

4th Symposium on Foundations of Responsible Computing

FORC 2023, June 7–9, 2023, Stanford University, California, USA

Edited by

Kunal Talwar



Editors

Kunal Talwar

Apple, Cupertino, CA, USA
ktalwar@apple.com

ACM Classification 2012

Applied computing; Computing Methodologies; Mathematics of computing; Security and Privacy; Social and professional topics; Theory of computation

ISBN 978-3-95977-272-3

Published online and open access by

Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany. Online available at <https://www.dagstuhl.de/dagpub/978-3-95977-272-3>.

Publication date

June, 2023

Bibliographic information published by the Deutsche Nationalbibliothek

The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at <https://portal.dnb.de>.

License

This work is licensed under a Creative Commons Attribution 4.0 International license (CC-BY 4.0):
<https://creativecommons.org/licenses/by/4.0/legalcode>.



In brief, this license authorizes each and everybody to share (to copy, distribute and transmit) the work under the following conditions, without impairing or restricting the authors' moral rights:

- Attribution: The work must be attributed to its authors.

The copyright is retained by the corresponding authors.

Digital Object Identifier: 10.4230/LIPIcs.FORC.2023.0

ISBN 978-3-95977-272-3

ISSN 1868-8969

<https://www.dagstuhl.de/lipics>

LIPICs – Leibniz International Proceedings in Informatics

LIPICs is a series of high-quality conference proceedings across all fields in informatics. LIPICs volumes are published according to the principle of Open Access, i.e., they are available online and free of charge.

Editorial Board

- Luca Aceto (*Chair*, Reykjavik University, IS and Gran Sasso Science Institute, IT)
- Christel Baier (TU Dresden, DE)
- Mikolaj Bojanczyk (University of Warsaw, PL)
- Roberto Di Cosmo (Inria and Université de Paris, FR)
- Faith Ellen (University of Toronto, CA)
- Javier Esparza (TU München, DE)
- Daniel Král' (Masaryk University - Brno, CZ)
- Meena Mahajan (Institute of Mathematical Sciences, Chennai, IN)
- Anca Muscholl (University of Bordeaux, FR)
- Chih-Hao Luke Ong (University of Oxford, GB and Nanyang Technological University, SG)
- Phillip Rogaway (University of California, Davis, US)
- Eva Rotenberg (Technical University of Denmark, Lyngby, DK)
- Raimund Seidel (Universität des Saarlandes, Saarbrücken, DE and Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Wadern, DE)

ISSN 1868-8969

<https://www.dagstuhl.de/lipics>

■ Contents

Preface	
<i>Kunal Talwar</i>	0:vii
Organizers	
.....	0:ix
Regular Papers	
From the Real Towards the Ideal: Risk Prediction in a Better World	
<i>Cynthia Dwork, Omer Reingold, and Guy N. Rothblum</i>	1:1–1:17
New Algorithms and Applications for Risk-Limiting Audits	
<i>Bar Karov and Moni Naor</i>	2:1–2:27
Bidding Strategies for Proportional Representation in Advertisement Campaigns	
<i>Inbal Livni Navon, Charlotte Peale, Omer Reingold, and Judy Hanwen Shen</i>	3:1–3:22
Multiplicative Metric Fairness Under Composition	
<i>Milan Mossé</i>	4:1–4:11
Setting Fair Incentives to Maximize Improvement	
<i>Saba Ahmadi, Hedyeh Beyhaghi, Avrim Blum, and Keziah Naggita</i>	5:1–5:22
Screening with Disadvantaged Agents	
<i>Hedyeh Beyhaghi, Modibo K. Camara, Jason Hartline, Aleck Johnsen, and Sheng Long</i>	6:1–6:20
Fair Grading Algorithms for Randomized Exams	
<i>Jiale Chen, Jason Hartline, and Onno Zoeter</i>	7:1–7:22
An Algorithmic Approach to Address Course Enrollment Challenges	
<i>Arpita Biswas, Yiduo Ke, Samir Khuller, and Quanquan C. Liu</i>	8:1–8:23
Fair Correlation Clustering in Forests	
<i>Katrin Casel, Tobias Friedrich, Martin Schirneck, and Simon Wietheger</i>	9:1–9:12
Distributionally Robust Data Join	
<i>Pranjal Awasthi, Christopher Jung, and Jamie Morgenstern</i>	10:1–10:15
Resistance to Timing Attacks for Sampling and Privacy Preserving Schemes	
<i>Yoav Ben Dov, Liron David, Moni Naor, and Elad Tzalik</i>	11:1–11:23



■ Preface

The Symposium on Foundations of Responsible Computing (FORC), now in its fourth year, is a forum for mathematically rigorous research in computation and society writ large. The Symposium aims to catalyze the formation of a community supportive of the application of theoretical computer science, statistics, economics, and other relevant analytical fields to problems of pressing and anticipated societal concern.

Twenty-seven papers were selected to appear at FORC 2023, held in Stanford, CA on June 7–9, 2023. These papers were selected by the program committee, with the help of additional expert reviewers, out of forty-nine submissions. FORC 2023 offered two submission tracks: archival-option (giving authors of selected papers the option to appear in this proceedings volume) and non-archival (in order to accommodate a variety of publication cultures, and to offer a venue to showcase FORC-relevant work that will appear or has recently appeared in another venue). Eleven archival-option and sixteen non-archival submissions were selected for the program.

A smaller sub-committee was formed to choose best paper awards. The paper *From the Real Towards the Ideal: Risk Prediction in a Better World* by Cynthia Dwork and Guy Rothblum was selected for the best paper award. Three papers were selected for the best student paper award: *New Algorithms and Applications for Risk-Limiting Audits* by Bar Karov and Moni Naor; *Fair Grading Algorithms for Randomized Exams* by Jiale Chen, Jason Hartline, and Onno Zoeter; and *Resistance to Timing Attacks for Sampling and Privacy Preserving Schemes* by Yoav Ben Dov, Liron David, Moni Naor, and Elad Tzalik.

Thank you to the entire program committee and to the external reviewers for their hard work during the review process. It has been an honor and a pleasure to work together with you to shape the program of this young conference. Finally, I would like to thank our generous sponsors: the Simons Collaboration on the Theory of Algorithmic Fairness for their conference support.

Kunal Talwar
Cupertino, CA
April 26, 2023



■ Organizers

Program Committee

Borja Balle
Amrita Roy Chowdhury
Aloni Cohen
Edith Cohen
Rachel Cummings
Sumegha Garg
Badih Ghazi
Parikshit Gopalan
Adam Tauman Kalai
Michael P. Kim
Katrina Ligett
Yishay Mansour
Ilya Mironov
Kobbi Nissim
Chara Podimata
Jessica Sorrell
Adam Smith
Nati Srebro
Thomas Steinke
Kunal Talwar (chair)

Steering Committee

Avrim Blum
Cynthia Dwork (co-chair)
Sampath Kannan
Jon Kleinberg
Shafi Goldwasser
Kobbi Nissim
Toni Pitassi
Omer Reingold (co-chair)
Guy Rothblum
Salvatore Ruggieri
Salil Vadhan
Adrian Weller

Best-paper awards sub-committee

Edith Cohen
Rachel Cummings
Adam Tauman Kalai
Adam Smith



From the Real Towards the Ideal: Risk Prediction in a Better World

Cynthia Dwork ✉

Harvard University, Cambridge, MA, USA

Omer Reingold ✉

Stanford University, CA, USA

Guy N. Rothblum ✉

Apple, Cupertino, CA, USA

Abstract

Prediction algorithms assign scores in $[0, 1]$ to individuals, often interpreted as “probabilities” of a positive outcome, for example, of repaying a loan or succeeding in a job. Success, however, rarely depends only on the individual: it is a function of the individual’s interaction with the environment, past and present. Environments do not treat all demographic groups equally.

We initiate the study of corrective transformations τ that map predictors of success in the real world to predictors in a better world. In the language of algorithmic fairness, letting p^* denote the true probabilities of success in the real, unfair, world, we characterize the transformations τ for which it is feasible to find a predictor \tilde{q} that is indistinguishable from $\tau(p^*)$. The problem is challenging because we do not have access to probabilities or even outcomes in a better world. Nor do we have access to probabilities p^* in the real world. The only data available for training are outcomes from the real world.

We obtain a complete characterization of when it is possible to learn predictors that are indistinguishable from $\tau(p^*)$, in the form of a simple-to-state criterion describing necessary and sufficient conditions for doing so. This criterion is inextricably bound with the very existence of uncertainty.

2012 ACM Subject Classification Theory of computation \rightarrow Theory and algorithms for application domains

Keywords and phrases Algorithmic Fairness, Affirmative Action, Learning, Predictions, Multicalibration, Outcome Indistinguishability

Digital Object Identifier 10.4230/LIPIcs.FORC.2023.1

Funding *Cynthia Dwork*: Partially supported by Microsoft, by the Simons Foundation Collaboration on the Theory of Algorithmic Fairness and a Sloan Foundation Grant 2020-13941.

Omer Reingold: Part of this work was done while the author was visiting Microsoft Research. Supported by the Simons Foundation Investigators Award 689988, Simons Foundation Collaboration on the Theory of Algorithmic Fairness and a Sloan Foundation Grant 2020-13941

Guy N. Rothblum: Part of this work was done while the author was at the Weizmann Institute of Science and while visiting Microsoft Research. This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No. 819702), from the Israel Science Foundation (grant number 5219/17), from the U.S-Israel Binational Science Foundation (grant No. 2018102) and from the Simons Collaboration on The Theory of Algorithmic Fairness.

1 Introduction

Prediction algorithms assign scores in $[0, 1]$ to individuals, often interpreted as “probabilities” of a positive outcome, for example, of repaying a loan or succeeding in a job. Success, however, rarely depends only on the individual: it is a function of the individual’s interaction with the



© Cynthia Dwork, Omer Reingold, and Guy N. Rothblum;
licensed under Creative Commons License CC-BY 4.0

4th Symposium on Foundations of Responsible Computing (FORC 2023).

Editor: Kunal Talwar; Article No. 1; pp. 1:1–1:17



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

environment, past and present. If we think of an individual x as a collection of features, *past* interaction affects those very features; that is, the accomplishments that individuals bring to a potential new job depend heavily on the opportunities afforded to the them and their families in the past. In addition, given a collection of features x , an individual’s chance of a positive outcome depends heavily on the *future* environment in which the individual will be operating; for example, a woman with a given degree of talent and experience is less likely to succeed at a news organization that is hostile to women than at an organization supportive of women.

We initiate the study of corrective transformations τ that map predictors of success in the real world to predictors in a better world. In the language of algorithmic fairness, letting p^* denote the true probabilities of success in the real, unfair, world, we characterize the transformations τ for which it is feasible to find a predictor \tilde{q} that is indistinguishable from $\tau(p^*)$. The problem is challenging because we do not have access to probabilities or even outcomes in a better world. Nor do we have access to probabilities p^* in the real world. The only data available for training are outcomes from the real world.

The meaning of a “probability” for a non-repeatable event is the subject of much debate [1], giving rise to the question of what we should want from an *ideal* scoring function. In one view, known as *Outcome Indistinguishability*, the scores offer a model for the real world, and we want the modeled world to be indistinguishable from the real world; this leads to a hierarchy of demands, according to the degree of access to the scoring function that is granted to the distinguisher [3]. A different, but compatible, view arises from the perspective of algorithmic fairness. Speaking informally, a scoring function is multi-calibrated with respect to a collection \mathcal{C} of arbitrarily intersecting subsets of the population if it is calibrated simultaneously on each $S \in \mathcal{C}$ when viewed in isolation [6]. The sets in \mathcal{C} need not be restricted to the demographic groups often described as “protected sets,” but can (and should) capture conditions that are predictive of positive or negative outcomes. With this flexibility in mind, it is perhaps not surprising that multi-calibration has been shown to be equivalent to the second level of the outcome-indistinguishability hierarchy [3]. We use the term “MC/OI” to denote these equivalent properties.

