role of implicit bias in hiring decisions [137]. Thus, automated methods have become commonplace in screening. Adjusting algorithmic techniques may be a more palatable idea and more feasible in an industry currently using automated processes than using algorithms in a heretofore un-automated process. New algorithmic interventions may, therefore, be more likely to be applied in practice.

Third, changes at early stages of the hiring pipeline are vital to address later bias. Changes at later stages are only meaningful if they act on a diverse pool of candidates. Without a diverse candidate pool at those stages, efforts to address bias become empty theater, because there will be few to no candidates from underrepresented groups for which the changes would help. As such, we focus specifically on automated screening processes: *how should applicant-screening methods be developed?* These algorithmic tools should be designed with the goal of (a) selecting applicants of a desired quality, (b) satisfying some agreed upon fairness criteria, and (c) adhering to US anti-discrimination law.

5.1.3 Biases in Data

We broadly refer to systematic inconsistencies in data which adversely affect certain groups as "bias." The first step in reducing discrimination is to understand the source of this bias. Unfair decisions can stem from many places, and identifying the origins of the bias allows for precise interventions. In the hiring process (automated or otherwise), applications will typically be assigned a score, thus allowing comparisons of applicants based on a single number or with respect to a single ranking of candidates [25, 138]. This evaluation metric can be hard-coded into an algorithm or developed dynamically, and in either case, can be unfair. A natural question is whether we can model this bias precisely and account for it within the algorithms to make them justifiably (provably) fairer.

Bias in evaluations can take different forms and be observed in different ways. For instance, a screening algorithm developed, *but not employed*, by Amazon penalized résumés which included the word "women's" due to data of past hiring trends in the company [11]. This algorithm penalized, for example, those who attended all-women colleges, and rewarded vocabulary typically used by men. In a similar vein, an empirical study showed that science faculty's assessment of résumés varied dependent on the gender of the student [58]. These are fairly blatant examples of discrimination, as toggling a protected attribute results in different treatment. Note that this form of unfairness—while blatant—can be hard to observe in practice, as applicants are never truly identical but for a small number of attributes.

Many cases of bias in evaluations, however, are more nuanced. Consider using SAT scores to screen candidates—a practice employers such as McKinsey, Bain, Goldman Sachs, and Amazon have been known to use even for candidates with advanced degrees [139, 140, 141]. Studies show that even if students are equally able to perform well on a test, if the test is announced to exhibit differences across groups, students in a negatively stereotyped group perform lower than the students in a non-stereotyped group [56]. Another study from 2013 shows that SAT scores are correlated with family income, potentially pointing to issues of access [142]. Inside Higher Education looked at SAT scores in 2015 and found that despite fee waivers and increased efforts to provide support and tutoring to low-income families,

> "In each of the three parts of the SAT, the lowest average scores were those with less than $20,000 in family income, and the highest averages were those with more than $200,000 in income, and the gaps are significant. In reading, for example, the average for those with family income below $20,000 is 433, while the average for those with income of above $200,000 is 570."

Thus, compared to 2013, gaps in performance with respect to racial groups not only persisted but increased. This problem with SAT scores is further evident in a recent study by Faenza et al. [143], which showed a shift by approximately 200 points in SAT scores from schools with different *economic need indices*. Thus, an employer using SAT scores appears neutral but sets up a *pre-selected* pool.

These issues regarding bias in data raise important *design* questions for algorithmic intervention. When designing a decision-making algorithm, can we control for bias in historic data (thus avoiding Amazon's situation discussed above)? In other words, what steps can be taken to control for historic, economic, and/or social factors that are known to skew seemingly objective metrics such as the SAT?

5.1.4 Approaches to addressing bias to date and their limits

A variety of algorithmic techniques have been proposed for coping with biased data and improving fairness, from pre-processing techniques which involve modifying data before feeding it to an algorithm [144]; to in-processing techniques, which modify the algorithm itself [145, 146]; to post-processing techniques, which modify decisions made by an algorithm after the fact [147, 148]. Current computer science literature highlights that merely scrubbing protected class information from an application may not help mitigate existing biases [149], and that algorithms have to use protected information to fix existing biases in data [7]. Using protected information, however, may put the hiring process at odds with anti-discrimination law. Other prevalent approaches include iteratively removing data which is correlated with protected information [146]; such approaches, however, may remove highly predictive information.

Algorithmic bias mitigation refers to the design of algorithms which perform well despite uncertainties about candidates' qualifications. This encompasses, for example, the design of procedures to select qualified candidates given biased data, or the design of algorithms which provably satisfy some notion of fairness. As discussed earlier, bias in evaluations can render bias-agnostic methods suboptimal [69, 30, 150, 7]; at the same time, imposing constraints such as demographic parity (i.e., proportional selection from different demographic groups) can hinder performance in some cases [31], which points to potential trade-offs between bias mitigation and quality of selections. In our approach, we will take the view of algorithmic bias mitigation, given fine-tuned uncertainties in the evaluation of each individual.

Algorithmic Bias Mitigation. Attempts to mitigate bias often begin with an understanding of the nature of the bias, or in other words, the inconsistencies in measurement of the ability of candidates. Mitigating the impact of such inconsistencies is an instance-specific endeavor; no cure-all exists. Nonetheless, there is theoretical work on mitigating bias under various mathematical assumptions. For instance, attempts have been made to address miscalibration of evaluations between multiple evaluators [151], and techniques have been developed for cases where some information is known about how biased each evaluator is in each evaluation [152]. In general, mathematical techniques can be developed as long as some assumptions on bias are made.[2] Certain "coarse" sources of bias seem to be prevalent across demographic groups, and algorithms can be designed with these in mind. One

[2] This points to multiple issues in bias-mitigation. First, the assumptions on bias are difficult to justify empirically, as "ground truth" is seldom available (for example, the true ability of a candidate is never truly known, especially for candidates who are not hired). Second, it is difficult to assess bias-mitigation techniques for a similar reason: if one does not know the ground truth, then it is hard to quantify how good any decision is.

might say these are the first approximations to incorporate the knowledge of large trends visible broadly across demographic groups, such as are seen in SAT scores discussed earlier [143]. Addressing these coarser sources of bias from a theoretical point of view can provide insight in dealing with other forms of bias.

A recent mathematical model that captures the dependence of errors in testing over groups is the *group model* of bias. The model is based on the empirical work of Wennerås and Wold [153], and was introduced by Kleinberg and Raghavan in the context of offline selection (e.g., applicant-screening) [69], further studied by Salem and Gupta [150], Faenza et al. [143], and Blum and Stangl [154] in the context of selection problems. This model assumes that bias is fairly consistent within each demographic group, and thus evaluations offer more accurate rankings within each group, but not across the groups. For example, once one accounts for difficulties in comparing one demographic group to another, there may be no way to confidently compare an 90% attained by a white male scholar Adam to a 85% attained by a Latina scholar Tia. But one can compare Adam against another white male scholar John with 83%, and note that Adam is better.

This model is at the same time appealing and dissatisfying in its simplicity. It is appealing in the sense that the model sheds light on best practices when the data is biased consistently for certain groups. That consistency indicates that information about group membership alone allows selection algorithms to reduce bias in selections. It is dissatisfying, however, in its coarseness, as it ignores intra-group differences in testing/evaluation errors and ignores any potential comparisons between groups. Adding to the example above, let us say that Tia also belongs to a low-income family, and we want to compare Tia to another Latina scholar May (not from a low-income family). This model does not account for such confounding variables of socio-economic status. Follow-up work by Celis et al. [70] proposed a multiplicative model of bias in the context of rankings, wherein candidates in the intersection of different groups face a consistently higher bias. This approach, however, again equalizes the amount of bias within each smallest "unique" group (e.g., male, white, and age above 45 or lesbian, Asian, aged 39). It may not be okay to equalize the experience of every male, white person above the age of 45 or of every lesbian, Asian, under age 50. The underlying problem with this is the assumption of group membership, which may not even be accurate. Indeed, whether a Chinese Asian, an Indian Asian, and a Filipino Asian faces the same amount of bias, and so should be treated the same, seems unlikely.

Current Industrial Practices. How then do companies actually hire candidates, while reconciling with anti-discrimination laws and biases in the hiring pipeline? In a recent survey, the only specific public claim made by vendors of pre-employment assessments was adherence to the 4/5ths rule—outlined in the 1978 Uniform Guidelines on Employee Selection Procedures—which requires that group-specific selection rates of any pre-screening are all within a factor of $4/5$ of each other [54]. Yet this approach is coarse as it is agnostic to quality of candidates. Applying a 4/5ths rule in selection up front (e.g., as the current practice in the industry suggests [54]) does not change the perceived potential of candidates, nor account for uncertainties and biases in the data systematically. It can therefore simply set up the underrepresented group's candidates for failure, and lead to resentment and enlivening of negative stereotypes [155, 156].

The trade-offs in algorithmic approaches track legal issues. If an employer uses an algorithmic tool to evaluate and screen candidates, the employer may face legal challenges depending on the outputs of the tool. A likely challenge is that the tool created illegal disparate impact. Disparate impact addresses when "facially neutral policies or practices have a disproportionate adverse effect or impact on a protected class" [157, 158]. The disparate impact doctrine is thus supposed to address situations where intent is not at hand or cannot be ascertained [159]. In short, outcomes based on unaware algorithms may fit quite well with disparate impact challenges, because unaware algorithms are facially neutral, may lack intent to discriminate, and nonetheless yield statistically discriminatory results.[3]

[3]Despite the fact that the 4/5 rule is mentioned as evidence of disparate impact in the 1978 Uniform Guidelines on Employee Selection Procedures, there is no precise quantification of disparate impact. The 4/5

The possibility of a disparate impact claim leads to an obvious approach. An employer may design a more aware algorithm that takes protected class status into account. However, this approach may run into a disparate treatment challenge. Disparate treatment is the legal doctrine that prohibits intentional use of race or other protected classes in making an employment decision. Thus, we return to the paradox described above, because it seems that an employer is trapped between using facially neutral systems that reflect systemic and historically conditioned, biased results or facing lawsuits for using aware systems to mitigate such effects. This paradox is exacerbated by current legal scholarship debating what algorithmic interventions to address bias, if any, are allowed and the implications of the lawsuit *Ricci v. DeStefano*, in which an action by the City of New Haven that tried to account for disparate impact of an administered promotion test led to litigation that was decided against the city. In Section 5.1.5, we outline the *poset approach*, which we argue provides a way to solve the hiring paradox. Section 5.1.6 turns to an in-depth discussion on the takeaways from *Ricci v. DeStefano* and explains how the poset approach fits within legal rules so that one can use a bias aware approach to hiring and yet maintain individualized assessments of candidates.

5.1.5 A new approach: coping with uncertainty using partial orders

As discussed in Section 5.1.3, coping with uncertainties in data is a fundamental problem in applicant screening systems, as well as in data-driven decision-making more generally. In this section, we will discuss one method, called the *poset approach*, for applicant-screening in the face of uncertainty which has emerged recently in the computer science literature [150]. In Section 5.1.6, we will use this approach as a vehicle for discussing the legality of algorithmic bias mitigation in hiring.