Happily, MC/OI predictors can be learned from real-world Boolean outcomes data $o^*(x) \sim \text{Ber}(p^*(x))$, without access to p^* [6]. Now, consider a corrective transformation τ mapping individual-score pairs $(x, p^*(x))$ to $[0, 1]$, where the intuition is that $q^*(x) = [\tau(p^*)](x)$ is the probability of a positive outcome in a better world for the individual whose features in the real world are given by x . Not only do we not have access to q^* , but we do not even have outcomes data for the better world – that world does not exist! How, then, can we hope to construct a predictor that is indistinguishable from q^* ? That is the problem studied in this work: for what kinds of corrective transformations τ can we obtain a predictor \tilde{q} that is MC/OI with respect to q^* ?

Taxonomy of transformations. We consider three kinds of corrective transformations. The conceptually simplest is fully deterministic transformations τ that are specified with no access to the underlying distribution \mathcal{D}^* . Due to the deterministic nature of the transformation, the transformed predictor $\tau(p)$ is completely and uniquely defined for any given predictor p . For example, the transformation that raises scores for members of a set S , setting $[\tau(p^*)](x) = \min\{p^*(x) + 0.2, 1\}$ for $x \in S$, is fully deterministic.

More generally, we consider *parameterized* transformations τ_π , where the parameters π are obtained via an efficient parameter-learning algorithm that operates on instance-outcome samples $(x, o^*(x))$ for $x \sim \mathcal{D}_X$, where $o^*(x) \sim \text{Ber}(p^*(x))$. Here we must be careful in

defining $\tau_\pi(p)$, as different randomness – in the samples seen by the parameter-learner and in random coins it may use – will lead to different choices of π . We also allow the resulting transformation τ_π to be randomized. We informally and implicitly cover all these sources of randomness when we say that the transformation is randomized.

For example, suppose we have disjoint groups A and B and the goal of the transformation is to ensure statistical parity, so that in the transformed world the probabilities of a positive outcome are equalized between the two groups. The exact transformation depends the disparity in the real world, p^* , between the two group, *i.e.*, the difference between $p_A \stackrel{\text{def}}{=} \mathbf{E}_{x \in A}[p^*(x)] = \mathbf{E}_{x \in A}[o_x^*]$, and $p_B \stackrel{\text{def}}{=} \mathbf{E}_{x \in B}[p^*(x)] = \mathbf{E}_{x \in B}[o_x^*]$. Both of these quantities can be estimated from real-world outcomes data during the parameter-learning phase, and from these one can approximately determine $\alpha \in [0, 1] = \frac{p_A - p_B}{1 - p_B}$ such that the transformation τ_α that leaves scores unchanged for members of A and sets the new score for members $x \in B$ to $[\tau_\alpha(p^*)](x) = \alpha + (1 - \alpha)p^*(x)$ satisfies $E_{x \in A}[(\tau(p^*))(x)] \approx E_{x \in B}[(\tau(p^*))(x)]$.

In a third type of transformation the parameter-learner \mathcal{L} has access to p^* . For example, consider a population with two disjoint subgroups S, T . A predictor achieves *balance for the positive class* [9] if the average score assigned to positive instances in S equals the average score assigned to positive instances in T . Now, consider a transformation that takes an arbitrary predictor p as input and produces a transformed $\tau(p)$ satisfying the balance condition. To do this, the parameter-learner needs access to the average p^* values for the members of T and of S . For example, suppose that $\forall x \in T, p^*(x) = 0.8$, and $\forall x \in S, p^*(x) = 0.2$. Ensuring balance for the positive class can then be achieved by setting $[\tau(p^*)](x) = 0.8$ for all members of S and setting $[\tau(p^*)](x) = p^*(x)$ for all members of T . Of course, our algorithms cannot have access to p^* , but the prospect of building a predictor that is multicalibrated with respect to $\tau(p^*)$ remains compelling.

Canonical transformed predictor. When the transformation is randomized, we cannot simply speak of $\tau(p^*)$, as this is a random variable. However, given all the sources of randomness and an initial predictor p , the expectation of the transformation $\tau(p)$, $\mathbb{C}[\tau(p)] \stackrel{\text{def}}{=} \mathbf{E}[\tau(p)]$, where the expectation is taken over the samples fed to the parameter-learner, as well as its randomness, and any randomness in the transformed predictor, is well defined. We refer to this as the *canonical* transformed predictor, and use the special symbol \mathbb{C} .

Uncertainty and randomized instantiations. A deep and unresolved question is whether uncertainty exists, or if instead it only appears to exist because of insufficient information about the state of the world and insufficient computing power to determine future outcomes. Thus, when we talk about real-life probabilities $p^*(x)$, we cannot know whether $p^*(x)$ must lie in $\{0, 1\}$ (determinism) or whether values in $(0, 1)$ are possible (uncertainty). In the real world, we only observe outcomes, not individual probabilities. If uncertainty exists, then real-world outcomes are consonant with a deterministic world p^{**} that is a specific *random instantiation* of the real-world probabilities p^* in which each x is assigned a probability $p^{**}(x) \sim \text{Ber}(p^*(x)) \in \{0, 1\}$.

If uncertainty exists, there are many different possible random instantiations of p^* . The central concept in a transformation τ is its robustness (or not) to random instantiations: Does $\mathbb{C}[\tau(p^*)]$ look like $E_{p^{**} \leftarrow \text{RI}(p^*)}[\mathbb{C}[\tau(p^{**})]]$? For example, are their average values, over elements in a large set S , close in expectation? Example 1 above, in which scores of members of S are increased by 0.2 but capped at 1, is *not* robust to random instantiations. To see this, consider two possible choices of the real world p^* . In the first, $p_1^*(x) = 1/2$ for all $x \in S$; in the second, $p_2^*(x) = 0$ for a random half of the $x \in S$ and $p_2^*(x) = 1$ for the remainder of S . Note

that p_2^* is a random instantiation of p_1^* . The average scores for members of S are different under these two transformations: $\mathbf{E}_{x \in S}[(\tau(p_1^*))(x)] = 0.7$, but $\mathbf{E}_{x \in S}[(\tau(p_2^*))(x)] = 0.6$. The Balance for the Positive Class transformation described above also fails to be robust to random instantiations; in a nutshell, this is because in a random instantiation there is no uncertainty, and all positive members of S have $p^{**}(x) = 1$.

In contrast, the parameterized statistical parity transformation described above *is* robust to random instantiations. Roughly speaking this is because every random instantiation of p^* yields (almost) the same value of the parameter α , and for any large set S the average value $\mathbf{E}_{x \in S}[(\tau(p^*))(x)] \approx \mathbf{E}_{x \in S}[(\tau(p^{**}))(x)]$ depends only on α and the expectations $E_{x \in S \cap A}[p^*(x)]$ and $E_{x \in S \cap B}[p^*(x)]$. These expectations are invariant under random instantiations (assuming the sizes of $S \cap A, S \cap B$ are sufficiently large).

It is mathematically impossible, given only real-world instance-outcome pairs, to distinguish a real-world p^* in which probabilities are real-valued (uncertainty exists) and a real world which is a random instantiation p^{**} of such a p^* (no uncertainty), an epistemic state of affairs we summarize as follows.

Unresolvability Axiom: The question of whether uncertainty exists cannot be resolved by computing on finitely many samples from \mathcal{D}^* .

A Complete Characterization. Quite surprisingly, the concept of robustness to random instantiations provides a complete characterization of when it is possible to learn predictors that are indistinguishable from $q^* = \tau(p^*)$:

► **Theorem 1** (Main Theorem – informal). *There is a multiaccurate learning algorithm, and a multi-calibrated learning algorithm, with respect to $q^* = \tau(p^*)$, if and only if τ is robust to random instantiations.*

Thus, not only is it sometimes possible to build predictors for a transformed world, but there is a simple-to-state criterion describing necessary and sufficient conditions for doing so, and this criterion is inextricably bound with the very existence of uncertainty.

To prove sufficiency, we show how to exploit robustness to random instantiation to create samples of outcomes in the better world of q^* . This sample generation process involves partitioning samples from \mathcal{D}^* into groups, viewing each group as samples from an independent random instantiation of p^* , and using these capture, on average, the behavior of $\mathbf{E}_{p^{**} \leftarrow \text{RI}(p^*)}[\mathbb{C}[\tau(p^{**})]]$. By employing known algorithms we can build the desired predictors using these samples. We note that at no point does our multicalibration algorithm have access to the probabilities p^* or q^* ; everything is done given access only to *real-world outcomes data*.

To prove necessity, we argue that any transformation that is not robust to random instantiations must behave very differently on p^* than it behaves on random instantiations $p^{**} \leftarrow \text{RI}(p^*)$. In principle, this is detectable (although not efficiently!), which would resolve the question of whether uncertainty exists, contradicting the unresolvability axiom.

Stability. A final important *stability* notion tells us when multicalibration with respect to the transformed world $q^* = \tau(p^*)$ is meaningful. *Globally stable transformations* have the property that for every fixed distribution \mathcal{D}^* on instance-outcome pairs, $\tau(p^*)$ is *close* to its expectation $\mathbb{C}[\tau(p)]$. There is some flexibility in defining closeness; a natural choice is L_1 norm. In fact, a weaker condition suffices for our purposes. *Large-set stability* requires only that for any set S fixed *a priori*, with high probability over the samples and random bits fed to the learner (and the randomness of the transformed predictor, if it, too, is randomized),

the average prediction of $\tau(p^*)$ on $x \sim \mathcal{D}^*|_S$ is close to its expectation $\mathbb{C}[\tau(p^*)](x)$. In consequence, given a candidate q , large-set stability ensures that the average values of $q^* = \tau(p^*)$ on the level sets S_v of $q(S)$ are well-defined. This is crucial for reasoning about whether or not q is a multicalibrated with respect to q^* .

On related work. A vast body of work spanning many disciplines has studied corrective transformations to real-life (for example, works that study affirmative action). This body of work is too vast for us to survey here. Our work studies this question in the context of risk prediction and through the lens of algorithmic fairness. While fairness in risk prediction is a widely-studied topic in algorithmic fairness, the focus has been on learning a predictor that satisfies fairness desiderata while maintaining fidelity to the underlying distribution (e.g. [2, 6–8]), or on applying corrective transformations to learned risk predictors (e.g. [5]). Our work, on the other hand, initiates a study of learning about (probabilities in) a better world, where the better world is obtained by applying a corrective transformation on the real world itself.

2 Preliminaries, Setup and Definitions

Notation. For a distribution \mathcal{D} over domain \mathcal{X} , we use $\text{Supp}(\mathcal{D})$ to refer to the support of the distribution (the set of elements in \mathcal{X} that have non-zero probability). For $x \in \mathcal{X}$ we use $\mathcal{D}[x]$ to refer to x 's probability. For a subset $S \subseteq \mathcal{X}$ we use $\mathcal{D}[S]$ to refer to the aggregate probability of the set S under \mathcal{D} (i.e. $\mathcal{D}[S] = \sum_{x \in S} \mathcal{D}[x]$). For a set S with non-zero probability, we use $(\mathcal{D}|_S)$ to refer to the conditional distribution of \mathcal{D} , conditioned on landing in S .

Underlying all of this is a modeling assumption, in which “Nature” assigns a probability $p^*(x)$ to each individual x . We are agnostic as to whether $p^*(x) \in \{0, 1\}$ for all x or $p^*(x)$ can be arbitrary in $[0, 1]$. Since we cannot have access to p^* (we don't even know if it is real-valued!), the OI/MC literature builds scoring functions trained on outcomes $o^*(x)$ that Nature provides. However, the nomenclature “Nature” (inherited from a long literature on forecasting) is singularly inapt when viewed from a perspective of social justice, where one's “probability” of success and actual outcome are not solely intrinsic to the individual but are influenced – positively or negatively – by family wealth, structural racism, antisemitism, sexism, ableism, hetero-normativity, (lack of) availability of contraception and access to abortion, and so on. These are not forces of “Nature”, they are social forces that shape the reality in which we live.

We model real-life as a joint distribution over individuals and outcomes, denoted \mathcal{D}^* . An individual is described by a d -dimensional boolean string representing their “features”, and we focus on Boolean outcomes. Thus, \mathcal{D}^* is supported on $\{0, 1\}^d \times \{0, 1\}$. We refer to $\mathcal{X} = \{0, 1\}^d$ as the *feature space*, and use $x \sim \mathcal{D}_{\mathcal{X}}$ to denote a sample from real-life's marginal distribution over individuals.

A *predictor* is a function $p : \mathcal{X} \rightarrow [0, 1]$ that maps individuals to an estimate of the conditional probability of the individual's outcome being 1. For ease of notation, we use $p_x = p(x)$ to denote a predictor's estimate for individual x . The marginal distribution over individuals $\mathcal{D}_{\mathcal{X}}$ paired with a predictor induce a joint distribution over $\mathcal{X} \times \mathcal{Y}$. Given a predictor p , we use $(x, y) \sim \mathcal{D}(p)$ to denote an individual-outcome pair, where $x \sim \mathcal{D}_{\mathcal{X}}$ is sampled from real-life's distribution over individuals, and the outcome $y \sim \text{Ber}(p_x)$ is sampled – conditional on x – according to the Bernoulli distribution with parameter p_x . We use $p^* : \mathcal{X} \rightarrow [0, 1]$ to denote the marginal distribution on outcomes of real-life's distribution \mathcal{D}^* .

A *randomized instantiation* of a predictor p is the randomized process of fixing the prediction on each $x \in \mathcal{X}$ to be boolean, where the probability of 1 is exactly $p(x)$ (the boolean prediction for each x is drawn independently). We denote the (probabilistic) outcome of this process by $\text{RI}(p)$.

2.1 Multicalibration and Multiaccuracy

We start with the notion of multi-accuracy. Given a collection of subpopulations \mathcal{C} , multi-accuracy requires that a predictor \tilde{p} reflect the expectations of p^* correctly over each subpopulation $S \in \mathcal{C}$.

► **Definition 2** (Multi-Accuracy [6]). *Fix a feature distribution $\mathcal{D}_{\mathcal{X}}$ and a predictor $p^* : \mathcal{X} \rightarrow [0, 1]$. For a collection of sets $\mathcal{C} \subseteq \{0, 1\}^{\mathcal{X}}$ and $\alpha, \gamma \geq 0$, a predictor $\tilde{p} : \mathcal{X} \rightarrow [0, 1]$ satisfies $(\mathcal{C}, \alpha, \gamma)$ -multi-accuracy w.r.t. p^* (under the feature distribution $\mathcal{D}_{\mathcal{X}}$) if for every $S \in \mathcal{C}$ s.t. $\mathcal{D}_{\mathcal{X}}[S] \geq \gamma$:*

$$\left| \mathbf{E}_{x \sim \mathcal{D}_{\mathcal{X}}} [p^*(x) \mid x \in S] - \mathbf{E}_{x \sim \mathcal{D}_{\mathcal{X}}} [\tilde{p}(x) \mid x \in S] \right| \leq \alpha \quad (1)$$

Multi-calibration is a stronger notion, requiring the predictor \tilde{p} to be calibrated with respect to p^* over each $S \in \mathcal{C}$. Here, a set of predictions is calibrated if amongst the individuals $x \in \mathcal{X}$ who receive prediction $\tilde{p}(x) = v$, their actual expectation is v . For a set S and a value $v \in [0, 1]$, let S_v be the subset of S to which \tilde{p} assigns value v . We use $\text{supp}_S(\tilde{p}) = \{v \in [0, 1] : \mathbf{Pr}_{x \sim \mathcal{D}_{\mathcal{X}}} [\tilde{p}(x) = v \mid x \in S] > 0\}$ to denote the support of \tilde{p} on S (the set of values v s.t. S_v has non-zero mass).

► **Definition 3** (Multi-Calibration [6]). *Fix a feature distribution $\mathcal{D}_{\mathcal{X}}$ and a predictor $p^* : \mathcal{X} \rightarrow [0, 1]$. For a collection of sets $\mathcal{C} \subseteq \{0, 1\}^{\mathcal{X}}$ and parameters $\alpha, \gamma > 0$, a predictor $\tilde{p} : \mathcal{X} \rightarrow [0, 1]$ satisfies $(\mathcal{C}, \alpha, \gamma)$ -multi-calibration w.r.t. p^* (under the feature distribution $\mathcal{D}_{\mathcal{X}}$) if for every set $S \in \mathcal{C}$ s.t. $\mathcal{D}_{\mathcal{X}}[S] \geq \gamma$, there exists a set $S' \subseteq S$ with $\mathcal{D}_{\mathcal{X}}[S'] \geq (1 - \alpha)\mathcal{D}_{\mathcal{X}}[S]$ where:*

$$\forall v \in \text{supp}_{S'}(\tilde{p}) : \left| \mathbf{E}_{x \sim (\mathcal{D}_{\mathcal{X}}|_{S_{v'}})} [p^*(x)] - v \right| \leq \alpha. \quad (2)$$

When p^* is real-life's distribution, we simply refer to the predictor \tilde{p} as multi-calibrated or multi-accurate, but we will also discuss these requirements w.r.t predictors that are not real-life. We often assume that the predictor \tilde{p} is discretized to precision $\lambda = \Theta(\alpha)$ (see [6]).

3 Corrective Transformations

We study corrective transformations that will be applied to risk predictors. The transformation may include an optional *parameter-learning phase*. If the transformation does not use a learning phase, then we say that it is *fully explicit*. Otherwise, the transformation specifies a parameter-learner that can observe individual-outcome pairs drawn from the underlying distribution, or even observe individual-prediction pairs (see Definition 5). The learning phase outputs parameters π that are plugged into the transformation τ , which can be deterministic or probabilistic.

We begin by defining fully explicit and deterministic corrective transformations.

► **Definition 4** (Fully explicit and deterministic corrective transformation.). *A fully explicit (and deterministic) transformation is a mapping $\tau : \mathcal{X} \times [0, 1] \rightarrow [0, 1]$ that transforms a predictor p into a new predictor $\tau(p)$, where $\forall x \in \mathcal{X}$, $(\tau(p))(x) = \tau(x, p(x))$.*

Parameterized transformations (see above) also include a parameter-learning phase:

► **Definition 5** (Parameterized transformation τ). *A transformation is a pair (\mathcal{L}, τ) , where \mathcal{L} is a parameter-learning algorithm that gets access to training data (see below) and outputs parameters π . For any fixing of the parameters π , the mapping τ , using those parameters, transforms a predictor p into a new predictor $\tau_\pi(p)$, where $\forall x \in \mathcal{X}$, $(\tau_\pi(p))(x) = \tau_\pi(x, p(x))$.*

We consider different options for the parameter-learning algorithm \mathcal{L} and its training data:

- Fully-explicit transformation: *There is no parameter learning. The learner \mathcal{L} always outputs the empty string (if τ is deterministic, then this equivalent to Definition 4).*
- Outcome-based parameters: *The transformation is applied to a predictor p with respect to an underlying feature distribution $\mathcal{D}_\mathcal{X}$. The learner \mathcal{L} gets access to individual-outcome examples (x, o) , where $x \sim \mathcal{D}_\mathcal{X}$ and $o \sim \text{Ber}(p(X))$, and outputs parameters π .*
- Prediction-based parameters: *The transformation is applied to a predictor p with respect to an underlying feature distribution $\mathcal{D}_\mathcal{X}$. The learner \mathcal{L} gets access to individual-prediction examples $(x, p(x))$, where $x \sim \mathcal{D}_\mathcal{X}$, and outputs parameters π .*

We use $\pi \leftarrow \mathcal{L}^{\mathcal{D}_\mathcal{X}, p}$ to denote the process of running the parameter learner w.r.t a feature distribution $\mathcal{D}_\mathcal{X}$ and a predictor p , producing learned parameters π . We allow both the learner and the mapping τ to be randomized, and denote the random strings they use by $r_\mathcal{L}$ and r_τ (respectively).

We sometimes abuse notation and refer to the transformation as τ , where the parameter-learning algorithm is implicit. We also use $\tau(p)$ as shorthand for $\tau_\pi(p)$, where the parameters π are learned by the parameter-learning process.

3.1 Stable Transformations

Our primary focus is on transformations that are *stable* with respect to the choice of samples and random coins used by the learner, as well as the coins used by τ . We consider two definitions of stability: global stability, which requires that the resulting predictor is close to its expectation (globally, in L_1 distance). The more relaxed property of Large-set stability only requires that for any sufficiently large set (fixed a-priori), w.h.p. the average prediction is close to the expectation (the latter expectation is over the learner's and τ 's random choices).

► **Definition 6** (Canonical transformed predictor). *Fix a feature distribution $\mathcal{D}_\mathcal{X}$, a corrective transformation (\mathcal{L}, τ) , and a predictor p . The canonical transformed predictor is defined as:*

$$\mathbb{C}[\tau(p)] \stackrel{\text{def}}{=} \mathbf{E}_{\pi \leftarrow \mathcal{L}^{\mathcal{D}_\mathcal{X}, p}, r_\tau} [\tau_{\pi, r_\tau}(p)].$$

For the remainder of this writeup, We will reserve the special symbol “ \mathbb{C} ” to remind the reader that we are referring to the canonical predictor.

► **Definition 7** (Globally stable transformation). *Fix a feature distribution $\mathcal{D}_\mathcal{X}$. A transformation (\mathcal{L}, τ) is (α, β) -globally stable w.r.t. $\mathcal{D}_\mathcal{X}$ if for any predictor p , w.h.p. its (randomized) transformation $\tau(p)$ is close to the canonical transformed predictor in L_1 distance:*

$$\Pr_{\pi \leftarrow \mathcal{L}^{\mathcal{D}_\mathcal{X}, p}, r_\tau} \left[\mathbf{E}_{x \sim \mathcal{D}_\mathcal{X}} [|\tau_{\pi, r_\tau}(p)(x) - \mathbb{C}[\tau(p)](x)|] > \alpha \right] < \beta.$$

If (\mathcal{L}, τ) is (α, β) -globally stable for every distribution $\mathcal{D}_\mathcal{X}$ then we say that it is universally globally stable.

► **Definition 8** (Large-set stable (LSS) transformation). *Fix a feature distribution $\mathcal{D}_{\mathcal{X}}$ and let $\alpha, \beta : [0, 1] \rightarrow [0, 1]$ be functions bounding the magnitude and probability of instability as a function of the set size (see below). A transformation (\mathcal{L}, τ) is (α, β) -large set stable (LSS) w.r.t. $\mathcal{D}_{\mathcal{X}}$ if for any predictor p and for any fixed set $S \subseteq \mathcal{X}$, taking $\gamma = \Pr_{x \sim \mathcal{D}_{\mathcal{X}}}[S]$:*

$$\Pr_{\pi \leftarrow \mathcal{L}^{\mathcal{D}_{\mathcal{X}}, p, r_{\tau}}} \left[\left[\mathbf{E}_{x \sim (\mathcal{D}_{\mathcal{X}}|S)} [\tau_{\pi, r_{\tau}}(p)](x) - \mathbb{C}[\tau(p)](x) \right] > \alpha(\gamma) \right] < \beta(\gamma).$$

We emphasize that the absolute value in the above equation is external: we compare the expectation of $\tau(p)$ on the entire set S with the expectation of the canonical transformed predictor on that set.

If (\mathcal{L}, τ) is (α, β, γ) -LSS for every distribution $\mathcal{D}_{\mathcal{X}}$ then we say that it is universally LSS.

The error probability β will usually be exponentially small, so we can take a Union bound over large collections of sets, and conclude that w.h.p. for all of them simultaneously, the expectation of the transformed predictor is close to the expectation of the canonical transformed predictor.

We omit the “universally” or “w.r.t a particular distribution” suffix when they are clear from the context, simply referring to a corrective transformation as globally or large-set stable.

3.2 Our Goal: Evidence-Based Corrective Action

Once a corrective transformation is specified, our goal is learning a risk predictor that is “close to” the probabilities specified by the transformation, when it is applied to real-life’s probabilities p^* , i.e. close to $\tau(p^*)$. However, we can only observe *outcomes* by real-life’s distribution: the *probabilities* p^* are unknowable. Thus, we study the relaxed (but still significant!) goals of obtaining predictors that are multicalibrated or multiaccurate with respect to $\tau(p^*)$.