Consider the following scenario: there are three candidates A, B, and C, with ability scores of 82, 68, and 67, respectively, and you wish to grant interviews to two of them. The

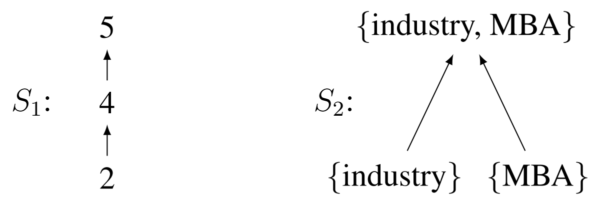
rule is often used as a trigger for litigation, but other statistical tests have been used in courts as well [160, 161].

ability scores are known to be a strong predictor of job performance, but are only known to be accurate up to 3 points. In this case, there is a significant chance that C is a better candidate than B, but the utilitarian approach of selecting the highest-scoring candidates would routinely select A and B. The core idea behind the poset approach is that the latter approach is unfair to C, or more generally, that ignoring uncertainty can result in unfair decisions. In other words:

> *Some applicants, due to insufficient or inaccurate data, cannot be reliably ranked. The solution need not involve producing a (possibly inaccurate) ranking. Instead, allowing for partial rankings can open the door to fairer decisions.*

The poset approach, which we explain in more detail below, makes use of a mathematical structure called a *partially ordered set*, or *poset*, which can be used to encode uncertainty in ordinal information. Consider, for example, a set $S_1 = \{4, 2, 5\}$ of true hirability of three candidates (which is often not observable in practice). This set is called *totally ordered* since any pair of the scores can be ordered (i.e., ranked) with respect to the relation $\leq$. In other words, we can rank the scores: $2 \leq 4 \leq 5$, thus inducing an order amongst the candidates.

However, in practice, one cannot observe directly how good a candidate might be at their job. This is where partil orders can help us. Intuitively speaking, one can think of a partial ordering as a set of comparisons, which may not cover all pairs of candidates (i.e., a total order with some comparisons missing). For example, consider a candidate A who has experience in industry, consider a candidate B who has experience in industry *and* who has an MBA, and a candidate C who has an MBA. Considering these traits as binary (yes/no) attributes, one can represent their qualifications as the set $S_2 = \big\{\{\text{industry}\}, \{\text{MBA}\}, \{\text{industry}, \text{MBA}\}\big\}$. From the given information, one might rank B above both A and C, since B is qualified with respect to both measures, and the other candidates are only qualified with respect to one. However, A and C might be considered *incomparable*, since their qualifications are complementary. In this case, S_2 is a partially ordered set, but not a totally ordered set. To be precise, a relation $\preceq$ is a partial order on a set S if three conditions hold for all $a, b, c \in S$: (1) $a \preceq a$; (2) if $a \preceq b$ and $b \preceq c$, then $a \preceq c$; and (3) if $a \preceq b$ and $b \preceq a$, then $a = b$. One can check that all these properties are satisfies for the set S_2. A poset is often visually depicted using its *Hasse diagram*, which is directed graph in which edges represent orderings. For example, the Hasse diagrams for S_1 and S_2 are as follows:



Note that Hasse diagrams omit redundant edges: even though $2 \leq 5$, the edge $2 \to 5$ is not included, since it is implied by the edges $2 \to 4$ and $4 \to 5$.

The *poset approach* is the process of (1) forming a partial ranking (i.e., a partial order) of the candidate pool based on uncertainties, inaccuracies, or biases in data, and (2) making selections based on this poset. By making selection decisions in this way, one can concretely take uncertainty into account and, say, avoid routinely harming candidate C in the example above. This can lead to bias mitigation in cases where the evaluation metric is biased against a certain group; e.g., if a group is underrepresented in training data and experiences large errors in the resulting ML model, the poset approach can confer benefit of the doubt to those underrepresented candidates.

We next illustrate how posets can model uncertainty using two examples.

Example 5.1. The poset approach can illustrate how one can account for uncertainties while also avoiding prohibiting discrimination based on gender.[4] Using the poset approach, one may incorporate demographic context of the candidates and quantify uncertainy in their evaluations (either by directly observing the context, or by unsupervised methods such as

[4] As recently as June 15, 2020, the Supreme Court of the United States ruled that the Title VII of the Civil Rights Act prohibits discrimination on the basis of sexual orientation and gender identity [162, 163].

clustering). Suppose that in a training dataset, nonbinary candidates are underrepresented, and as a consequence have high variance in errors in the prediction model. One may find that a nonbinary candidate Max has a wide score range of 80-90% (e.g., due lack of training data on nonbinary candidates), another male candidate Adam has a score between 85-87%, and a third female candidate Trisha has a score between 92-95% (see Fig. 5.2). Now, using only the score ranges to compare candidates, Trisha compares favorably to Max, but it is unclear if Max is more qualified than Adam as their ranges overlap. In this case, we can think of Max and Adam as mutually incomparable. The poset approach therefore allows for <u>individualized treatment</u> of inconsistencies in data processed.

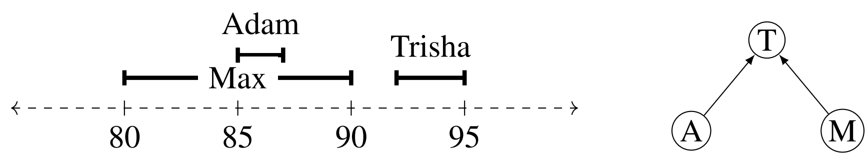


Figure 5.2: Score ranges and resulting Hasse diagram for the scenario in Example 5.1.

Example 5.2. Suppose that three candidates are to be selected based on two attributes: work experience and college GPA. You have set cutoffs for each of these attributes and only wish to select candidates exceeding each cutoff. See Figure 5.3 for a depiction of the candidate pool, where each color represents a particular demographic group. Let the colored areas around each candidate node represent a "confidence region;" i.e., with some high degree of confidence, the candidate's latent ability lies in the drawn region. Note that we can infer partial rankings from these confidence regions in a similar way to Example 5.1: if the confidence region of candidate A is strictly above and to the right of the confidence region of candidate B, then A is ranked above B.

Using only raw scores, only the two blue candidates meet the cutoffs. However, taking confidence regions into account, we see that the two green candidates might meet the cutoffs as well. How, then, should one choose three candidates among the green and blue ones? One way to do so is to look at the partial ranking induced by the confidence regions (shown using arrows in Figure 5.3). In this partial ranking, there are three candidates

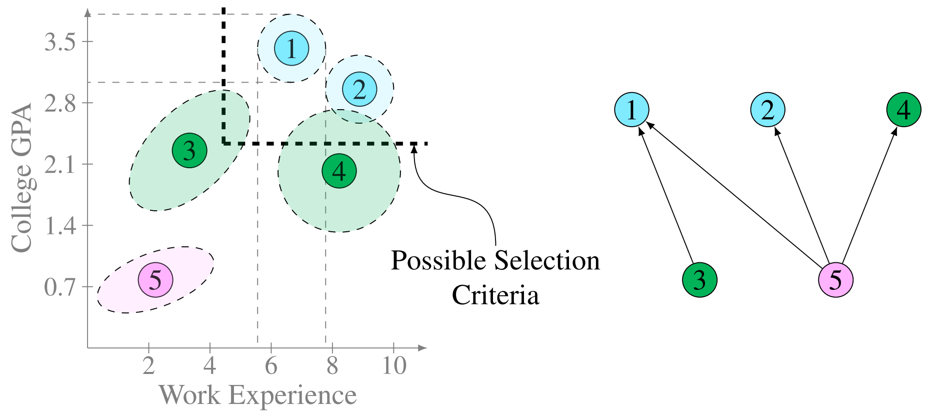


Figure 5.3: A depiction of the work experience and college GPA of five candidates coming from three groups (differentiated by color). The thick dashed line represents a possible selection criteria which, in this case, imposes a threshold on each of the two attributes. Confidence regions are drawn around each data point to indicate, say, 95% confidence in the inclusion of a candidate's true ability. Given these confidence regions, one can construct a partial ranking, as depicted by the Hasse diagram on the right. Arrows between candidates indicate ranking with certainty with respect to both attributes (e.g., Candidate 1 is ranked higher than Candidate 5 since the best work experience score (≈ 4) in the confidence region of Candidate 5 is worse than the worst work experience score (≈ 5.5) of Candidate 1, and similarly for College GPA. Note that there may be other reasonable ways of constructing partial rankings as well.

who are maximally ranked (i.e., are not ranked below any other candidates): the two blue candidates and the right-most green candidate. This is one justification for selecting these candidates.

The process outlined in these examples (forming score ranges/regions for each candidate and inferring comparisons therefrom) can be applied quite generally, and allows for explicit treatment of bias in data. Data-driven techniques, such as estimating latent group-bias in a machine learning model, can be applied to generate these score ranges, which in turn induce a partial ranking. Such methods can be used to avoid penalizing applicants who come from underrepresented groups, who are more likely to face inaccurate evaluation via machine learning models. A recent paper by Emelianov et al. shows that groups with high error variances can receive worse treatment, even if the evaluations are unbiased for all candidates [30], pointing to the need for interventions like the poset approach that take uncertainty into account.

The poset approach in practice. We end this section by providing a framework for using the poset approach in practice. While this framework does not encompass every possible use of the poset approach [150], it will describe the process from beginning to end and put the poset approach in broader context.

Step 1. Clean-up and Process Past Hiring Data. To start, collect data from previous hiring cycles. This data might include scores derived from textual analysis of résumés, test scores for job-related tasks (e.g., computer programming test scores), automated scores based on analysis of video interviews [54], college GPAs, courses taken, years of work experience, job performance of those who were hired, and so on.

Step 2. Quantify Uncertainty and Bias. Use data analysis to quantify potential data biases. Clusterings, for example, can help determine if evaluations unfairly favor one group over another. Looking at the data along different demographics (e.g., based on race, gender, age) can point to potentially discriminatory decisions in the past. Use social science studies (e.g., [56]) that highlight the impact of social status on the considered metrics (e.g., standardized test scores). This will help highlight qualitative and quantitative reasons for disparities in the past hiring data.

Step 3. Construct a Partial Order. Trends identified in Step 2 can be used to construct a partial ranking of candidates. For example, score ranges can be constructed for each attribute of interest using a prediction model and estimates of its error variances, and these ranges can inform partial rankings as in Figure 5.3. These ranges can take into account distributional differences across protected attributes, differing error variances due to training data imbalance,[5] observed inaccuracies in past predictions, and so on. When feedback is available on past data, one can construct a partial order to account for group-specific errors, even if the evaluation metric was provided by a third party and its inner workings

[5]This refers to the observation that a group which is underrepresented in training data often experiences large errors in a resulting prediction model. In the poset approach, these larger errors could translate to larger score ranges for the underrepresented group. Note that the groups in question could come from a clustering and need not be demographic groups.

are unknown to the user: the predicted scores can be modified to be distributionally similar to the true scores on a group-by-group basis, and score ranges can be constructed around these transformed scores. Unsupervised methods such as clustering can be used without the specific knowledge of protected information, and a partial order can be constructed based on the extent of uncertainty or bias in each cluster. More examples are discussed in Chapter 3. The goal here is to account for uncertainties, inaccuracies, and biases in a direct and mathematically justified way, thereby paving the way to fairer decisions.

Step 4. Adapt Selection Algorithms. Once the partial ranking has been constructed, selections need to be made. Presumably, a hiring committee already has a screening process (automated or otherwise) which aligns with the goals of the employer. In order to implement the poset approach, this screening process must be adapted to take a partial ranking as input instead of numeric scores or a total ranking. Typically, this can be done by prioritizing maximality and randomizing wherever incomparabilities necessitate (see Chapter 3 for an example of this in an online setting).

Step 5. Auditing for Policy Compliance. The entire hiring pipeline may be subject to auditing for compliance with anti-discrimination policy. It is prudent to document and be able to justify each decision made in the hiring process, particularly those pertaining to the four steps outlined above. For example, one should be able to explain how the partial ranking was constructed and be able to justify those decisions by pointing to data and relevant research. A deeper discussion of the legality of the poset approach (and algorithmic bias mitigation more generally) is in Section 5.1.6.

5.1.6 Discussion and Best Practices—Law, Mathematics, and Posets in Practice

We now return to the business cases with which we started and the tensions they present regarding diversity, equity, and legal interests. On the one hand, firms are seeking to address diversity regardless of a history of discrimination. On the other hand, when evidence of past or present practices creating barriers is found, companies addressing those practices

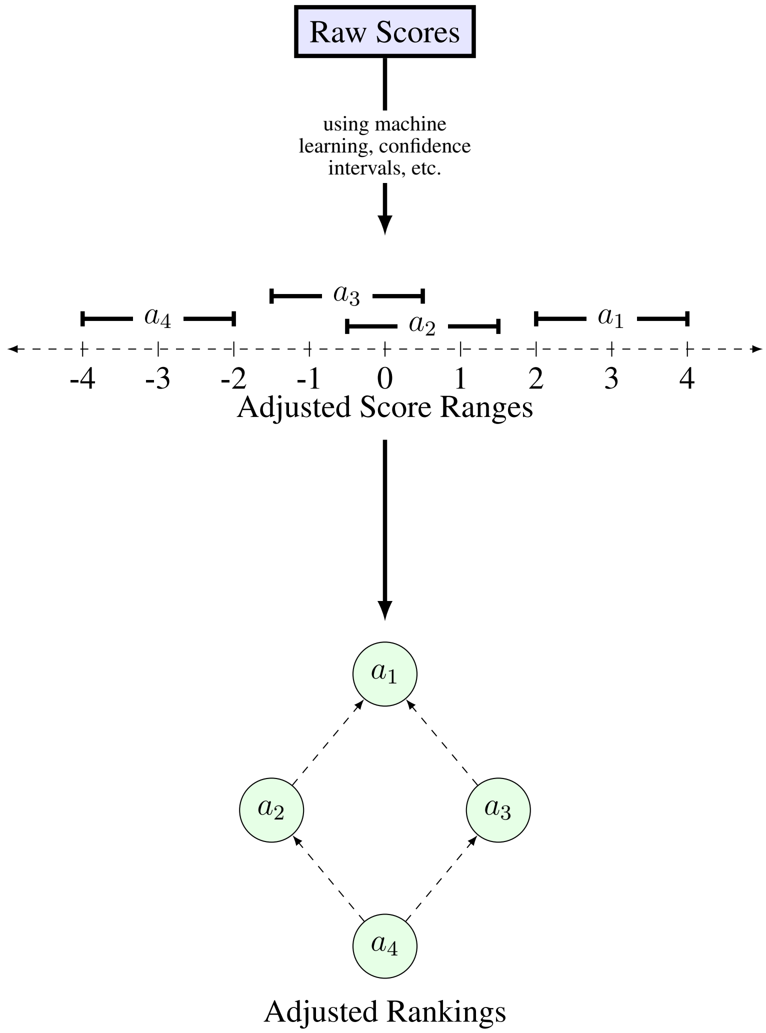


Figure 5.4: This figure outlines the process of correcting for bias according to the poset model. First, using each applicant's raw score (and any estimated bias or inaccuracy), adjusted score ranges are produced, and partial rankings can be derived from these ranges. The goal is for decisions made using these partial (adjusted) rankings to mitigate the bias present in raw scores.

are pursuing affirmative action plans. In general, a firm that does little to account for race, gender, and other protected classes may find it has created disparate impact; and yet, when that firm seeks to take protected classes into account, such steps may violate the ban on disparate treatment.

We offer that in the unlikely case where a firm has no reason to believe that norms, traditions, or societal inequalities are negatively affecting the ability for members of a protected group to pass through the stages of the hiring pipeline, action may be possible under diversity interests but not required by law. At least two shifts point to increased diversity activity. First, many companies have made public commitment to large steps to address diversity in employment. Second, there is a new push for companies to disclose workforce diversity data, which has resulted in 82 of the top 100 companies doing so. The public imperative combined with the data supports companies taking the initiative to address workforce imbalances regardless of legal requirements to do so [164]. In contrast, as matter of affirmative action, a firm with evidence of discrimination seeking to address imbalances in its workforce should be able to take steps to do so. Such steps could involve, for instance, scoring applicants using a machine learning model and developing confidence intervals around scores using the poset approach. From a legal perspective, it is important to be able to support the legality of each action, from the decision to address diversity to the decision to use protected class information, to each design choice in the algorithm, to each adjustment to future rounds of hiring.

The beauty of the poset approach is that it is agnostic to the motivation, diversity or addressing discrimination via affirmative action, behind a company's plan. To be clear, whether a purely diversity-driven plan is legal is an unsettled question and beyond the scope of this paper [109, 115]. Nonetheless, because of the current drive to address inequity, we expect this question to arise in the near future and suggest that the poset approach would aid and support such efforts. Furthermore, because many announced diversity programs are likely backed by data about imbalances and unnecessary barriers to employment, such efforts will likely be seen as affirmative action plans under the law. Thus in this section we address the core question of how well the poset approach stands up to legal scrutiny as an allowed method to address affirmative actions plans.

Given that efforts to modify evaluation mechanisms or selection algorithms can raise both disparate impact and disparate treatment issues, we now use a hypothetical employer perspective in line with Microsoft's and other companies' announced goals to suggest best practices. Insights are derived from a series of questions about how to identify workforce imbalances (Section 5.1.6) and how to address said imbalances (Section 5.1.6).

Diagnosis

Q1: *An employer is concerned that its workforce under-represents women and minorities. May they do anything to change their current hiring practices?*

Yes. The purpose behind Title VII is "[T]o achieve equality of employment opportunities," and Congress "directed the thrust of the Act to the consequences of employment practices, not simply the motivation" [165]. That means that "unnecessary barriers to employment" must fall, even if "neutral on their face" and "neutral in terms of intent" [166]. Federal courts have disallowed a host of hiring and promotion practices that "operate[d] as 'built in headwinds' for minority groups" [167]. In addition, the Supreme Court has upheld the legality of employment plans to address discrimination without reference to its past practices or evidence of a possible violation of the law [168].

To take action, an employer "need[s] to point only to a 'conspicuous ... imbalance in traditionally segregated job categories' " [168]. Logically, this requirement implies that initial, proactive analysis identifying the imbalance problems can serve as justification for adjustments to hiring practices. As such, employers can and should use data science and analytics to identify the imbalance in their hiring pipeline that it seeks to address [169, 170, 171].

As one example, the employer can use human resources data to examine its employment practices. First, it can audit its current workforce and get fine-grained information about who works at the company and at what levels. Such an approach allows the company to look beyond simple questions such as "Does it have an equal number of men and women in the workforce?" Instead, the company can see the gender and minority makeup at different levels of employment such as upper management, upper-middle management, middle management, administration, hourly workers, contractors, and so on. Visualizing the data with pie-charts or heat maps will provide clear, vivid ways to see the current state of affairs. Second, after such a study, the company can see potential sources of issues. It may find that women and minorities rarely move beyond middle management, are rarely interviewed for promotion, or that screening to date has not selected, or under-selected, women and minorities for interviews to be potential employees. At a general level, these types of analyses support the case that there is something to fix. This gets us to the next step in the process.

Q2: *If a company finds that women and minorities are rarely interviewed and further finds that screening to date has not selected, or under-selected, women and minorities for interviews to be potential employees, do these conditions support allowing an employer to use protected-class information to build or apply a bias-aware algorithm at the screening stage?*

Identifying a problem with a screening process or a structural problem in the company's workforce, reveals a clear "unnecessary barrier to employment" even if the algorithm is neutral on its face and in intent. For example, if men tend to be scored higher than women (e.g., as in Fig. 5.5), then a facially neutral selection algorithm would disproportionately select men, even if true ability is similar across genders. In general, the identified, strong evidence of bias in current algorithmic sorting in the hiring process, including the screening stage, should constitute the sort of "built in headwind[] for minority groups" that the law seeks to eliminate. With sufficient evidence of bias and systemic barriers to equality of employment opportunities, an employer can make a case for using bias-aware algorithms.

Corrective action

Voluntary action to comply with the goals of Title VII is not only allowed; it is favored [172]. Nonetheless, in some cases, trying to further the goals of Title VII to address discrimination raises the paradox where one approach looks like disparate impact and a corrective action looks like disparate treatment. What can a company actually do?

Q3: *May an employer use protected-class information to increase diversity among interviewees?*

This question is complex as it entwines various parts of the process that need to be slowly unpacked. A recent case *Ricci v. DeStefano* [173] illustrates some problems and provides guidance on allowed and prohibited actions.

Background. In *Ricci v. DeStefano*, the City of New Haven had developed a test for firefighter promotion with the help and validation of experts. When administered, 77 people took the lieutenant exam: "43 whites, 19 blacks, and 15 Hispanics. Of those, 34 candidates passed: 25 whites, 6 blacks, and 3 Hispanics." 41 people took the captain's exam: "25 whites, 8 blacks, and 8 Hispanics. Of those, 22 candidates passed: 16 whites, 3 blacks, and 3 Hispanics." Despite the experts' opinions and validations of the test, the City rejected the results because the pass rate caused the city to believe it might be sued for disparate impact. The Supreme Court did not allow this after-the-fact change, because New Haven's actions relied on race, (the race of those who passed the test), to reject the results, and in that sense, New Haven engaged in disparate treatment. Thus, it may appear that an entity cannot account for and alter employment practices when there is evidence of potential disparate impact in the entity's practices, because such changes will necessarily constitute disparate treatment [130]. That is incorrect [174].

Analysis. As the Supreme Court put it, not allowing an entity to account for race to avoid disparate impact liability "if the employer knows its practice violates the disparate-impact provision," is contrary to "Congress's intent that "voluntary compliance" be "the preferred means of achieving the objectives of Title VII" [175]. This rule, however, does not mean an entity can simply assert there has been a history of past discrimination and so a need to throw out a practice, because that might lead to "an unyielding racial quota" [176]. As stated above, the entity has to show why the change is needed in light of the goals of Title VII. In addition, the timing of when an entity makes changes matters.

The way the test was developed and administered by New Haven doomed the City's decision to reject the test's outcomes. New Haven began well by hiring experts to design a likely *valid test*. The City spent $100,000 on experts on designing the tests for fire departments [177]. The experts conducted interviews, went on ride-alongs, interviewed incumbents at the promotional level at issue, and designed "job-analysis questionnaires and administered them to most of the incumbent battalion chiefs, captains, and lieutenants in the Department" [177]. As the Supreme Court noted, "At every stage of the job analyses, IOS [the company that developed the test], by deliberate choice, oversampled minority firefighters to ensure that the results—which IOS would use to develop the examinations—would not unintentionally favor white candidates" [178]. Once the test was approved, New Haven set a 3-month study period and gave candidates a study guide including the "source material for the questions, including the specific chapters from which the questions were taken" [178]. Nonetheless, after the tests were given, the results indicated disparate impact [179].