Here the importance of *stability* (see Section 3.1) becomes apparent: parameter learners are inherently randomized (as they draw samples), and there can also be additional randomization in \mathcal{L} or in τ . We want to be “close” to the transformed predictor, but which of the many possibly predictors in the support of $\tau(p^*)$ ’s output distribution should we aim to be close to? For stable transformations, the behavior of $\tau(p^*)$ on any (large enough) set is close to its expectation w.h.p. Thus, it is natural to aim to be close to the canonical transformed predictor $\mathbb{C}[\tau(p^*)]$:

► **Definition 9** (multiaccurate/multicalibrated learning algorithm for (\mathcal{L}, τ)). *Let (\mathcal{L}, τ) be a transformation. An algorithm \mathcal{A} for learning a multi-calibrated (respectively, multi-accurate) predictor for the transformation gets as input a collection of subsets $\mathcal{C} \subseteq 2^{\mathcal{X}}$, an error bound $\alpha \in [0, 1]$, a failure probability $\beta \in [0, 1]$, a set size $\gamma \in [0, 1]$, and labeled individual-outcomes pairs drawn from a distribution \mathcal{D}^* . Let $\mathbb{C}[\tau(p^*)]$ be the canonical transformation of p^* (see Definition 6).*

We say that \mathcal{A} is a $(\mathcal{C}, \alpha, \beta, \gamma)$ -multicalibration (respectively, multi-accuracy) learning algorithm for the transformation (\mathcal{L}, τ) if, when we run \mathcal{A} on input $(\mathcal{C}, \alpha, \beta, \gamma)$, with all but β probability over \mathcal{A} ’s random coin tosses and the training samples drawn i.i.d. from \mathcal{D}^ , it outputs a predictor \tilde{q} that is $(\mathcal{C}, \alpha, \gamma)$ multi-calibrated (respectively, $(\mathcal{C}, \alpha, \gamma)$ multi-accurate) w.r.t $\mathbb{C}[\tau(p^*)]$ (under the distribution $\mathcal{D}_{\mathcal{X}}^*$).*

Discussion. If (\mathcal{L}, τ) satisfies large-set stability (or the more stringent requirement of global stability), then multi-calibration w.r.t. $\mathbb{C}[\tau(p^*)]$ is quite meaningful: suppose \tilde{q} is a \mathcal{C} -multicalibrated predictor w.r.t. $\mathbb{C}[\tau(p^*)]$. Large-set stability implies that w.h.p. over the coins and samples of the transformation, for each set S in the collection \mathcal{C} , and for each (sufficiently large) level set S_v of \tilde{q} in S , the expectation of $\tau(p^*)$ (with the above random choices and samples) is close to the expectation by the canonical transformed predictor. Thus, with high probability over the coins and samples of the transformation, the predictions of \tilde{q} will be calibrated on all the sets in \mathcal{C} w.r.t. the (probabilistic) outcome of the corrective transformation applied to real-life. We find this to be a strong guarantee. Note that we assume here that the high probability guarantee is strong enough to allow union bounding over the sets in the collection and their prediction categories.

Multi-calibration with respect to $\mathbb{C}[\tau(p^*)]$ is not appropriate for corrective transformations that make arbitrary randomized distinctions between members of a protected class S , because random but “baseless” distinctions can nonetheless be averaged out in $\mathbb{C}[\tau(p^*)]$. For example, consider a protected group S where $p^* = 0.5$ for all members of S , because the data representation fails to capture appropriate features for members of S that permit accurate prediction¹. Suppose further that on $T = S^c$, half the elements have $p^*(x) = 1$ and half have $p^*(x) = 0$. One might consider a corrective τ that addresses the situation by arbitrarily assigning a random value in $\{0, 1\}$ to each member of S . This transformation is large-set stable (though it is very much *not* globally stable). However, we have that $\mathbb{C}[\tau(p^*)] = p^*$, so the effect of the transformation is “washed out” in the canonical transformed predictor, and in any \tilde{q} that is multicalibrated w.r.t. $\mathbb{C}[\tau(p^*)]$. One can argue that a corrective transformation, aiming to move the predictions towards a better world, should not make such arbitrary distinctions, and we are sympathetic to this argument. In the full version of this work we address this issue by including in the multicalibration set collect \mathcal{C} sets that may depend on the randomness used by the transformation τ . Finally, we remark that the above discussion is mainly for interpreting the positive direction of our characterization (*i.e.*, how meaningful is multicalibration with respect to $\mathbb{C}[\tau(p^*)]$). The negative direction characterizes the transformations for which achieving multicalibration with respect to $\mathbb{C}[\tau(p^*)]$ is impossible, regardless of how meaningful such a guarantee would be.

4 The Characterization

As discussed in the introduction (and the literature), we are agnostic on the question of whether real-life’s outcomes are deterministic (binary) or probabilistic. Our view is that this question is unanswerable, and thus corrective transformations should also be agnostic to it. We formalize this as a *robustness* property from the transformation (\mathcal{L}, τ) : we require that the canonical transformed predictor should be “similar” regardless of whether p^* is binary (deterministic) or not (probabilistic). Similarity is captured by requiring that $\mathbb{C}[\tau(p^*)]$ is close to the expectation, over a randomized instantiation p^{**} of p^* , of the canonical transformation of p^{**} . Closeness is measured in L_1 distance, and recall that each x ’s probability in p^{**} is binary, drawn from the Bernoulli distribution with expectation $p^*(x)$ (see Section 2). For example, this implies that (at least in expectation), the transformed probabilities should look similar regardless of whether real-life assigned a 0.5 probability to all the individuals, or whether the individuals were randomly partitioned into equally-sized sets with probability 0 and probability 1.

¹ See Chapter 4 of [4] for a real life example involving child protective services.

► **Definition 10** (Robustness to RI.). Fix a feature space \mathcal{X} and a distribution $\mathcal{D}_{\mathcal{X}}$ over features. A transformation (\mathcal{L}, τ) is (ε, δ) -robust to random instantiations w.r.t $\mathcal{D}_{\mathcal{X}}$ if for every predictor p :

$$\Pr_{x \sim \mathcal{D}_{\mathcal{X}}} \left[\left| \mathbb{C}[\tau(p)](x, p(x)) - \left(\mathbf{E}_{p' \leftarrow \text{RI}(p)} [\mathbb{C}[\tau(p')](x, p'(x))] \right) \right| > \varepsilon \right] \leq \delta$$

► **Theorem 11** (Main theorem: transformation characterization). Fix a feature space \mathcal{X} and a distribution $\mathcal{D}_{\mathcal{X}}$. Let (\mathcal{L}, τ) be a transformation. Then for every $\varepsilon, \delta > 0$:

- If (\mathcal{L}, τ) is (ε, δ) -robust to random instantiations (as per Definition 10), then there is an algorithm \mathcal{A} s.t. for every collection \mathcal{C} , and every $(\bar{\alpha}, \bar{\beta}, \bar{\gamma})$ s.t. $\bar{\alpha} = O((\delta/\bar{\gamma}) + \varepsilon)$, \mathcal{A} is a $(\mathcal{C}, \bar{\alpha}, \bar{\beta}, \bar{\gamma})$ multi-calibration learning algorithm for the transformation (\mathcal{L}, τ) . The sample complexity of \mathcal{A} is $\text{poly}(\log |\mathcal{C}|, 1/\bar{\alpha}, \log(1/\bar{\beta}), 1/\bar{\gamma})$.
- If (\mathcal{L}, τ) is not (ε, δ) -robust to random instantiations w.r.t $\mathcal{D}_{\mathcal{X}}$, then there exists a set S s.t. for any α, β s.t. $(\alpha + \beta) < (\varepsilon/2 - \text{negl})$ where negl bounds the probability that there is a feature-collision in the algorithm's training sample (some feature vector appears more than once), there is no $(\mathcal{C} = \{S\}, \alpha, \beta, \gamma = \delta/2)$ multi-accurate learning algorithm for the transformation.

Theorem 11 characterizes the transformations for which, for any given finite collection of sets \mathcal{C} , it is sample-theoretically possible to learn a predictor that is \mathcal{C} -multi-calibrated (or multi-accurate) with respect to $\mathbb{C}[\tau(p^*)]$. The positive direction constructs an algorithm whose sample complexity is logarithmic in $|\mathcal{C}|$, whereas the negative direction shows a *singleton* collection for which even multi-accuracy is impossible to obtain. The impossibility holds unless the algorithm uses sufficiently many samples to start observing “collisions” or repeated events (i.e. multiple instances of the same feature vector), whereas we are interested in the setting where events are non-repeatable. Thus, we think of the collision probability as negligible. Finally, the theorem does not assume the transformation is stable; our study of stability (Section 3.1) elucidates the qualitative *significance* of being multicalibrated with respect to $\mathbb{C}[\tau(p^*)]$, finding that the concept is meaningful under large-set stability.

Proof of Theorem 11.

Direction I: Non-Robustness \Rightarrow no multiaccuracy. If (\mathcal{L}, τ) is not δ -robust to random instantiations w.r.t $\mathcal{D}_{\mathcal{X}}$, then there exists a predictor $p : \mathcal{X} \rightarrow [0, 1]$ s.t.:

$$\Pr_{x \sim \mathcal{D}_{\mathcal{X}}} \left[\left| \mathbb{C}[\tau(p)](x, p(x)) - \left(\mathbf{E}_{p' \leftarrow \text{RI}(p)} [\mathbb{C}[\tau(p')](x, p'(x))] \right) \right| > \varepsilon \right] \geq \delta.$$

The above probability considers the absolute value of the difference between the two terms. Since the absolute value is large at least δ probability, there must be a subset $S \subseteq X$ (defined ex-post) where the predictions of the canonical transformed predictor are either significantly larger or significantly smaller than those of the canonical transformation of a randomized instantiation of p . Suppose w.l.o.g that the former is true, i.e. we have that:

$$\mathcal{D}_{\mathcal{X}}[S] \geq \frac{\delta}{2}, \tag{3}$$

and that:

$$\forall x \in S : \mathbb{C}[\tau(p)](x, p(x)) - \mathbf{E}_{p' \leftarrow \text{RI}(p)} [\mathbb{C}[\tau(p')](x, p'(x))] > \varepsilon. \tag{4}$$

Suppose towards contradiction that \mathcal{A} is an algorithm for learning a multi-accurate transformed predictor \tilde{q} . We run \mathcal{A} with parameters α, β (see below) and $\gamma = \delta/2$ and on the collection of sets $\{S\}$ (i.e. the collection is a singleton). \mathcal{A} gets i.i.d. feature-outcome samples $\{(x_i, y_i)\}$, where $x_i \in \mathcal{X}$ is sampled from $\mathcal{D}_{\mathcal{X}}$ and $y_i \in \{0, 1\}$ is Bernoulli with expectation $p^*(x)$. Consider two experiments of running \mathcal{A} with different p^* 's:

1. In Experiment 1, we set $p^* = p$.
2. In Experiment 2, we draw $p^* \leftarrow \text{Rl}(p)$.

In both experiments we run \mathcal{A} on outcomes drawn by p^* , and let \tilde{q} be the predictor that \mathcal{A} outputs.

Consider the random variables Q_1 and Q_2 , where Q_c is defined to be the value $\mathbf{E}_{x \sim (\mathcal{D}_{\mathcal{X}}|S)}[\tilde{q}(x)]$ in Experiment c (the RVs Q_1, Q_2 are over the domain $[0, 1]$). If \mathcal{A} is an $(\alpha, \beta, \gamma = \delta/2)$ -multiaccuracy learning algorithm for the transformation (\mathcal{L}, τ) , then since $\mathcal{D}_{\mathcal{X}}[S] \geq \gamma$ (see Equation (3)), by Definition 9:

$$\Pr \left[\left| Q_1 - \mathbf{E}_{x \sim (\mathcal{D}_{\mathcal{X}}|S)} [\mathbb{C}[\tau(p)](x, p(x))] \right| > \alpha \right] < \beta. \quad (5)$$

On the other hand, consider Experiment 2 and consider a *fixed* randomized instantiation p' (Experiment 2 includes the random process of drawing the randomized instantiation, whereas here we consider a fixed instantiation that has positive probability). Let $(Q_2|p')$ be the RV obtained by conditioning Q_2 on this fixed p' . Again, since $\mathcal{D}_{\mathcal{X}}[S] \geq \gamma$, by Definition 9:

$$\Pr \left[\left| (Q_2|p') - \mathbf{E}_{x \sim (\mathcal{D}_{\mathcal{X}}|S)} [\mathbb{C}[\tau(p')](x, p'(x))] \right| > \alpha \right] < \beta.$$

Experiment 2 consists of choosing a random instantiation p' , and then running the learning algorithm. By the above, adding an expectation over the randomized instantiation p' , we have that:

$$\left| \mathbf{E}[Q_2] - \mathbf{E}_{x \sim (\mathcal{D}_{\mathcal{X}}|S), p' \leftarrow \text{Rl}(p)} [\mathbb{C}[\tau(p')](x, p'(x))] \right| \leq \alpha + \beta. \quad (6)$$

Thus, by Equation (4), the value of Q_1 is w.h.p. higher than the expectation of Q_2 . This implies a lower bound on the statistical distance between Q_1 and Q_2

▷ **Claim 12.** $\Delta(Q_1, Q_2) > \frac{\varepsilon}{2} - \alpha - \beta$.