The city's ex-post actions were the problem. The Court rejected "invalidating the test results" after the fact without "a strong basis in evidence of an impermissible disparate impact" [180]. The ex-post rejection of the results created "visible victims"—that is, those who studied for the test, passed, and whose hard work was discarded [181]. After the city gave the test, it needed strong evidence that the test would be invalidated if the city were sued for disparate impact and lose, because otherwise those who had passed would be harmed. The Court did not see such evidence and so did not allow the city to reject the results.

Answer to Q3. Designing a screening system is quite different than what happened in *Ricci*. *Ricci* was about a later stage of employment (i.e., promotions), and it involved a test for which many test-takers had prepared, including spending money on test preparation aid. The advantage of building a screening system is that the actions are ex-ante, and the system is not a test for which someone can prepare [115]. Unlike in *Ricci*, where applicants were seen as having an expectation that a potentially valid test for which they could study be accepted, designing and using a screening algorithm occurs at an earlier stage of the hiring process where no hiring or promotion decision is made. Thus in designing a screening algorithm, one might observe selections over time and change the parameters to create a more representative sample of qualified candidates, including making adjustments during the "training" of the algorithm. These steps are analogous to the design steps—such as making overt choices and oversampling at every stage to ensure that the test did "not unintentionally favor white candidates"—taken by New Haven and of which the Supreme Court wrote with approval [178]. In other words, designing and vetting a screening system to ensure that the results are not having discriminatory outcomes should be legal.

Recall that one of the goals of Title VII is to reduce, if not eliminate, "unnecessary barriers to employment." The *Ricci* Court did not "question an employer's affirmative efforts to ensure that all groups have a fair opportunity" at a given stage of the hiring process. An employer is allowed to examine "how to design…[a] practice in order to provide a fair opportunity for all individuals, regardless of their race" before deploying it [180]. Designing a screening algorithm is by its nature an ex-ante event for which a candidate cannot prepare in the way one might for a test.

In short, if Question 2's requirement is met, an employer should be able to develop a bias-aware algorithm to avoid disparate impact. Of course, we still need to address the validity of the new practice and what is allowed in its design, which brings us to the next question, which we partially answer through the lens of the poset approach.

Q4: *What is allowed in the design of a bias-aware algorithm? Can it be designed to improve the yield of whom to interview?*

This is one of the grand challenges in this area. Let us focus our attention to the proposed poset approach, and draw arguments from the Supreme Court's decision in *Johnson*. The key to using a bias-aware algorithm such as the poset approach of Salem and Gupta is to establish the facts and evidence of a need to address bias (or more generally, inconsistencies in the data) as set forth above, and then to build a plan that assesses individuals rather than setting up a purely number-driven process with quotas for each category [172]. If a plan is "blind hiring," that is, dictates hiring "solely by reference to statistics" or "by reflexive adherence to a numerical standard," the plan is not likely to be allowed [182]. But, if a plan takes "numerous factors. . .into account in making hiring decisions, including specifically the qualifications of [all] applicants for particular jobs," the plan may take a protected class into account as part of the overall evaluation [172]. In that sense, the protected class status "may be deemed a 'plus' in a particular applicant's file, yet it does not insulate the individual from comparison with all other candidates for the available seats" [183, 184].

Comparison does not require pure, numeric ranking; indeed, that might tip into the sort of "blind hiring" that is disfavored. As the Sixth Circuit stated, the "practice of rank-order hiring from a single list grouping together males and females was impermissible under Title VII because the City could not establish that higher scores on the test meant better job performance." [185]. The Second Circuit has explained that evaluations should be sufficiently correlated with job performance to induce a rank ordering, where the quantification of "sufficiently correlated" may depend on the extent of adverse impact of the evaluation metric [186]. The Sixth Circuit additionally asserted that a certain cognitive ability test could not be used as the sole basis for a rank-ordering despite being predictive of job performance, since the test failed to measure certain qualities of interest. Rank orderings based on evaluations should therefore not be thought of as implicit to a screening practice, but instead as a design choice which must be justified [185].

Discretion in comparison of candidates is allowed when it is part of the overall, individual assessment. For example, in *Johnson v. Transportation Agency of Santa Clara County*, two candidates were deemed well-qualified based on a range of metrics, such as experience, background, and test scores taken together. But each candidate had differences within a given metric. One had more clerical work and more road maintenance work; the other had more experience at a specific part of the business. As for test scores, the man scored 75 on the interview portion of the assessment and the woman scored 73. The employer had set 70 as the minimum threshold for the interview and seven applicants crossed the 70 mark. The range of acceptable scores was 70 to 80 [187]. The woman was given the promotion over the man who had the higher score. Because the scores were within the range of acceptable scores and the final hiring manager looked at a set of metrics with gender as "but one of numerous factors he took into account in arriving at his decision," the plan's incorporation of bias-awareness, here gender, was allowed [183].

Others cases also acknowledge the need for an approach beyond using an absolute score or ranking. Given problems with rank-ordering, the Second Circuit of Appeals has allowed a rather coarse approach where an employer may "acknowledge his inability to justify rank-ordering and resort to random selection from within either the entire group that achieves a properly determined passing score, or some segment of the passing group shown to be appropriate" [188]. Courts have also indicated an acceptance for more nuanced methods. For example, the act of "banding," or considering score ranges instead of singular scores, has been accepted to account for inaccuracies in evaluation. [189, 190]. Although these cases consider banding in a quite limited sense in that scores ranges are centered on original scores and are of uniform length, they support that one might relax the assumption of an absolute ranking of candidates.

In language that tracks the poset approach, the Second Circuit has also acknowledged "that small differences between the scores of candidates indicate very little about the candidates' relative merit and fitness" [188]. Thus the court embraced an approach that assessed "a statistical computation of the likely error of measurement inherent" in its exam. The employer then used that measurement to set up zones of candidates clustered by test scores within that error measurement. That practice was seen as a good solution to "insur[e] compliance" with Title VII. The Second Circuit explained, "by creating a more valid method to assess the significance of test scores, [the approach] eliminated the central cause of the adverse impact, i.e., the rank-ordering system, while assuring appointments on the basis of merit." As such, if one is able to use protected information (as in *Johnson*, or in the context of a valid affirmative action plan [115]), then the banding cases provide guideposts for adopting the poset approach as described in Section 5.1.5.

Answer to Q4. 1. An algorithmic approach should be allowed. A takeaway from *Johnson* and the cases on banding and rank-ordering is that a precise numerical score is not necessarily indicative of an applicant's potential, and courts welcome approaches that better compare candidates. Thus, score ranges can be used as part of an applicant-screening procedure. This supports the use of score ranges to account for uncertainties in evaluations, as outlined in Section 3.

Further, note that incorporating the poset model of bias is not the same thing as normalizing distributions of scores across groups. When we normalize scores across groups, we are essentially transforming all scores so that group-specific distributions look similar, and this process results in a full ranking of applicants. In contrast, the poset approach intentionally does not reduce each applicant to a number and allows for incomparabilities between applicants. This allows for a more individual treatment of candidates, where uncertainty in rankings can be acknowledged. The result is that applicants are assessed as individuals,

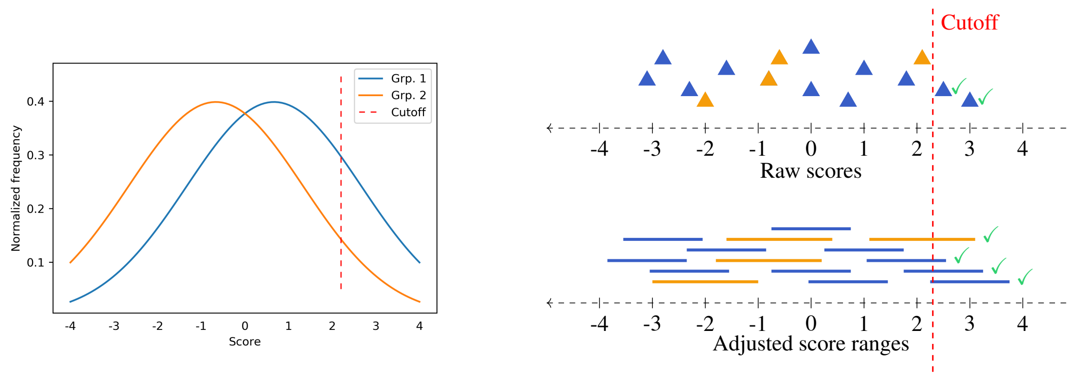


Figure 5.5: (left) Example of score distributions (blue: Group 1, orange: Group 2) and (right) potential score ranges for candidates from these distributions. Suppose a hiring committee wants to select two of the applicants represented in the right plot. If only the raw evaluations (the centers of the intervals) are used to make these decisions, then only the two high-scoring Group 1 candidates could be selected, as they are the only applicants meeting the cutoff. However, if score ranges are considered, then the highest-scoring Group 2 candidate meets the cutoff as well. In this example, adopting the poset method results in a more diverse slate of candidates meeting the cutoff, vis-à-vis using raw scores.

potentially in a more mathematically sound way.

2. There are rules about when bias-aware algorithms can be used. Recall that the stage at which an entity uses bias-aware algorithms matters. In the promotion context of *Johnson*, the Court gave a further reason the plan was allowed. Unlike *Ricci*, where applicants were seen as having an expectation that a potentially valid test for which they could study be accepted, there was "no absolute entitlement" to the position at issue in *Johnson*. The entity had seven qualified and eligible applicants, and choosing one over the other "unsettled no legitimate, firmly rooted expectation" of any of the candidates. By extension, a bias-aware applicant-screening plan that used a protected class as part of an overall assessment then had all selected applicants compete on the same metrics should be allowed under the law.

3. There are legal rules on the goals of any hiring plan. The law respects plans that seek to remedy an imbalance and that do not set aside positions for a given group while also conducting annual reviews of goals as it fashions future rounds of hiring and promotion [191]. One may work "to attain a balanced work force, not to maintain one" [192].

The *Johnson* Court also noted with approval that "the Plan sought annually to develop even more refined measures of the under-representation in each job category that required

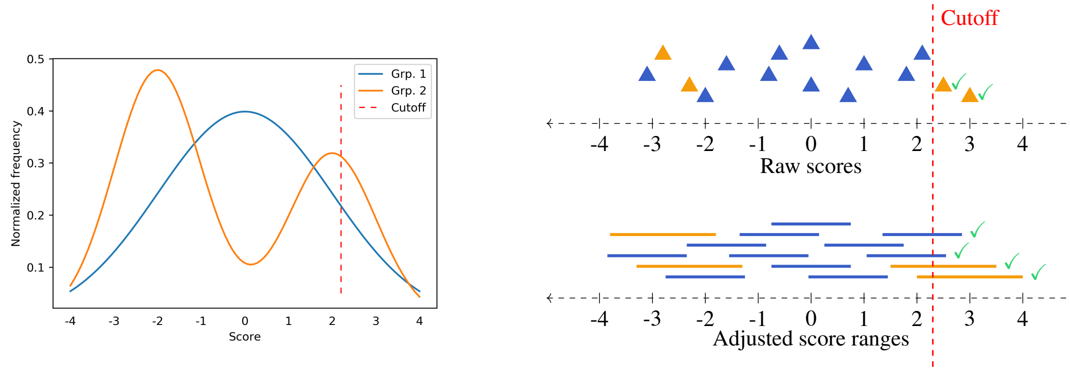


Figure 5.6: (left) Example of (somewhat unusual) score distributions (blue: Group 1, orange: Group 2) and (right) potential score ranges for candidates from these distributions. Suppose a hiring committee wants to select two of the applicants corresponding to the right plot. If only the raw evaluations (the centers of the intervals) are used to make these decisions, then only the two high-scoring Group 2 candidates could be selected, as they are the only applicants meeting the cutoff. However, if score ranges are considered, then the two highest-scoring Group 1 candidates meet the cutoff as well. From this example, we see that adopting the poset approach can be beneficial to the majority group as well and does not routinely advantage the lower-mean group (in this case, Group 2).

attention" [193]. This idea of not maintaining a balanced workforce reflects the idea that an entity cannot use a plan that sets up quotas to maintain balance based purely on class statuses. By extension, suppose balance is achieved in a company through bias-aware methods, and they notice this by continuous monitoring of their hiring practices (in a sense, returning to Question 1). The company may then have to stop using bias-aware methods, even if demographic imbalance persists in the general workforce for that line of work.

4. The poset approach does not impose quotas. In contrast to methods described in some recent work [30], using score ranges (e.g., using the poset approach) instead of raw scores does not set up a quota system.[6] When using the poset approach, selection rates may be influenced by protected information (e.g., when accounting for observed, group-specific biases), but such protected information is not necessarily a determining factor in selection decisions. For example, the poset approach could result in a set of candidates which is

[6]One can set aside seats for interviews as happens with the Rooney Rule in the NFL, but such a rule is best-protected by following the legal constraints for affirmative action plans. See e.g., https://www.aclusocal.org/en/inclusion-targets-whats-legal. Further, quota-based approaches at screening stages may create a pool of candidates destined for later rejection depending on the downstream decision process. The poset approach enables selections based on the possibility of a candidate being qualified and so better fits legal requirements at any stage of hiring.

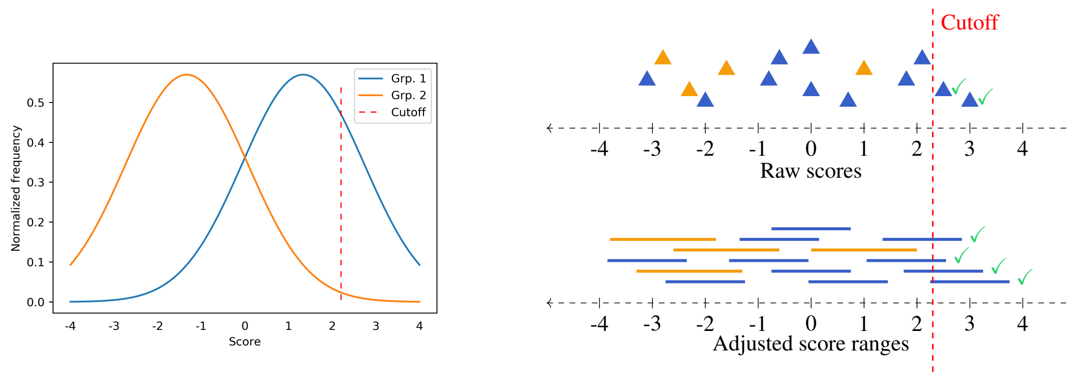


Figure 5.7: (left) Example of score distributions (blue: Group 1, orange: Group 2) and (right) potential score ranges for candidates from these distributions. Suppose a hiring committee wants to select two of the applicants corresponding to the right plot. If only the raw evaluations (the centers of the intervals) are used to make these decisions, then only the two highest-scoring Group 1 candidates could be selected, as they are the only applicants meeting the cutoff. If the score ranges are considered, then the four highest-scoring Group 1 candidates meet the cutoff. This example shows that adopting the poset approach does not necessarily increase the selection rate for the group with the lower mean score, and that the poset approach does not necessarily constitute a quota system.

less demographically proportional than what raw scores might produce (see, e.g., Figures 5.7-5.8), or more demographically proportional (see, e.g., Figures 5.5-5.6), depending on the data and the ascertained uncertainty therein.

5.1.7 Conclusion

We summarized recent work in the context of hiring, with a focus on screening algorithms. We highlighted the seeming paradox of mathematics, law and practice that a company might observe workforce imbalance due to its past practices, but the solutions to correct for this imbalance are either at a contradiction with mathematics or anti-discrimination law. The new poset-based approach [150] provides a framework for incorporating uncertainties in rankings into a candidate-screening practice which allows, for example, hiring committees to base decision on confidence intervals of ability scores. This approach can potentially be legally justified based on past disparate impact and can be adjusted over time as the data grows and hiring goals evolve; and thus can help avoid having a static plan as the law requires.

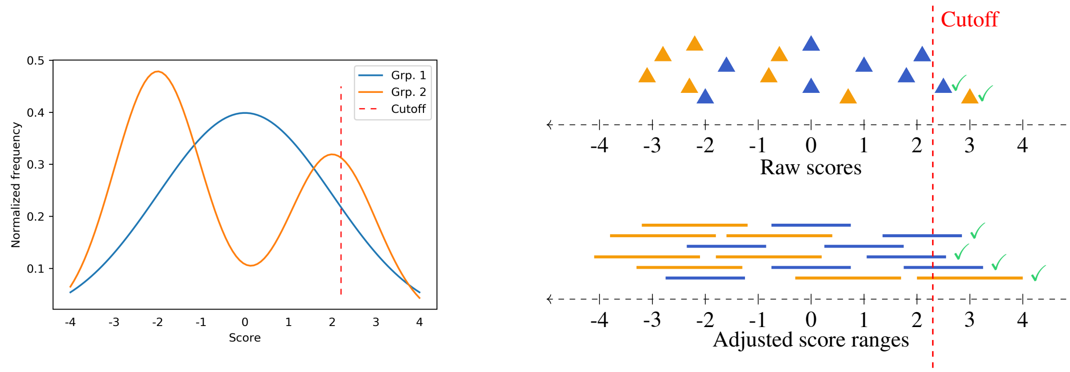


Figure 5.8: (left) Example of (somewhat unusual) score distributions (blue: Group 1, orange: Group 2) and (right) potential score ranges for candidates from these distributions. Suppose a hiring committee wants to select two of the applicants corresponding to the right plot. If only the raw evaluations (the centers of the intervals) are used to make these decisions, then one Group 1 and one Group 2 candidate will be selected, as they are the only applicants meeting the cutoff. In this case, demographic parity is achieved, as both groups have equal size. However, if score ranges are considered, then two additional Group 1 candidates meet the cutoff as well. This example shows that adopting the poset approach does not necessarily make the new candidate slate (i.e., those meeting the cutoff) more representative compared to using raw scores—indeed, in this example, adopting the poset approach moves the new candidate slate farther away from demographic parity.

No approach, however, is a fix-all solution. The poset approach cannot discount for undetectable errors undetectable, or modeling errors due to missing data. The ranges of the intervals impact the quality of selections. Further, two different mathematical approaches could be used to define score ranges for candidates and result in different sets of selected candidates. A legal dispute may require addressing which one of these approaches is more valid. Further, there is an "are we there yet?" issue built into the Supreme Court's rulings. That is, it may be unclear at which point a workforce becomes "balanced" and the current plan must be replaced. Although the poset approach is adaptive, detecting where there is no longer any impact of societal biases in the data is non-trivial and we leave this as an open question.

For any intervention in an existing framework, one has to consider if the intervention is serving those for whom it is designed [194]. Partially ordered sets that are interval-based might create an impression that certain underrepresented minorities carry high uncertainty in their potentials and as a result, lead a risk-averse hiring committee to reject those candidates. On the contrary, the poset approach is able to highlight *missed opportunities* in representation in the hiring pipeline. Taking uncertainties into account can expand and improve the talent pool to include candidates who are qualified and would have been competitive had there been no bias in the data. Thus, we believe that the analysis presented here can pave the way forward for hiring qualified candidates in a fair way in the evolving legal landscape.

5.2 Pricing and Privacy

In this section, I raise some open questions relating to pricing, law, and privacy. In general, as laws regarding data use and and automated decision-making emerge, questions about the legality of existing methods will as well.

First, consider *price gouging*: the act of dramatically increasing a pricing during an emergency. Amazon was recently sued for allegedly price-gouging during the pandemic, as they increased the price of essential goods by more than 450% (e.g., see Figure 5.9) compared to previously seen prices (*McQueen and Ballinger v. Amazon.com*[7]). The legal definition of price gouging varies across jurisdictions; e.g., bill H.R.7736 introduced to the House in 2022 defined price gouging with respect to prices set over the prior 120 days. Observe that price gouging is a temporally defined phenomenon and can potentially be prevented using a temporal constraint. This prompts the following questions:

1. How can price gouging laws be encoded as constraints?
2. Let p_t be the price set for a good on day t, and consider the constraint

   $$p_t \leq \frac{c}{m} \sum_{s=1}^{m} p_{t-s} \quad \text{for all } t > m,$$

   which may prevent some forms of price gouging. What are upper and lower bounds on regret for bandit convex optimization subject to these constraints?

[7]McQueen and Ballinger v. Amazon.com, Inc., 422 U.S. `Case 4:20-cv-02782` (2020).

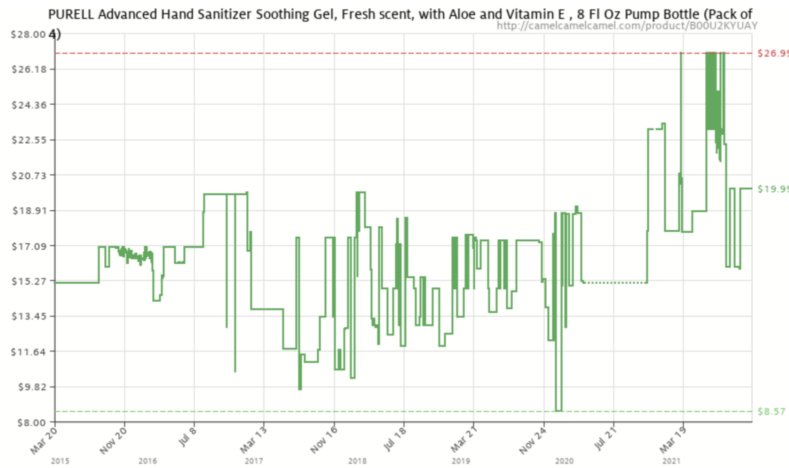


Figure 5.9: The price path of Purell hand sanitizer on Amazon from March 2015 to March 2021 (from https://camelcamelcamel.com/product/B00U2KYUAY)

Next, I am interested in the relationship between data privacy laws and algorithmic decision-making. Data privacy laws, such as the California Consumer Privacy Act (CCPA) [8] in California and the General Data Protection Regulation (GDPR) [9] in the E.U. restrict the use of data in decision-making. These laws can limit discrimination (as in the CCPA) or limit the use of automated decision-making (as in the GDPR) and potentially affect personalized pricing. This leads to the following question:

1. How do existing data privacy laws limit the implementation of dynamic, personalized pricing algorithms?

2. What ramifications does the "right to opt out" have on machine learning? If a certain group opts out of data collection moreso than another group, will the prediction accuracy on that group suffer?

CHAPTER 6
CONCLUSION

In this dissertation, I discussed models of fairness in online decision-making and relevant algorithm design. I showed that offline notions of fairness can sometimes be satisfied with full memory in an online setting (Chapter 3). In other cases, this results in poor performance (Section 4.1). In such cases, there is a range of options for extending the offline notion of fairness to an online setting, and both performance and impact might be considerations in deciding which to take.