Proof. The proof follows by the fact that the expectations of two random variables supported on $[0, 1]$ cannot differ by more than their statistical distance:

$$\mathbf{E}[Q_1] - \mathbf{E}[Q_2] = \sum_{v \in [0,1]} (Q_1[v] \cdot v - Q_2[v] \cdot v) \leq \sum_{v \in [0,1]} |Q_1(v) - Q_2(v)| = 2\Delta(Q_1, Q_2).$$

Further, putting together Equations (4), (5) and (6) we conclude that:

$$E[Q_1] - E[Q_2] > \varepsilon - 2(\alpha + \beta).$$

The claim follows. ◁

The only difference between the two experiments is in the distributions of the feature-outcome samples fed to the learning algorithm. In particular, the difference is in the distribution of the binary outcomes: by p , or by a randomized instantiation of p . The feature-vectors are identically distributed in both experiments (i.i.d. from $\mathcal{D}_{\mathcal{X}}$). If the feature-vectors sampled by the learning algorithm are all distinct, then the conditional distributions on the outcomes in the two experiments (for those fixed feature vectors) are also identical: for each x , the outcome is Bernoulli with expectation $p(x)$. In Experiment 1 this is by design. In Experiment 2, this is due to the choice of a randomized instantiation p' of p , and so long as the samples are all distinct, the outcomes are drawn i.i.d. from the above distribution.

1:12 From the Real Towards the Ideal: Risk Prediction in a Better World

The only difference between the experiments is that if the same feature vector x is observed more than once, then in Experiment 1, the outcomes for the different occurrences of x will be independent, whereas in Experiment 2 they will be *identical* (since the predictor p' is instantiated once). The random variable Q is just a function of the algorithm's training sample. Thus, so long as the probability of observing the same feature vector more than once is negligible, we have:

▷ Claim 13. $\Delta(Q_1, Q_2) \leq \text{negl}$.

Claims 12 and 13 give a contradiction to the assumption that \mathcal{A} is a $(\alpha, \beta, \gamma = \delta/2)$ multiaccuracy algorithm for any values of α and β for which $\alpha + \beta < \frac{\varepsilon}{2} - \text{negl}$.

Direction II: Robustness implies calibration-feasibility. We construct an algorithm that learns a predictor that is multicalibrated with respect to the canonical transformation of p^* for any robust transformation. For a robust transformation, the canonical transformation of any p^* is close to the expectation, over a randomized instantiation p^{**} of p^* , of the canonical transformation of p^{**} . The main step in our algorithm is using outcomes drawn by p^* to generate outcomes whose distributions are close to $\mathbf{E}_{p^{**} \leftarrow \text{RI}(p^*)}[\mathbf{C}[\tau(p^{**})]]$. Robustness guarantees that this distribution is close to that of the canonical transformation of p^* . We then use a standard outcome-based multi-calibration learning algorithm (e.g. [6]), trained over the aforementioned samples, to obtain a predictor \tilde{q} that is multicalibrated w.r.t. the canonical transformation of p^* . The theorem follows.

Our goal, then, is generating outcomes that are close in distribution to $\mathbf{E}_{p^{**} \leftarrow \text{RI}(p^*)}[\mathbf{C}[\tau(p^{**})]]$. To do this, we treat the observed *outcomes* drawn by p^* as specifying *probabilities* according to a fictitious randomized instantiation p^{**} of p^* . These probabilities are fed into the (probability-based) parameter learner \mathcal{L} to learn parameters π for the transformation τ , towards applying it on (the fictitious) p^{**} . The key point is that these learned parameters will be *identically* distributed to parameters learned by \mathcal{L} on an actual randomized instantiation of p^* . Algorithm 1 details the sample-generation procedure.

The predictor q . Step 1 of the sample generation algorithm produces a set of learned parameters $\{\pi_i\}$. These parameters are then used in Step 2 to generate new samples, where we also take care (both in training and in sample generation) to ensure that the untransformed outcome for each feature vector $x \in X$ is consistent across all its appearances in training the i -th parameters and in generating samples. Fixing a run of Step 1 of the sample generator, for any fixed feature vector $x \in \mathcal{X}$ that is in the support of \mathcal{D}_X , let $q(x)$ denote the conditional probability that Step 2 produces the sample $(x, y' = 1)$ (conditioned on the feature vector x). The following claim shows that w.h.p. over the coins used in Step 1, for almost all x drawn from \mathcal{D}_X , the conditional probability $q(x)$ is close to the expectation, over a randomized instantiation p^{**} of p^* , of the probability assigned by the canonical transformed predictor. The notation $E_{q \leftarrow \text{Step 1}}$ emphasizes that we are taking expectation only over the randomness in the first step, in which the parameters $\{\pi_i, i \in [\ell]\}$ are learned, and not over the randomness in Step 2 in which a random $i \in [\ell]$ is selected.

▷ Claim 14. Fix parameters $\mu, \rho \in [0, 1]$. For the sample-generation algorithm (Algorithm 1) it holds that:

$$\Pr_{q \leftarrow \text{Step 1}, x \sim \mathcal{D}_X^*} \left[\left| q(x) - \mathbf{E}_{p^{**} \leftarrow \text{RI}(p^*)} [\mathbf{C}[\tau(p^{**})](x, p^{**}(x))] \right| \geq \mu \right] < \rho$$

Algorithm 1 Sample Generation for Robust Transformations.

Input: feature-outcome pairs, outcomes by p^* , error parameters $\mu, \rho \in [0, 1]$

Output: feature-outcome pairs, outcomes close to $\mathbf{E}_{p^{**} \leftarrow \text{RI}(p^*)}[\mathbb{C}[\tau(p^{**})]]$

1. Run $\ell = O(\sqrt{\log(1/\rho)}/\mu^2)$ indep. executions of the parameter learner \mathcal{L} . For each $i \in [\ell]$:
 - a. For every $x \in \mathcal{X}$, the i -th *untransformed outcome* o_x^i of x is initialized to be “undefined”.
 - b. The i -th execution uses freshly drawn random coins $r_{\mathcal{L},i}$.
 - c. To produce the j -th feature-probability sample requested by the i -th execution of \mathcal{L} , sample $(x_{i,j}, y_{i,j} \in \{0, 1\}) \sim \mathcal{D}(p^*)$. If $x_{i,j}$'s i -th untransformed outcome $o_{x_{i,j}}^i$ is defined, then proceed to the next step. Otherwise, set it to $y_{i,j}$.
 - d. Use $(x_{i,j}, o_{x_{i,j}}^i)$ as the j -th sample in the i -th execution of the parameter-learner.
 - e. The parameter-learner outputs parameters π_i .
 2. Produce each new feature-outcome output sample as follows:
 - a. Draw $(x, y \in \{0, 1\}) \sim \mathcal{D}(p^*)$. Pick $i \in [\ell]$ uniformly at random.
 - b. If x 's i -th untransformed outcome o_x^i is defined, then proceed to the next step. Otherwise, set it to y .
 - c. Draw $y' \in \{0, 1\}$ from the Bernoulli distribution with expectation $\tau_{\pi_i}(x, o_x^i)$ and output the sample (x, y') .
-

Proof. In Step 1 of the algorithm, consider a single execution i of the parameter-learning algorithm: the distribution of the learned parameters π_i is *identical* to the distribution of the parameters that would be learned by taking a randomized instantiation p^{**} of p^* : the randomized instantiation is simply determined by the observed binary outcomes (which are drawn by $p^*(x)$), where we take care to make sure that if a feature-vector x appears more than once in the training examples, then it is always “assigned” the binary outcome with which it first appeared (the i -th untransformed outcome is set only once). Moreover, we also take care that for any feature vector x that appears in Step 2, its untransformed outcome is set only once (when it first appeared, in training or in sample-generation for the i -th learned parameters).

Thus, for each $i \in [\ell]$, the distribution of outcomes that are generated in Step 2, conditioned on that using the i -th learned parameters, is identical to the distribution that would be obtained in a mental experiment, where we take a randomized instantiation $p_i^{**} \leftarrow \text{RI}(p^*)$, and learn the parameters π_i by training on examples drawn by p_i^{**} .

For x in the support of $\mathcal{D}_{\mathcal{X}}$, recall that $q(x)$ denotes the probability that the sample generator assigns outcome 1 to x . We conclude that q is in fact the average of ℓ predictors q_i , where each q_i is drawn by choosing a random instantiation of p^* and transforming it using (\mathcal{L}, τ) . Thus:

$$\begin{aligned}
 & \Pr_{q \leftarrow \text{Step 1}} \left[\left| q(x) - \mathbf{E}_{p^{**} \leftarrow \text{RI}(p^*)} [\mathbb{C}[\tau(p^{**})](x, p^{**}(x))] \right| > \mu \right] \\
 &= \Pr_{\{p_i^{**} \leftarrow \text{RI}(p^*), \pi_i\}_{i \in [\ell]}} \left[\left| \mathbf{E}_{i \in [\ell]} [\tau_{\pi_i}(x, p_i^{**}(x))] - \mathbf{E}_{p^{**} \leftarrow \text{RI}(p^*)} [\mathbb{C}[\tau(p^{**})](x, p^{**}(x))] \right| > \mu \right] \\
 &< \rho,
 \end{aligned}$$

1:14 From the Real Towards the Ideal: Risk Prediction in a Better World

where the first equality is by the mental experiment discussed above, and the second inequality is by a Chernoff bound. The above holds for any fixed x in the support of $\mathcal{D}_{\mathcal{X}}$, and thus it also holds for a randomly drawn $x \sim \mathcal{D}_{\mathcal{X}}$. \triangleleft

Speaking intuitively, Claim 14 tells us that, with high probability over the randomness in defining the building blocks of q , the resulting predictor is close to the expectation, over randomness in $p^{**} \leftarrow \text{RI}(p^*)$, of the canonical transformation of p^{**} . By the robustness of τ , this in turn is close to the canonical transformed $\mathbb{C}[\tau(p^*)]$. Hence, q is close to $\mathbb{C}[\tau(p^*)]$. The remainder of the proof will show that this closeness is maintained under multicalibration; that is, multicalibrating with respect to q yields a predictor that is close to something multicalibrated with respect to $\mathbb{C}[\tau(p^*)]$. Before proceeding with that argument, we first state a corollary that follows directly from Claim 14 via a standard argument.