Apart from the more specific open questions mentioned throughout this dissertation, there are some broad directions that merit study. First, it would be interesting to quantify the trade-offs between *memory* of a fairness constraint and *learnability*, in various online learning settings and notions of fairness. Having these trade-offs quantified could give decision-makers useful information in deciding how to constrain their decisions.

Second, it would be interesting to see more work done on long-term effects of constrained decision-making. While there has been some work done in this area (e.g., studying the long-term effects of affirmative action [72, 73]), much of this work focuses on demographic parity. It would be interesting to know the effects of other constraints, such as conditional demographic parity, on population qualifications and behaviors over time. The question of how decisions affect the relative qualification distribution across groups over time is complex and dependent on the policy used.

Third, I would like to highlight a broad legal question which I believe has far-reaching implications on algorithmic decision-making: *what exactly constitutes disparate treatment?* In Chapter 5, I mentioned disparate treatment as a form of illegal discrimination which involves the intentional use of protected information to harm a protected group. This definition, however, is quite fuzzy. If an employer makes hiring decisions using a feature which is correlated with a protected attribute, does this constitute disparate treatment? If an employer uses an ensemble approach, running several race-blind algorithms and choosing the one which produces the most racially-balanced output, has the employer applied disparate treatment? Outlining the limits of disparate treatment, to whatever extent is possible, will help guide algorithmic decision-making going forward.

With the rise of online decision-making in important social contexts, the question of impact is central. Deciding on fairness goals can be a difficult process, as there may be conflicting considerations based on public perceptions, law, ethics, and utility. That said, knowing which fairness goals can be achieved and understanding their impact allows for more meaningful consideration.

REFERENCES

[1] B. Friedman and H. Nissenbaum, “Bias in computer systems,” *ACM Transactions on Information Systems (TOIS)*, vol. 14, no. 3, pp. 330–347, 1996.

[2] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, “A survey on bias and fairness in machine learning,” *ACM Computing Surveys (CSUR)*, vol. 54, no. 6, pp. 1–35, 2021.

[3] K. V. Deshpande, S. Pan, and J. R. Foulds, “Mitigating demographic bias in ai-based resume filtering,” in *Adjunct publication of the 28th ACM conference on user modeling, adaptation and personalization*, 2020, pp. 268–275.

[4] R. Kleinberg, “A multiple-choice secretary algorithm with applications to online auctions,” in *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA ’05, Vancouver, British Columbia: Society for Industrial and Applied Mathematics, 2005, pp. 630–631, ISBN: 0-89871-585-7.

[5] AspiringMinds, *Aspiring Minds: AI powered talent evaluation*, Windows Phone Central, Ed., 22.05.2020.

[6] S. Gupta and V. Kamble, “Individual fairness in hindsight.,” *J. Mach. Learn. Res.*, vol. 22, no. 144, pp. 1–35, 2021.

[7] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. S. Zemel, “Fairness through awareness,” in *ITCS ’12*, 2011.

[8] CCPA, “TITLE 1.81.5. California Consumer Privacy Act of 2018 (Title 1.81.5 added by Stats. 2018, Ch. 55, Sec. 3.),”

[9] GDPR, “Regulation (EU) 2016/679 (General Data Protection Regulation),”

[10] S. Gupta, V. Kamble, and J. Salem, “Ai and ethics,” in Springer, TBD, ch. Temporal Fairness in Online Decision-Making.

[11] J. Dastin, “Amazon scraps secret ai recruiting tool that showed bias against women,” *Reuters*, Oct. 2018.

[12] S. Bansal, A. Srivastava, and A. Arora, “Topic modeling driven content based jobs recommendation engine for recruitment industry,” *Procedia computer science*, vol. 122, pp. 865–872, 2017.

[13] W. Baker, D. Kiewell, and G. Winkler, “Using big data to make better pricing decisions,” *McKinsey Analysis*, 2014.

[14] J. Jagtiani and C. Lemieux, "The roles of alternative data and machine learning in fintech lending: Evidence from the lendingclub consumer platform," *Financial Management*, vol. 48, no. 4, pp. 1009–1029, 2019.

[15] C. Rigano, "Using artificial intelligence to address criminal justice needs," *National Institute of Justice Journal*, vol. 280, pp. 1–10, 2019.

[16] A. Panesar, *Machine learning and AI for healthcare*. Springer, 2019.

[17] S. Barocas, M. Hardt, and A. Narayanan, *Fairness and Machine Learning: Limitations and Opportunities*. fairmlbook.org, 2019, http://www.fairmlbook.org.

[18] A. Chouldechova and A. Roth, "The frontiers of fairness in machine learning," *arXiv preprint arXiv:1810.08810*, 2018.

[19] J. W. Fuchs and Q. R. Youmans, "Mitigating bias in the era of virtual residency and fellowship interviews," *Journal of Graduate Medical Education*, vol. 12, no. 6, pp. 674–677, 2020.

[20] M. P. Kim, A. Ghorbani, and J. Zou, "Multiaccuracy: Black-box post-processing for fairness in classification," in *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, 2019, pp. 247–254.

[21] S. Ramraj, V. Sivakumar, *et al.*, "Real-time resume classification system using linkedin profile descriptions," in *2020 International Conference on Computational Intelligence for Smart Power System and Sustainable Energy (CISPSSE)*, IEEE, 2020, pp. 1–4.

[22] A. V. Den Boer, "Dynamic pricing and learning: Historical origins, current research, and new directions," *Surveys in operations research and management science*, vol. 20, no. 1, pp. 1–18, 2015.

[23] T. Calders, F. Kamiran, and M. Pechenizkiy, "Building classifiers with independency constraints," in *2009 IEEE international conference on data mining workshops*, IEEE, 2009, pp. 13–18.

[24] M. Joseph, M. Kearns, J. H. Morgenstern, and A. Roth, "Fairness in learning: Classic and contextual bandits," *Advances in neural information processing systems*, vol. 29, 2016.

[25] J. Sánchez-Monedero, L. Dencik, and L. Edwards, "What does it mean to "solve" the problem of discrimination in hiring? social, technical and legal perspectives from the uk on automated hiring systems," in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, ser. FAT* '20, Barcelona, Spain: Association for Computing Machinery, 2020, pp. 458–468, ISBN: 9781450369367.

[26] K. L. Haws and W. O. Bearden, "Dynamic pricing and consumer fairness perceptions," *Journal of Consumer Research*, vol. 33, no. 3, pp. 304–311, 2006.

[27] N. A. Saxena, K. Huang, E. DeFilippis, G. Radanovic, D. C. Parkes, and Y. Liu, "How do fairness definitions fare? testing public attitudes towards three algorithmic definitions of fairness in loan allocations," *Artificial Intelligence*, vol. 283, p. 103 238, 2020.

[28] D. Bertsimas, V. F. Farias, and N. Trichakis, "Fairness, efficiency, and flexibility in organ allocation for kidney transplantation," *Operations Research*, vol. 61, no. 1, pp. 73–87, 2013.

[29] M. Zehlike, F. Bonchi, C. Castillo, S. Hajian, M. Megahed, and R. Baeza-Yates, "Fa* ir: A fair top-k ranking algorithm," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 1569–1578.

[30] V. Emelianov, N. Gast, K. P. Gummadi, and P. Loiseau, "On fair selection in the presence of implicit variance," in *Proceedings of the 2020 ACM Conference on Economics and Computation*, 2020.

[31] S. Corbett-Davies, E. Pierson, A. Feller, S. Goel, and A. Huq, "Algorithmic decision making and the cost of fairness," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2017, pp. 797–806.

[32] A. Chouldechova, "Fair prediction with disparate impact: A study of bias in recidivism prediction instruments," *Big data*, vol. 5, no. 2, pp. 153–163, 2017.

[33] J. Kleinberg, "Inherent trade-offs in algorithmic fairness," in *Abstracts of the 2018 ACM International Conference on Measurement and Modeling of Computer Systems*, 2018, pp. 40–40.

[34] G. Pleiss, M. Raghavan, F. Wu, J. Kleinberg, and K. Q. Weinberger, "On fairness and calibration," in *Advances in Neural Information Processing Systems*, 2017, pp. 5680–5689.

[35] J. Kleinberg, S. Mullainathan, and M. Raghavan, "Inherent trade-offs in the fair determination of risk scores," in *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[36] H. Heidari and A. Krause, "Preventing disparate treatment in sequential decision making," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, pp. 2248–2254.

[37] Y. Bechavod, C. Jung, and S. Z. Wu, "Metric-free individual fairness in online learning," *Advances in neural information processing systems*, vol. 33, pp. 11 214–11 225, 2020.

[38] E. B. Dynkin, *The optimum choice of the instant for stopping markov process*, Soviet Mathmatics, Doklady 4, 1963.

[39] A. Borodin and R. El-Yaniv, *Online Computation and Competitive Analysis*. USA: Cambridge Univ. Press, 1998, ISBN: 0521563925.

[40] S. Albers and L. Ladewig, "New results for the k-secretary problem," *Theoretical Computer Science*, vol. 863, pp. 102–119, 2021.

[41] E. Hazan and K. Y. Levy, "Bandit convex optimization: Towards tight bounds.," in *NIPS*, 2014, pp. 784–792.

[42] A. Daniely and Y. Mansour, "Competitive ratio versus regret minimization: Achieving the best of both worlds," *arXiv preprint arXiv:1904.03602*, 2019.

[43] R. Motwani and P. Raghavan, *Randomized algorithms*. Cambridge university press, 1995.

[44] A. Rakhlin, K. Sridharan, and A. Tewari, "Online learning: Random averages, combinatorial parameters, and learnability," *Advances in Neural Information Processing Systems*, vol. 23, 2010.

[45] M. Feldman, O. Svensson, and R. Zenklusen, "A simple O(log log(rank))-competitive algorithm for the matroid secretary problem," in *Proc. of the Twenty-sixth Annual ACM-SIAM Symp. on Discrete Algorithms*, ser. SODA '15, San Diego, California: Society for Industrial and Applied Mathematics, 2015, pp. 1189–1201.

[46] J. Soto, "Matroid secretary problem in the random-assignment model," *SIAM Journal on Computing*, vol. 42, no. 1, pp. 178–211, 2013.

[47] N. Buchbinder, K. Jain, and M. Singh, "Secretary problems via linear programming," *Mathematics of Operations Research*, vol. 39, no. 1, pp. 190–206, 2014.

[48] E. Hazan *et al.*, "Introduction to online convex optimization," *Foundations and Trends® in Optimization*, vol. 2, no. 3-4, pp. 157–325, 2016.

[49] R. Vershynin, *High-dimensional probability: An introduction with applications in data science*. Cambridge university press, 2018, vol. 47.

[50] H. M. MacNeille, "Partially ordered sets," *Transactions of the American Mathematical Society*, vol. 42, no. 3, pp. 416–460, 1937.

[51] B. Schroeder, *Ordered Sets*, 1st. Birkhäuser Basel, 2003, ISBN: 978-1-4612-0053-6.

[52] P. Rudnicki, "Dilworth's decomposition theorem for posets," *Formalized Mathematics*, vol. 17, pp. 223–232, Jan. 2009.

[53] J. Salem and S. Gupta, "Secretary problems with biased evaluations using partial ordinal information," *Available at SSRN 3444283*, 2019.