► **Corollary 15.** *Fix parameters $\alpha', \beta', \sigma', \rho' \in [0, 1]$. For the sample-generation algorithm (Algorithm 1), run with parameters $\mu = \alpha'$ and $\rho = (\alpha' \cdot \beta' \cdot \sigma' \cdot \rho')$ it holds that:*

$$\Pr_{q \leftarrow \text{Step 1}} \left[\Pr_{x \sim \mathcal{D}_{\mathcal{X}}} \left[\left| q(x) - \mathbf{E}_{p^{**} \leftarrow \text{RI}(p^*)} [\mathbb{C}[\tau(p^{**})](x, p^{**}(x))] \right| > \alpha' \right] > (\alpha' \cdot \beta' \cdot \sigma' \cdot \rho') \right] < \beta'$$

Proof. Plugging the values of μ, ρ into Claim 14, we conclude that:

$$\Pr_{q \leftarrow \text{Step 1}, x \sim \mathcal{D}_{\mathcal{X}}} \left[\left| q(x) - \mathbf{E}_{p^{**} \leftarrow \text{RI}(p^*)} [\mathbb{C}[\tau(p^{**})](x, p^{**}(x))] \right| > \alpha' \right] < (\alpha' \cdot \beta' \cdot \sigma' \cdot \rho').$$

By a standard argument, it follows that it cannot be that with probability larger than β' over the q that is defined by Step 1, the probability, over $x \sim \mathcal{D}_{\mathcal{X}}$, that $q(x)$ is far from its “target” in the above equation is larger than $(\alpha' \cdot \sigma' \cdot \rho')$:

$$\Pr_{q \leftarrow \text{Step 1}} \left[\Pr_{x \sim \mathcal{D}_{\mathcal{X}}} \left[\left| q(x) - \mathbf{E}_{p^{**} \leftarrow \text{RI}(p^*)} [\mathbb{C}[\tau(p^{**})](x, p^{**}(x))] \right| > \alpha' \right] > (\alpha' \cdot \beta' \cdot \sigma' \cdot \rho') \right] < \beta'. \quad \blacktriangleleft$$

From MC w.r.t q to MC w.r.t. the canonical transformed predictor. Running a multicalibration algorithm on outcomes generated by the sample generation algorithm (Algorithm 1) will w.h.p. produce a predictor \tilde{q} that is approximately multicalibrated w.r.t. q . We use Corollary 15 and the robustness of the transformation (\mathcal{L}, τ) to show that \tilde{q} is also approximately MC w.r.t. the canonical transformation of p^* .

In more detail, let \mathcal{C} be the collection of sets, and let α, β, γ be parameters to be set below. We run the sample-generation algorithm (Algorithm 1) with parameters $\alpha' = \Theta(\alpha), \beta' = \Theta(\beta), \sigma' = \gamma, \rho' = \Theta(\alpha^2)$. By Corollary 15, with all but $\Theta(\beta)$ probability over the training in Step 1, the sample generator trains a predictor q for which there exists a “bad” set $B_q \subseteq \text{Supp}(\mathcal{D}_{\mathcal{X}})$ s.t. $\mathcal{D}_{\mathcal{X}}[B_q] \leq (\alpha^3 \cdot \gamma)/100$ where:

$$\forall x \in (\text{Supp}(\mathcal{D}_{\mathcal{X}}) \setminus B_q) : \left| q(x) - \mathbf{E}_{p^{**} \leftarrow \text{RI}(p^*)} [\mathbb{C}[\tau(p^{**})](x, p^{**}(x))] \right| \leq \alpha/100. \quad (7)$$

Further, by the (ε, δ) -robustness of the transformation (Definition 10), there exists a “bad” set $B_{\text{robust}} \subseteq \mathcal{X}$ where $\mathcal{D}_{\mathcal{X}}[B_{\text{robust}}] \leq \delta$ and

$$\forall x \in (\text{Supp}(\mathcal{D}_{\mathcal{X}}) \setminus B_{\text{robust}}) : \left| \mathbb{C}[\tau(p^*)](x, p^*(x)) - \left(\mathbf{E}_{p^{**} \leftarrow \text{RI}(p^*)} [\mathbb{C}[\tau(p^{**})](x, p^{**}(x))] \right) \right| \leq \varepsilon \quad (8)$$

We are now ready to analyze the guarantee of the multicalibrated predictor \tilde{q} w.r.t. the canonical transformation of p^* . We train \tilde{q} by running an outcome-based multicalibration algorithm on samples generated by Algorithm 1, where the MC algorithm is run on a collection of sets \mathcal{C} , and with parameters $\alpha'' = \Theta(\alpha)$, $\beta'' = \Theta(\beta)$ and $\gamma'' = \gamma$. Let \tilde{q} be the predictor trained by the MC learning algorithm. We assume w.l.o.g. that \tilde{q} is discretized to precision $\lambda = \Theta(\alpha)$. In what follows, we assume both that the MC algorithm does not fail (this happens with all but β'' probability), and that q trained by the sample generator satisfies Equation (7) (happens with all but β' probability). By a Union bound, this is the case with all but β probability.

Let $S \in \mathcal{C}$ be a set in the collection s.t. $\mathcal{D}_{\mathcal{X}}[S] \geq \gamma$. For a value $v \in [0, 1]$, let S_v be the subset of S to which \tilde{q} assigns value v . We define the “bad” level sets to be the elements assigned values v for which the set S_v has small mass by $\mathcal{D}_{\mathcal{X}}$:

$$B_{\text{levels}}(S) = \bigcup_{v \in [0, 1]: \mathcal{D}_{\mathcal{X}}(S_v) \leq (\alpha \cdot \lambda \cdot \gamma)/10} S_v, \quad (9)$$

where recall that the predictor was discretized to precision $\lambda = \Theta(\alpha)$, so there are at most $1/\lambda$ “level sets”. Thus, by construction, $\mathcal{D}_{\mathcal{X}}[B_{\text{levels}}(S)] \leq (\alpha \cdot \gamma)/10$.

By Definition 3, for any set $S \in \mathcal{C}$, s.t. $\mathcal{D}_{\mathcal{X}}[S] \geq \gamma$, there is a subset $S' \subseteq S$ where Equation (2) holds. Let S'' be the subset of S' that does not contain members of B_q , of B_{robust} , or of $B_{\text{levels}}(S)$. We have that:

$$\begin{aligned} \mathcal{D}_{\mathcal{X}}[S''] &\geq \mathcal{D}_{\mathcal{X}}[S'] - \mathcal{D}_{\mathcal{X}}[B_q] - \mathcal{D}_{\mathcal{X}}[B_{\text{robust}}] - \mathcal{D}_{\mathcal{X}}[B_{\text{levels}}(S)] \\ &\geq (1 - \alpha'') \mathcal{D}_{\mathcal{X}}[S] - \frac{\alpha^3 \gamma}{100} - \delta - \frac{\alpha \cdot \gamma}{10} \\ &\geq \left(1 - \alpha'' - \frac{\alpha^3}{100} - \frac{\delta}{\gamma} - \frac{\alpha}{10}\right) \mathcal{D}_{\mathcal{X}}[S] \\ &\geq \left(1 - \alpha - \frac{\delta}{\gamma}\right) \mathcal{D}_{\mathcal{X}}[S]. \end{aligned}$$

Since we removed the members of $B_{\text{levels}}(S)$ from S'' , it is the case that for every $v \in [0, 1]$ for which S''_v has non-zero mass, it has mass at least $(\alpha \cdot \lambda \cdot \gamma)/10$ (see Equation (9)). Thus:

$$\left| \mathbf{E}_{x \sim (\mathcal{D}_{\mathcal{X}}|_{S''_v})} [\mathbb{C}[\tau(p^*)](x, p^*(x))] - v \right| \leq \left| \mathbf{E}_{x \sim (\mathcal{D}_{\mathcal{X}}|_{S''_v})} [E_{p^{**} \leftarrow \text{RI}(p^*)} [\mathbb{C}[\tau(p^{**})](x, p^{**}(x))] - v] \right| + \varepsilon \quad (10)$$

$$\leq \left| \mathbf{E}_{x \sim (\mathcal{D}_{\mathcal{X}}|_{S''_v})} [q(x)] - v \right| + \varepsilon + \frac{\alpha}{100} \quad (11)$$

$$\leq \left| \mathbf{E}_{x \sim (\mathcal{D}_{\mathcal{X}}|_{S''_v})} [q(x)] - v \right| + \varepsilon + \frac{\alpha}{100} + \Theta\left(\alpha + \frac{\delta}{\alpha^2 \cdot \gamma}\right) \quad (12)$$

$$\leq \alpha'' + \varepsilon + \Theta\left(\alpha + \frac{\delta}{\alpha^2 \cdot \gamma}\right) \quad (13)$$

$$= \Theta\left(\alpha + \frac{\delta}{\alpha^2 \cdot \gamma}\right) + \varepsilon. \quad (14)$$

Where in the above: Equation (10) follows by the definition of S'' (which excludes elements in B_{robust} , and by Equation (8)). Equation (11) follows because S'' excludes elements in B_q (and by Equation (7)). In Equation (12) we switch the expectation from S''_v to S'_v using Proposition 16 below, which follows by standard manipulations. Finally, Equation (13) is by the multicalibration guarantee of \tilde{q} w.r.t q .

► **Proposition 16.** For $v \in [0, 1]$ s.t. S''_v has non-zero mass:

$$\left| \mathbf{E}_{x \sim (\mathcal{D}_{\mathcal{X}}|S'_v)} [q(x)] - \mathbf{E}_{x \sim (\mathcal{D}_{\mathcal{X}}|S''_v)} [q(x)] \right| = \Theta \left(\alpha + \frac{\delta}{\alpha^2 \cdot \gamma} \right)$$

Proof. The proof is by a case analysis on the sign of the difference in the absolute value. Suppose that the sign is positive, i.e. the first term is larger, then the absolute value is bounded by:

$$\begin{aligned} \mathbf{E}_{x \sim (\mathcal{D}_{\mathcal{X}}|S'_v)} [q(x)] - \mathbf{E}_{x \sim (\mathcal{D}_{\mathcal{X}}|S''_v)} [q(x)] &\leq \frac{1}{\mathcal{D}_{\mathcal{X}}[S''_v]} \cdot \left(\sum_{x \in S'_v} \mathcal{D}_{\mathcal{X}}[x] \cdot q(x) - \sum_{x \in S''_v} \mathcal{D}_{\mathcal{X}}[x] \cdot q(x) \right) \\ &= \frac{1}{\mathcal{D}_{\mathcal{X}}[S''_v]} \cdot \sum_{x \in (S'_v \setminus S''_v)} \mathcal{D}_{\mathcal{X}}[x] \cdot q(x) \\ &\leq \frac{\mathcal{D}_{\mathcal{X}}[(S'_v \setminus S''_v)]}{\mathcal{D}_{\mathcal{X}}[S''_v]} \\ &\leq \frac{(\alpha^3 \cdot \gamma/100) + \delta}{(\alpha \cdot \lambda \cdot \gamma)/10}. \end{aligned}$$

If the second term is larger, then the absolute value is bounded by:

$$\begin{aligned} \mathbf{E}_{x \sim (\mathcal{D}_{\mathcal{X}}|S''_v)} [q(x)] - \mathbf{E}_{x \sim (\mathcal{D}_{\mathcal{X}}|S'_v)} [q(x)] &= \frac{1}{\mathcal{D}_{\mathcal{X}}[S''_v]} \cdot \left(\sum_{x \in S''_v} \mathcal{D}_{\mathcal{X}}[x] \cdot q(x) - \frac{\mathcal{D}_{\mathcal{X}}[S''_v]}{\mathcal{D}_{\mathcal{X}}[S'_v]} \cdot \sum_{x \in S'_v} \mathcal{D}_{\mathcal{X}}[x] \cdot q(x) \right) \\ &\leq \frac{\sum_{x \in S''_v} \mathcal{D}_{\mathcal{X}}[x] \cdot q(x) - \frac{\mathcal{D}_{\mathcal{X}}[S''_v]}{\mathcal{D}_{\mathcal{X}}[S'_v]} \cdot \sum_{x \in S'_v} \mathcal{D}_{\mathcal{X}}[x] \cdot q(x)}{\mathcal{D}_{\mathcal{X}}[S''_v]} \\ &= \frac{\left(1 - \frac{\mathcal{D}_{\mathcal{X}}[S''_v]}{\mathcal{D}_{\mathcal{X}}[S'_v]}\right) \cdot \sum_{x \in S'_v} \mathcal{D}_{\mathcal{X}}[x] \cdot q(x)}{\mathcal{D}_{\mathcal{X}}[S''_v]} \\ &\leq \left(\frac{\mathcal{D}_{\mathcal{X}}[S'_v]}{\mathcal{D}_{\mathcal{X}}[S''_v]} - 1 \right) \cdot \frac{\mathcal{D}_{\mathcal{X}}[S'_v]}{\mathcal{D}_{\mathcal{X}}[S''_v]}, \end{aligned}$$

where the last inequality holds because $\frac{\mathcal{D}_{\mathcal{X}}[S''_v]}{\mathcal{D}_{\mathcal{X}}[S'_v]} \in (0, 1]$, and thus:

$$1 - \frac{\mathcal{D}_{\mathcal{X}}[S''_v]}{\mathcal{D}_{\mathcal{X}}[S'_v]} \leq \frac{1 - \frac{\mathcal{D}_{\mathcal{X}}[S''_v]}{\mathcal{D}_{\mathcal{X}}[S'_v]}}{\frac{\mathcal{D}_{\mathcal{X}}[S''_v]}{\mathcal{D}_{\mathcal{X}}[S'_v]}} = \frac{\mathcal{D}_{\mathcal{X}}[S'_v]}{\mathcal{D}_{\mathcal{X}}[S''_v]} - 1.$$

The claim follows by observing that:

$$\begin{aligned} \frac{\mathcal{D}_{\mathcal{X}}[S'_v]}{\mathcal{D}_{\mathcal{X}}[S''_v]} &\leq \frac{\mathcal{D}_{\mathcal{X}}[S''_v] + (\alpha^3 \cdot \gamma/100) + \delta}{\mathcal{D}_{\mathcal{X}}[S''_v]} \\ &\leq 1 + \frac{(\alpha^3 \cdot \gamma/100) + \delta}{(\alpha \cdot \lambda \cdot \gamma)/10} \end{aligned} \quad \blacktriangleleft$$

We conclude that, with all but β probability over the sample generation and learning procedures, \tilde{q} is $(\Theta(\alpha + \delta/(\alpha^2 \cdot \gamma)) + \varepsilon, \gamma)$ -multicalibrated w.r.t. the canonical transformation of p^* . The second direction of the theorem follows by setting $\beta = \bar{\beta}$, $\gamma = \bar{\gamma}$ and setting:

$$\alpha = \bar{\alpha} - \Theta \left((\delta/\gamma)^{1/3} \right) - \varepsilon.$$

The restriction on $\bar{\alpha}$ implies that $\alpha = \Omega(\bar{\alpha})$ (so the sample complexity of the multicalibrated learning algorithm will be polynomial in $(1/\bar{\alpha})$), and that $\alpha > (\delta/\gamma)^{1/3}$. Thus:

$$\Theta(\alpha + \delta/(\alpha^2 \cdot \gamma)) + \varepsilon \leq \Theta(\alpha + (\delta/\gamma)^{1/3}) + \varepsilon = \bar{\alpha}.$$

We conclude that the algorithm indeed achieves $(\bar{\alpha}, \bar{\beta}, \bar{\gamma})$ multicalibration w.r.t. the transformed predictor, and (this direction of) the theorem follows. \blacktriangleleft

References

- 1 Philip Dawid. On individual risk. *Synthese*, 194(9):3445–3474, 2017.
- 2 Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard S. Zemel. Fairness through awareness. In Shafi Goldwasser, editor, *Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012*, pages 214–226. ACM, 2012. doi:10.1145/2090236.2090255.
- 3 Cynthia Dwork, Michael P Kim, Omer Reingold, Guy N Rothblum, and Gal Yona. Outcome indistinguishability. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1095–1108, 2021.
- 4 Virginia Eubanks. *Automating Inequality: How High-Tech Tools Profile, Police, and Punish the Poor*. St. Martin’s Press, Inc., USA, 2018.
- 5 Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3315–3323, 2016. URL: <https://proceedings.neurips.cc/paper/2016/hash/9d2682367c3935defcb1f9e247a97c0d-Abstract.html>.
- 6 Ursula Hébert-Johnson, Michael Kim, Omer Reingold, and Guy Rothblum. Multicalibration: Calibration for the (computationally-identifiable) masses. In *International Conference on Machine Learning*, pages 1939–1948. PMLR, 2018.
- 7 Lunjia Hu, Inbal Livni Navon, Omer Reingold, and Chutong Yang. Omnipredictors for constrained optimization. *CoRR*, abs/2209.07463, 2022. doi:10.48550/arXiv.2209.07463.
- 8 Michael J. Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2569–2577. PMLR, 2018. URL: <http://proceedings.mlr.press/v80/kearns18a.html>.
- 9 Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. *arXiv preprint*, 2016. doi:10.48550/arXiv.1609.05807.

New Algorithms and Applications for Risk-Limiting Audits

Bar Karov ✉

Weizmann Institute of Science, Rehovot, Israel

Moni Naor ✉ 

Weizmann Institute of Science, Rehovot, Israel

Abstract

Risk-limiting audits (RLAs) are a significant tool in increasing confidence in the accuracy of elections. They consist of randomized algorithms which check that an election's vote tally, as reported by a vote tabulation system, corresponds to the correct candidates winning. If an initial vote count leads to the wrong election winner, an RLA guarantees to identify the error with high probability over its own randomness. These audits operate by sequentially sampling and examining ballots until they can either confirm the reported winner or identify the true winner.

The first part of this work suggests a new generic method, called “Batchcomp”, for converting classical (ballot-level) RLAs into ones that operate on batches. As a concrete application of the suggested method, we develop the first RLA for the Israeli Knesset elections, and convert it to one which operates on batches using “Batchcomp”. We ran this suggested method on the real results of recent Knesset elections.

The second part of this work suggests a new use-case for RLAs: verifying that a population census leads to the correct allocation of parliament seats to a nation's federal-states. We present an adaptation of ALPHA [12], an existing RLA method, to a method which applies to censuses. This suggested census RLA relies on data from both the census and from an additional procedure which is already conducted in many countries today, called a post-enumeration survey.

2012 ACM Subject Classification Applied computing → Voting / election technologies

Keywords and phrases Risk-Limiting Audit, RLA, Batch-Level RLA, Census

Digital Object Identifier 10.4230/LIPIcs.FORC.2023.2

Supplementary Material *Software (Source Code and Additional Plots):* <https://github.com/TGKar/Batch-and-Census-RLA>; archived at `swh:1:dir:9bd16e71658b0883e3ac966d48f81f48310fc9f3`

Funding Research supported in part by grants from the Israel Science Foundation (no.2686/20), by the Simons Foundation Collaboration on the Theory of Algorithmic Fairness and by the Israeli Council for Higher Education (CHE) via the Weizmann Data Science Research Center.

Moni Naor: Incumbent of the Judith Kleeman Professorial Chair.

1 Introduction

Running an election is a delicate endeavour, since casting and tallying votes entails seemingly contradictory requirements: counting the votes should be accurate and it must also be confidential. A risk-limiting audit (RLA) is a process whose goal is to increase the confidence that results of an election were tallied appropriately, or more accurately that the winner/s were chosen correctly. It is usually described for election systems where there is an electronic vote tabulation, whose tally is referred to as the **reported results**, but also backup paper-ballots, whose tally is assumed to be the **true results**. The procedure examines what is hopefully a relatively small number of the backup paper-ballots, and comparing them to the full reported results of the electronic voting system. These audits are randomized algorithms, where the randomization is manifested in the choice of ballots to examine, and potentially the order in which they are examined.



© Bar Karov and Moni Naor;
licensed under Creative Commons License CC-BY 4.0
4th Symposium on Foundations of Responsible Computing (FORC 2023).
Editor: Kunal Talwar; Article No. 2; pp. 2:1–2:27



Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

A risk-limiting audit ends either when the reported winners of the election are confirmed, or after a full recount of the backup paper-ballots of all voters. The audit’s goal is to confirm that the reported winners according to the electronic vote tabulation (the reported tally) match the winners according to the paper-backups (the true tally). Note that RLAs verify that the elections resulted in the correct winners according to the backup paper-ballots, and not that the reported vote tally was completely accurate; an RLA will approve election results that contain counting errors which do not change the winners of the elections. This fact is useful since it would be infeasible to expect the vote tally to be accurate up to every single ballot, but we should avoid at all cost counting errors which change the winners of the elections.

The claimed guarantee of RLAs is that if the reported winners of the elections are not correct (with regards to the full paper count), then the probability that the audit will mistakenly confirm the results is lower than some predetermined parameter, referred to as the *risk-limit* of the audit.

► **Definition 1. The RLA Guarantee:** *If the reported winners of the elections are not correct, an RLA will approve them w.p. of at most α , where α is a parameter which is set before the audit begins. α is referred to as the audit’s **risk-limit**.*

The efficiency of an RLA is measured by the number of paper-ballots it requires to read, given that the reported tally matches the true one. In most cases, an RLA should remain relatively efficient even if the reported tally isn’t completely accurate, as long as it results in the same winners as the true tally. The efficiency of any specific RLA method is limited by the election system it operates on. If a system has a sensitive social choice function, meaning that small tallying errors can often change the election winners, then it is more difficult to audit efficiently.

RLA methods generally belong to one of three categories, as defined by Lindeman and Stark [9]:

1. **Ballot-comparison:** In ballot-comparison audits, the auditor knows which paper-ballot matches which electronic-ballot. This category of audits is the most efficient, since it contains the most information about the election results.
2. **Ballot-polling:** In ballot-polling audits, a single paper-ballot can be sampled and examined, but it does not need to be matched to its corresponding electronic-ballot.
3. **Batch-level:** In batch-level audits, ballots are partitioned into batches, usually according to the precinct in which they were cast. The reported tally of each batch is available, but there’s no guarantee that a paper-ballot in the batch can be matched to its electronic counterpart. Ballots are usually not randomly partitioned, and different batches are of different sizes. Batch-level audits are generally the least efficient of the three categories, since the partition into batches is not random, making it more difficult to get a representative sample of the overall vote distribution.

1.1 Our Contributions

The goal of the work is to expand the realm where RLAs are used. Its new contributions are:

1. A new and general method for performing **batch-level RLAs**, which can be applied for many election systems, is presented in Section 3. This method, which we call “*Batchcomp*”, is usable for any election system that can be audited using the SHANGRLA framework [11].
2. An RLA method for the Israeli Knesset (The Israeli parliament) elections, based on the SHANGRLA framework, is presented in Section 4. This method can be applied as-is to conduct ballot-level RLAs, or be combined with Batchcomp to conduct a batch-level RLA.

To test both the Knesset RLA method and Batchcomp, we simulate their combination on real election results. While our Knesset RLA method is essentially a synthesis and adaptation of previous suggestions in the literature, it is the first time RLAs are applied to this setting.

3. A new type of RLA that applies to population censuses. This new type of audit is applicable in nations where political representatives are allocated to the nation’s geographical regions based on their population, like the United States, Germany, Cyprus and more. It relies on data that is already collected in many countries, as part of an existing method for assessing the accuracy of population censuses called a “post enumeration survey” (PES). To the best of our knowledge, this is the first and only method which verifies the census’ resulting allocation of representatives to federal-states with a clear statistical guarantee. The method is presented in Section 5.

2 Related Work

2.1 SHANGRLA

SHANGRLA [11] is an auditing framework which aids in adapting existing RLA algorithms to new social choice functions. It can be applied to a variety of election methods used globally, such as plurality, Hamiltonian elections [2], many proportional representation methods [1], and more.

This method is based on an abstraction called “sets of half-average nulls” (SHAN), where given a collection of finite lists containing unknown non-negative numbers, we wish to test whether the average of all of those lists is greater than $\frac{1}{2}$ by querying for the values at different indexes. Each query in this problem returns the values all lists hold at some specified index. An election system can be audited using SHANGRLA by reducing the problem of approving its reported winners to the SHAN problem. Once such a reduction is found, a number of existing algorithms [12, 11, 14] for the SHAN problem can be used to perform an RLA on that system.

This reduction is found by defining ℓ mappings a_1, \dots, a_ℓ from the paper-ballots to non-negative values, such that the mean of every mapping across all backup paper-ballots is above $\frac{1}{2}$ iff the reported winner/s of the election are true.

► **Definition 2.** *The reported winners of an election system can be audited using SHANGRLA if there exist ℓ non-negative functions a_1, \dots, a_ℓ , called **assertors**, such that the reported winners of the elections are true iff for every $k \in [\ell]$:*

$$\frac{1}{|B|} \sum_{b \in B} a_k(b) > \frac{1}{2}, \quad (1)$$

where B is a list of the backup paper-ballots of the elections. The ℓ inequalities above (for each $k \in [\ell]$) are referred to as the election’s **assertions**.