[54] M. Raghavan, S. Barocas, J. Kleinberg, and K. Levy, "Mitigating bias in algorithmic hiring: Evaluating claims and practices," in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 2020, pp. 469–481.

[55] S. Spencer, C. M. Steele, and D. M.Quinn, "Stereotype threat and women's math performance," *Journal of Experimental Social Psychology*, vol. 35, pp. 4–28, 1 Jan. 1999.

[56] C. M. Steele and J. Aronson, "Stereotype threat and the intellectual test performance of african americans," *Journal of Personality and Social Psychology*, vol. 69, pp. 797–811, 5 1995.

[57] M. J. Lovaglia, J. W. Lucas, and S. R. Thye, "Status processes and mental ability test scores," *American Journal of Sociology*, vol. 104, no. 1, AJSv104p195–228, 1998.

[58] C. Moss-Racusin, V. Brescoll, M. Graham, and J. Handelsman, "Science faculty's subtle gender biases favor male students," *Proceedings of the National Academy of Sciences*, vol. 109, pp. 16 474–16 479, Sep. 2012.

[59] M. De-Arteaga *et al.*, "Bias in bios: A case study of semantic representation bias in a high-stakes setting," in *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 2019, pp. 120–128.

[60] R. Goodman, *Why amazon's automated hiring tool discriminated against women*, Published Oct. 12, 2018. Last accessed Jun. 6, 2019., 2018.

[61] M. Cavicchia, *How to fight implicit bias? with conscious thought, diversity expert tells nabe. american bar association: Bar leader*, 2015.

[62] E. Pierson *et al.*, "A large-scale analysis of racial disparities in police stops across the united states," *The Stanford Open Policing Project*, 2019.

[63] K. Lum, "Predictive policing reinforces police bias," *Human Rights Data Analysis Group*, 2016.

[64] J. Larson, S. Mattu, L. Kirchner, and J. Angwin, "How we analyzed the compas recidivism algorithm," *Propublica*, 2016.

[65] M. Babaioff, N. Immorlica, D. Kempe, and R. Kleinberg, "A knapsack secretary problem with applications," in *Approximation, randomization, and combinatorial optimization. Algorithms and techniques*, Springer, 2007, pp. 16–28.

[66] M. Babaioff, M. Dinitz, A. Gupta, N. Immorlica, and K. Talwar, "Secretary problems: Weights and discounts," in *Proceedings of the twentieth annual ACM-SIAM Symposium on Discrete Algorithms*, SIAM, 2009, pp. 1245–1254.

[67] J. A. Soto, A. Turkieltaub, and V. Verdugo, "Strong algorithms for the ordinal matroid secretary problem," in *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, SIAM, 2018, pp. 715–734.

[68] R. Kumar, S. Lattanzi, S. Vassilvitskii, and A. Vattani, "Hiring a secretary from a poset," Jan. 2011, pp. 39–48.

[69] J. M. Kleinberg and M. Raghavan, "Selection problems in the presence of implicit bias," in *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, 2018, 33:1–33:17.

[70] L. E. Celis, A. Mehrotra, and N. K. Vishnoi, "Interventions for ranking in the presence of implicit bias," in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 2020, pp. 369–380.

[71] V. Emelianov, N. Gast, K. P. Gummadi, and P. Loiseau, "On fair selection in the presence of implicit variance," in *Proceedings of the 2020 ACM Conference on Economics and Computation (EC 2020)*, 2020.

[72] H. Mouzannar, M. I. Ohannessian, and N. Srebro, "From fair decision making to social equality," in *Proceedings of the Conference on Fairness, Accountability, and Transparency*, ser. FAT* '19, Atlanta, GA, USA: ACM, 2019, pp. 359–368, ISBN: 9781450361255.

[73] S. Coate and G. Loury, "Will affirmative-action policies eliminate negative stereotypes?" *American Economic Review*, vol. 83, pp. 1220–40, Feb. 1993.

[74] S. O. Gharan and J. Vondrák, "On variants of the matroid secretary problem," *CoRR*, vol. abs/1104.4081, 2011.

[75] H. Kierstead and W. Trotter, "An extremal problem in recursive combinatorics," *Proceedings of the Twelfth Southeastern Conference on Combinatorics, Graph Theory and Computing*, vol. Vol. II, 1981.

[76] S. Greenberg and M. Mohri, "Tight lower bound on the probability of a binomial exceeding its expectation," *Statistics & Probability Letters*, vol. 86, pp. 91–98, 2014.

[77] M. Raghavan, S. Barocas, J. M. Kleinberg, and K. Levy, "Mitigating bias in algorithmic employment screening: Evaluating claims and practices," *CoRR*, vol. abs/1906.09208, 2019. arXiv: 1906.09208.

[78] M. Feldman, O. Svensson, and R. Zenklusen, "A framework for the secretary problem on the intersection of matroids," *CoRR*, vol. abs/1704.02608, 2017. arXiv: 1704.02608.

[79] T. Thomas, "Reconsidering the remedy of gender quotas," *Harvard Journal of Law & Gender Online*, 2016.

[80] C. Duffy, "Plans at microsoft and wells fargo to increase black leadership are under scrutiny from the labor dept.," *CNN Business*, Oct. 2020.

[81] P. Kim, "Race-aware algorithms: Fairness, nondiscrimination and affirmative action," *California Law Review, Forthcoming*, 2022.

[82] J. Salem, D. Desai, and S. Gupta, "Don't let ricci v. destefano hold you back: A bias-aware legal solution to the hiring paradox," in *2022 ACM Conference on Fairness, Accountability, and Transparency*, ser. FAccT '22, Seoul, Republic of Korea: Association for Computing Machinery, 2022, pp. 651–666, ISBN: 9781450393522.

[83] D. Desai, S. Gupta, and J. Salem, "Using algorithms to tame discrimination: A path to algorithmic diversity, equity, and inclusion," *Conditionally accepted to the UC Davis Law Review*, 2023.

[84] J. Salem, S. Gupta, and V. Kamble, "Algorithmic challenges in ensuring fairness at the time of decision," *arXiv preprint arXiv:2103.09287*, 2021.

[85] G.-Y. Ban and N. B. Keskin, "Personalized dynamic pricing with machine learning: High-dimensional features and heterogeneous elasticity," *Management Science*, vol. 67, no. 9, pp. 5549–5568, 2021.

[86] S. Qiang and M. Bayati, "Dynamic pricing with demand covariates," *arXiv preprint arXiv:1604.07463*, 2016.

[87] J. Broder and P. Rusmevichientong, "Dynamic pricing under a general parametric choice model," *Operations Research*, vol. 60, no. 4, pp. 965–980, 2012.

[88] M. C. Cohen, A. N. Elmachtoub, and X. Lei, "Price discrimination with fairness constraints," *Management Science*, vol. 68, no. 12, pp. 8536–8552, 2022.

[89] K. T. Hansen, K. Misra, and M. M. Pai, "Frontiers: Algorithmic collusion: Supra-competitive prices via independent algorithms," *Marketing Science*, vol. 40, no. 1, pp. 1–12, 2021.

[90] W. L. Cooper, T. Homem-de-Mello, and A. J. Kleywegt, "Learning and pricing with models that do not explicitly incorporate competition," *Operations research*, vol. 63, no. 1, pp. 86–103, 2015.

[91] J. Kiefer, "Sequential minimax search for a maximum," *Proceedings of the American mathematical society*, vol. 4, no. 3, pp. 502–506, 1953.

[92] Y. Nesterov and V. Spokoiny, "Random gradient-free minimization of convex functions," *Foundations of Computational Mathematics*, vol. 17, no. 2, pp. 527–566, 2017.

[93] J. C. Spall *et al.*, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE transactions on automatic control*, vol. 37, no. 3, pp. 332–341, 1992.

[94] A. D. Flaxman, A. T. Kalai, and H. B. McMahan, "Online convex optimization in the bandit setting: Gradient descent without a gradient," in *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, 2005, pp. 385–394.

[95] E. W. Cope, "Regret and convergence bounds for a class of continuum-armed bandit problems," *IEEE Transactions on Automatic Control*, vol. 54, no. 6, pp. 1243–1253, 2009.

[96] J. Kiefer and J. Wolfowitz, "Stochastic estimation of the maximum of a regression function," *The Annals of Mathematical Statistics*, vol. 23, no. 3, pp. 462–466, 1952.

[97] A. Agarwal, D. P. Foster, D. Hsu, S. M. Kakade, and A. Rakhlin, "Stochastic convex optimization with bandit feedback," in *Proceedings of the 24th International Conference on Neural Information Processing Systems*, 2011, pp. 1035–1043.

[98] S. Jia, A. Li, and R. Ravi, "Markdown pricing under unknown demand," *Available at SSRN 3861379*, 2021.

[99] N. Chen, "Multi-armed bandit requiring monotone arm sequences," *arXiv preprint arXiv:2106.03790*, 2021.

[100] J. Salem, S. Gupta, and V. Kamble, "Taming wild price fluctuations: Monotone stochastic convex optimization with bandit feedback," *arXiv preprint arXiv:2103.09287*, 2021.

[101] P. R. Darke and D. W. Dahl, "Fairness and discounts: The subjective value of a bargain," *Journal of Consumer psychology*, vol. 13, no. 3, pp. 328–338, 2003.

[102] L. E. Bolton, L. Warlop, and J. W. Alba, "Consumer perceptions of price (un) fairness," *Journal of consumer research*, vol. 29, no. 4, pp. 474–491, 2003.

[103] J. Heyman and B. A. Mellers, "Perceptions of fair pricing," 2008.

[104] L. Xia, K. Monroe, J. Cox, B. Kent, K. Monroe, and J. Jones, "The price is unfair! a conceptual framework of price fairness perceptions," *Journal of Marketing*, vol. 68, pp. 1–15, Nov. 2004.

[105] C. Duffy, "In the face of a cultural reckoning, it turns out massive corporations can move fast and fix things," *CNN Business*, Jun. 2020.

[106] D. Bass and J. Eidelson, "Microsoft plan to add black executives draws u.s. labor inquiry," *Seattle Times*, Oct. 2020.

[107] C. Duffy, "Adidas says at least 30% of new us positions will be filled by black or latinx people," *CNN Business*, Jun. 2020.

[108] OFCCP, "U.s. department of labor and microsoft corp. enter agreement to resolve alleged hiring discrimination affecting 1,229 applicants in four states," *CNN Business*, Sep. 2020.

[109] C. L. Estlund, "Putting grutter to work: Diversity, integration, and affirmative action," *Berkeley J. of Labor and Employment*, vol. 26, p. 1, 2005.

[110] Title VII, 29 CFR Parts 1600, 1607, 1608. 2006.

[111] EEOC Guidelines on Affirmative Action, 29 C.F.R. § 1608.1(c). 1979.

[112] J. March, "Exploration and exploitation in organizational learning," *Organizational Science*, vol. 2, p. 71, 1989.

[113] P. Domingos, *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World*, 1st. New York, NY: Basic Books, 2015.

[114] D. R. Desai and J. A. Kroll, "Trust but verify: A guide to algorithms and the law," *Harv. JL & Tech.*, vol. 31, p. 1, 2017.

[115] J. R. Bent, "Is algorithmic affirmative action legal," *Geo. LJ*, vol. 108, p. 803, 2019.

[116] O. Schwartz, "Untold history of AI: Algorithmic bias was born in the 1980s," *IEEE Spectrum*, 2019.

[117] J. Shields, *Over 98% of fortune 500 companies use applicant tracking systems (ats)*, 2018.

[118] J. Angwin, N. Scheiber, and A. Tobin, "Dozens of companies are using facebook to exclude older workers from job ads," *ProPublica, December*, 2017.

[119] P. T. Kim, "Manipulating opportunity," *Va. L. Rev.*, vol. 106, p. 867, 2020.

[120] T. S. Mohr, "Why women don't apply for jobs unless they're 100% qualified," *Harvard Business Review*, vol. 25, 2014.

[121] R. Goodman, *Why amazon's automated hiring tool discriminated against women*, Published Oct. 12, 2018. Last accessed Jun. 6, 2019., 2018.

[122] S. Yucer, S. Akçay, N. Al-Moubayed, and T. P. Breckon, "Exploring racial bias within face recognition via per-subject adversarially-enabled data augmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 18–19.

[123] L. Dixon, J. Li, J. Sorensen, N. Thain, and L. Vasserman, "Measuring and mitigating unintended bias in text classification," in *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 2018, pp. 67–73.

[124] S. D. Schlachter and J. R. Pieper, "Employee referral hiring in organizations: An integrative conceptual review, model, and agenda for future research.," *Journal of Applied Psychology*, 2019.

[125] A. Hannák, C. Wagner, D. Garcia, A. Mislove, M. Strohmaier, and C. Wilson, "Bias in online freelance marketplaces: Evidence from taskrabbit and fiverr," in *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, 2017, pp. 1914–1933.

[126] A. B. Batastini, A. D. Bolaños, R. D. Morgan, and S. M. Mitchell, "Bias in hiring applicants with mental illness and criminal justice involvement: A follow-up study with employers," *Criminal Justice and Behavior*, vol. 44, no. 6, pp. 777–795, 2017.

[127] V. Das Swain, K. Saha, M. D. Reddy, H. Rajvanshy, G. D. Abowd, and M. De Choudhury, "Modeling organizational culture with workplace experiences shared on glassdoor," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–15.

[128] C. Hanks, "Technology and values: Essential readings," in John Wiley & Sons, 2009, p. 7.

[129] M. Kearns and A. Roth, *The Ethical Algorithm: The Science of Socially Aware Algorithm Design*, 1st. New York, NY: oxford University Press, 2020.

[130] S. Barocas and A. D. Selbst, “Big data’s disparate impact,” *Calif. L. Rev.*, vol. 104, p. 671, 2016.

[131] A. Caliskan, J. J. Bryson, and A. Narayanan, “Semantics derived automatically from language corpora contain human-like biases,” *Science*, vol. 356, no. 6334, pp. 183–186, 2017.

[132] B. Edelman, M. Luca, and D. Svirsky, “Racial discrimination in the sharing economy: Evidence from a field experiment,” *American Economic Journal: Applied Economics*, vol. 9, no. 2, pp. 1–22, 2017.

[133] M. Bogen and A. Rieke, “Help wanted: An examination of hiring algorithms, equity, and bias,” 2018.

[134] K. Lum and W. Isaac, “To predict and serve?” *Significance*, vol. 13, no. 5, pp. 14–19, 2016.

[135] S. Barocas, “Data mining and the discourse on discrimination,” in *Data Ethics Workshop, Conference on Knowledge Discovery and Data Mining*, 2014, pp. 1–4.

[136] S. Barocas, M. Hardt, and A. Narayanan, *Fairness and Machine Learning*. fairmlbook.org, 2019, http://www.fairmlbook.org.

[137] R. N. Hanna and L. L. Linden, “Discrimination in grading,” *American Economic Journal: Economic Policy*, vol. 4, pp. 146–68, Feb. 2012.

[138] Jobscan, *Applicant tracking systems*, Available at https://www.jobscan.co/applicant-tracking-systems., Accessed Sept. 11, 2020.

[139] S. Dewan, “How businesses use your sats,” *New York Times*, Mar. 2014.

[140] *Mckinsey’s online application faqs: Careers*, Accessed January 17, 2022.

[141] A. Griswold, “Why major companies like amazon ask job candidates for their sat scores,” *Business Insider*, Mar. 2014.

[142] E. Dixon-Roman, H. Everson, and J. Mcardle, “Race, poverty and sat scores: Modeling the influences of family income on black and white high school students’ sat performance,” *Teachers College Record*, vol. 115, May 2013.

[143] Y. Faenza, S. Gupta, and X. Zhang, *Impact of bias on school admissions and targeted interventions*, 2020. arXiv: 2004.10846 [cs.CY].

[144] S. A. Friedler, C. Scheidegger, S. Venkatasubramanian, S. Choudhary, E. P. Hamilton, and D. Roth, "A comparative study of fairness-enhancing interventions in machine learning," in *Proceedings of the conference on fairness, accountability, and transparency*, 2019, pp. 329–338.

[145] T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma, "Fairness-aware classifier with prejudice remover regularizer," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, 2012, pp. 35–50.

[146] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork, "Learning fair representations," in *International Conference on Machine Learning*, 2013, pp. 325–333.

[147] M. Hardt, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," in *Advances in neural information processing systems*, 2016, pp. 3315–3323.

[148] F. Kamiran, T. Calders, and M. Pechenizkiy, "Discrimination aware decision tree learning," in *2010 IEEE International Conference on Data Mining*, IEEE, 2010, pp. 869–874.

[149] M. De-Arteaga *et al.*, "Bias in bios: A case study of semantic representation bias in a high-stakes setting," in *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 2019, pp. 120–128.

[150] J. Salem and S. Gupta, "Closing the gap: Group-aware parallelization for online selection of candidates with biased evaluations," in *International Conference on Web and Internet Economics (WINE)*, Springer, 2020. Under major revision at Management Science, 2021.

[151] J. Wang and N. Shah, "Your 2 is my 1, your 3 is my 9: Handling arbitrary miscalibrations in ratings," in *AAMAS Conference proceedings*, 2019.

[152] J. Wang, I. Stelmakh, Y. Wei, and N. B. Shah, "Debiasing evaluations that are biased by evaluations," *arXiv preprint arXiv:2012.00714*, 2020.

[153] A. Wold and C. Wennerås, "Nepotism and sexism in peer review," *Nature*, vol. 387, no. 6631, pp. 341–343, 1997.

[154] A. Blum and K. Stangl, "Recovering from biased data: Can fairness constraints improve accuracy?" In *Symposium on Foundations of Responsible Computing (FORC)*, vol. 1, 2020.

[155] M. J. Fischer and D. S. Massey, “The effects of affirmative action in higher education,” *Social Science Research*, vol. 36, no. 2, pp. 531–549, 2007.

[156] M. E. Heilman, C. J. Block, and P. Stathatos, “The affirmative action stigma of incompetence: Effects of performance information ambiguity,” *Academy of Management Journal*, vol. 40, no. 3, pp. 603–625, 1997.

[157] FTC Report, *Big data: a tool for inclusion or exclusion?* Federal Trade Commission, January 2016.

[158] § 2000e-2(k)(1)(A). 42 U.S.C.

[159] C. A. Sullivan, “Disparate impact: Looking past the desert palace mirage,” *William & Mary Law Review*, 2005.

[160] W. Miao and J. L. Gastwirth, “Properties of statistical tests appropriate for the analysis of data in disparate impact cases,” *Law, Probability and Risk*, vol. 12, no. 1, pp. 37–61, 2013.

[161] M. G. Sobol and C. J. Ellard, “Measures of employment discrimination: A statistical alternative to the four-fifths rule,” *Industrial Relations Law Journal*, pp. 381–399, 1988.

[162] Bostock v. Clayton County, 590 U.S. ___. 2020.

[163] N. Totenberg. “Supreme court delivers major victory to lgbtq employees.” (2020).

[164] M. Kempner, “Georgia’s big businesses reveal staff — and management — diversit,” *The Atlanta Constitution Journal*, Oct. 2021.

[165] Griggs v. Duke Power Co., 401 U.S. 424, 432. 1971.

[166] Griggs v. Duke Power Co., 401 U.S. 424, 431. 1971.

[167] Ricci v. DeStefano, 557 U.S. 557, 632. 2009.

[168] Johnson v. Transportation Agency, Santa Clara Cty., 480 U.S. 616, 627. 1987.

[169] M. MacCarthy, “Standards of fairness for disparate impact assessment of big data algorithms,” *Cumberland L. Rev.*, vol. 48, p. 102, 2017.

[170] K. A. Houser, “Can ai solve the diversity problem in the tech industry? mitigating noise and bias in employment decision-making,” *Stanford Tech. L. Rev.*, vol. 22, p. 290, 2019.

[171] P. Kim, "Auditing algorithms for discrimination," *U. Pa. L. Rev. Online*, vol. 166, p. 189, 2017.

[172] Johnson v. Transportation Agency, Santa Clara Cty., 480 U.S. 616, 631. 1987.

[173] Ricci v. DeStefano, 557 U.S. 557. 2009.

[174] P. Kim, "Data-driven discrimination at work," *Wm. & Mary L. Rev.*, vol. 58, p. 8657, 2017.

[175] Ricci v. DeStefano, 557 U.S. 557, 580–581. 2009.

[176] Ricci v. DeStefano, 557 U.S. 557, 583. 2009.

[177] Ricci v. DeStefano, 557 U.S. 557, 564. 2009.

[178] Ricci v. DeStefano, 557 U.S. 557, 565. 2009.

[179] Ricci v. DeStefano, 557 U.S. 557, 567. 2009.

[180] Ricci v. DeStefano, 557 U.S. 557, 585. 2009.

[181] R. Primus, "The future of disparate impact," *Mich. L. Rev.*, vol. 108, p. 1341, 2010.

[182] Johnson v. Transportation Agency, Santa Clara Cty., 480 U.S. 616, 636–637. 1987.

[183] Johnson v. Transportation Agency, Santa Clara Cty., 480 U.S. 616, 638. 1987.

[184] Regents of University of California v. Bakke, 438 U.S. 265, 317. 1978.

[185] Brunet v. City of Columbus, Ohio, 58 F.3d 251, 255 (6th Cir.) 1995.

[186] Guardians Ass'n of New York City v. Civil Serv, 630 F.2d 79 (2d Cir.) 1980.

[187] Johnson v. Transportation Agency, Santa Clara Cty., 480 U.S. 616, 623–624. 1987.

[188] Kirkland v. N.Y. State Dep't of Correctional Serv., 711 F.2d 1117, 1133 (2d Cir.) 1983.

[189] Bradley v. City of Lynn, 403 F. Supp. 2d 161. 2005.

[190] Boston Police Superior Officers Fed'n v. City of Boston, 147 F.3d 13. 1998.

[191] Johnson v. Transportation Agency, Santa Clara Cty., 480 U.S. `616, 640-641`. 1987.

[192] Johnson v. Transportation Agency, Santa Clara Cty., 480 U.S. `616, 639`. 1987.

[193] Johnson v. Transportation Agency, Santa Clara Cty., 480 U.S. `616, 635`. 1987.

[194] S. Corbett-Davies and S. Goel, "The measure and mismeasure of fairness: A critical review of fair machine learning," *arXiv preprint arXiv:1808.00023*, 2018.

BIOGRAPHY

Jad Salem is an overly caffeinated mathematician, cooking enthusiast, and crossword lover. After studying mathematics in college and graduate school, Jad became interested in theoretical and practical aspects of fair decision-making.