Some social choice functions have simple conversions to SHANGRLA assertions. E.g., a majority election between two candidates, Alice and Bob, can be audited using SHANGRLA with a single assorter. If Alice won the election according to the reported vote tally, this can be verified by using an assorter which has a mean greater than $\frac{1}{2}$ iff Alice truly received more votes than Bob:

► **Definition 3.** *An assorter which verifies that Alice received more votes from Bob in majority elections is:*

$$a(b) = \begin{cases} 1 & \text{if } b \text{ is for Alice} \\ 0 & \text{if } b \text{ is for Bob} \\ \frac{1}{2} & \text{if } b \text{ is invalid} \end{cases}$$

2.2 Finding SHANGRLA Assertions

In the example above, finding the correct assorter is relatively simple. For other election systems, which use more complicated social choice functions, verifying the correctness of the election winners can sometimes be reduced to verifying a set of linear inequalities regarding the various vote tallies. In such situations, it may not be immediately clear how to reduce them to assertions of the form $\frac{1}{|B|} \sum_{b \in B} a(b) > \frac{1}{2}$. For such cases, Blom et al. [1] suggests a generic solution. This solution reduces the problem of verifying that a set of linear inequalities that depend on the various vote tallies are all true to the problem of verifying that a set of assorters all have a mean greater than $\frac{1}{2}$ across all paper-ballots. We describe this solution for a single inequality. Given multiple inequalities, each inequality can be converted to a single SHANGRLA assertion in the same manner.

Say we have a linear inequality which is true iff the reported winner/s of some election system are the true ones:

$$\sum_{c \in \mathcal{C}} \beta_c v^{true}(c) > d, \quad (2)$$

where \mathcal{C} is the set of all ballots that a single voter may cast (e.g. in plurality elections, \mathcal{C} would be the set of candidates), $v^{true}(c)$ is the number of cast ballots of type c according to the true results, and d and β_c (for each $c \in \mathcal{C}$) are constants. To perform an RLA for this election system, we wish to find a SHANGRLA assertion which is equivalent to (2). Meaning, given (2), we wish to find a non-negative function $a: \mathcal{C} \rightarrow [0, \infty)$ such that (1) is equivalent to (2). As Blom et al. suggest, this can be achieved by defining:

$$a(b) := \frac{q - \beta_b}{2 \left(q - \frac{d}{|B|} \right)}, \quad (3)$$

where $q := \min_{c \in \mathcal{C}} \{\beta_c\}$, and β_b is determined by the type of ballot b is - if b is of type $c \in \mathcal{C}$, we have $\beta_b = \beta_c$. As noted by Blom et al., the assorters generated by this method are non-negative as long as the inequality they are derived from isn't trivially true or trivially false, for any distribution of votes.

2.3 The ALPHA Martingale Test

The ALPHA Martingale Test [12] is a specific RLA algorithm for election systems which have a SHANGRLA reduction as described in Section 2.1. I.e., when there exist ℓ assorters a_1, \dots, a_ℓ such that the reported winners of the elections are true iff for all $k \in [\ell]$ the inequality (1) is true.

The test operates by keeping ℓ variables T_1, \dots, T_ℓ , each representing the inverse of a p-value for the hypothesis that a certain list has an average greater than $\frac{1}{2}$. It then queries sequentially for random backup paper-ballots, where after each ballot it updates these ℓ variables. If at any point T_k for some $k \in [\ell]$ surpasses the threshold $\frac{1}{\alpha}$, it means

that we have sufficient evidence that the mean of its corresponding assorter a_k over all ballots is greater than $\frac{1}{2}$. If after a certain query, all of T_1, \dots, T_ℓ have surpassed $\frac{1}{\alpha}$ at some point during the audit, then the reported winners of the elections are approved.

After each queried backup paper-ballot b_i , the algorithm updates T_k for every $k \in [\ell]$ by comparing $a_k(b_i)$ to the following values, which are set before b_i is revealed:

- a. μ_k : The mean value of a_k over all ballots that have yet to be audited, given that the mean of a_k over all ballots is $\frac{1}{2}$. Recall that if the mean of a_k over all ballots is at most $\frac{1}{2}$, then the reported winners of the elections are wrong, which is the case the algorithm wishes to detect. Thus, if at some point during the audit we sample a ballot b with $a_k(b) \leq \mu_k$, it provides evidence that the reported winners of the elections are less likely to be correct, and vice-versa.
- b. η_k : A guess for what we would expect $a_k(b_i)$ to be based on the reported results and the ballots we previously queried. This guess can be made in several ways while maintaining the algorithm's correctness. One reasonable way to do so is to set η_k to be the mean of a_k over ballots that have yet to be audited, assuming that the reported tally is completely accurate. As explained by Stark [12], The audit becomes more efficient, meaning less ballots need to be examined, the more accurate this guess is.
- c. u_k : In the paper presenting ALPHA, u_k was defined as the maximal value a_k may return. In reality, the ALPHA Martingale Test is risk-limiting even for other choices of u_k , as long as the inequality $\mu_k < \eta_k < u_k$ is always maintained. For our purposes, u_k can be thought of as a guess for whether the next sampled ballot would indicate that assertion k is more or less likely to be true. If the next ballot to be sampled increases our confidence that the assertion is true, the audit is more efficient when u_k is large, and vice-versa.

The ALPHA Martingale Test can be adapted to sample ballots either with or without replacement. It can also be adapted to perform batch-level audits, where batches of ballots are sampled instead of individual ones. We refer to this batch-level version of the ALPHA Martingale Test as *ALPHA-Batch*. The Batchcomp method presented in Section 3 is based on ALPHA-Batch and attempts to improve on it by adjusting its assorters and utilizing the new definition for u_k .

3 The Batchcomp RLA

This section describes a generic way of performing batch-level RLAs, when the results of the elections can be verified using SHANGRLA assertions, as described in Section 2.1. This algorithm is original to this work and is based on ALPHA-Batch. Batchcomp relies on the following assumptions:

1. The election's social choice function can be audited using the SHANGRLA framework.
2. The reported and true results agree on the total number of ballots within each batch.

3.1 Model and Notation

Fix some elections system with a set of ballots B and a partition of these ballots into d batches B_1, \dots, B_d . We make no assumptions regarding this partition, and different batches may be of different size. By assuming that the election system can be audited using SHANGRLA, we assume the following:

► **Assumption.** *There are ℓ assorters a_1, \dots, a_ℓ such that the reported winners are true iff for all $k \in [\ell]$:*

$$\frac{1}{|B|} \sum_{b \in B} a_k(b) > \frac{1}{2}.$$

Throughout the following sections, we sometimes abuse notation and apply assorters over entire batches. When doing so, $a_k(B_i)$ is defined as the mean of a_k over all ballots in B_i :

$$a_k(B_i) := \frac{1}{|B_i|} \sum_{b \in B_i} a_k(b). \quad (4)$$

In accordance with this, $a_k(B)$ denotes the mean value of a_k across all ballots.

Finally, note that each batch has a reported tally, which is known before the audit begins, and a true tally, which we may only learn during the audit. Therefore, each assorter has a reported and true mean value over each batch, which can be calculated from its reported and true tally, respectively. We denote the reported mean of an assorter a_k over a batch B_i as $a_k^{rep}(B_i)$, and its true mean over that batch as $a_k^{true}(B_i)$. Using this notation, the audit's goal is to test whether $a_k^{true}(B) > \frac{1}{2}$ for all $k \in [\ell]$.

3.2 Batchcomp Overview

Batchcomp attempts to confirm that the mean of ℓ assorters over all ballots are all greater than $\frac{1}{2}$ by sequentially sampling batches of backup paper-ballots and examining them. In each iteration, it samples a previously unsampled batch, such that each batch is sampled w.p. proportional to its size.

After each sampled batch, it updates ℓ p-values, each corresponding to the hypothesis that an assorter has a mean greater than $\frac{1}{2}$ across all ballots. The algorithm keeps the inverses of these p-values, T_1, \dots, T_ℓ . Each variable T_k is updated according to the backup paper-ballots in the sampled batch and according to 3 additional variables - μ_k, η_k, U_k . μ_k and η_k are defined as they were in the ALPHA Martingale Test (see Section 2.3). U_k , which is Batchcomp's version of u_k from the ALPHA Martingale Test, controls how significantly T_k changes per audited batch. μ_k, η_k and U_k are updated after each iteration, while always maintaining $U_k > \eta_k > \mu_k$.

During the audit, Batchcomp uses a modified version of the election's assorters a_1, \dots, a_ℓ . We denote these modified assorters as A_1, \dots, A_ℓ . Each new assorter A_k has a mean greater than $\frac{1}{2}$ iff its corresponding assorter a_k also has a mean which is greater than $\frac{1}{2}$. Thus, to approve that the reported winners of the elections are correct, it suffices to approve that $A_k(B) > \frac{1}{2}$ for all $k \in [\ell]$. Auditing A_1, \dots, A_ℓ instead of a_1, \dots, a_ℓ makes the audit agnostic to the order in which batches are sampled, as long as the reported batch-level vote tallies are accurate. As explained in the following section, this can increase the audit's efficiency.

3.3 Comparing Batchcomp and ALPHA-Batch

The ALPHA-Batch method, which Batchcomp is based on, is performed by examining the mean of every assorter over each sampled batch according to its backup paper-ballots. It does not use the reported vote tally of the batches beyond the total number of ballots they contain. Batchcomp attempts to improve on the efficiency of ALPHA-Batch by auditing something slightly different - instead of auditing the mean value of an assorter a_k over the backup paper-ballots (true results) in a sampled batch, it audits the discrepancy between the mean value taken by a_k over a batch according to its reported tally, and the mean value it returns over the same batch according to its paper-ballots.

The values returned by the ALPHA-Batch assorters can change drastically from batch to batch, depending on their vote distribution according to the true results. The values the Batchcomp assorters return depend only on the accuracy of the reported tally; if two batches with different vote distributions were both counted accurately in the reported results, a Batchcomp assorter will return the same value when applied on each of them. This fact is shown in Section 3.4.

As an example of this, examine majority elections with accurate reported tallies. In such elections, ALPHA-Batch operates by applying the assorter from Definition 3 on the sampled batches. Applying this assorter on a batch returns the share of votes won by the reported winner of the elections inside that batch. This value can swing heavily depending on the specific batch that is sampled. A batchcomp assorter for the same elections returns the same value on every batch, regardless of the vote distribution within it.

Recall that before sampling and reading a backup paper-ballot, the ALPHA Martingale Test guesses the value that each assorter would return on this ballot (this guess is η_k , for each assorter a_k). As explained by Stark when presenting ALPHA [12], the audit is more efficient when these guesses are accurate. If each assorter returns a similar value for all batches, as happens in Batchcomp, then the audit can make guesses which are more accurate. This is the root cause for Batchcomp outperforming ALPHA-Batch in the simulations shown in Section 4.3.

3.4 The Batchcomp Assorters

This section converts the election assorters a_1, \dots, a_ℓ to equivalent assorters A_1, \dots, A_ℓ which depend on the accuracy of the batch-level tallies instead of their vote distribution. These new assorters, which we refer to as the *Batchcomp assorters*, are equivalent to the original ones in the sense that they all have a mean greater than $\frac{1}{2}$ iff the original ones all have a mean greater than $\frac{1}{2}$.

► **Definition 4.** For each assorter a_k , define the *Batchcomp-assorter* $A_k : C^* \rightarrow [0, \infty)$:

$$A_k(B_i) := \frac{1}{2} + \frac{M_k + a_k^{true}(B_i) - a_k^{rep}(B_i)}{2(w_k - M_k)}.$$

Where M_k is the reported margin of assorter a_k across all batches, and w_k is the maximal reported value of a_k , across all batches:

$$M_k := a_k^{rep}(B) - \frac{1}{2}, \quad w_k := \max_{j \in [d]} \{a_k^{rep}(B_j)\}.$$

As explained in Section 3.3, when the reported batch-level tallies are accurate, each Batchcomp assorter returns the same value on all batches. This is since accurate batch-level tallies indicate that for any batch B_i we have $a_k^{rep}(B_i) = a_k^{true}(B_i)$, and:

$$A_k(B_i) = \frac{1}{2} + \frac{M_k}{2(w_k - M_k)}.$$

To use these Batchcomp assorters instead of the original assorters a_1, \dots, a_ℓ , we need to show that they are non-negative and that $A_k(B) > \frac{1}{2}$ iff $a_k^{true}(B) > \frac{1}{2}$ (recall that $a(B)$ denotes the mean of an assorter a over all ballots).

▷ **Claim 5.** For any assorter a_k , its conversion to a Batchcomp assorter A_k is non-negative.

Proof. Fix an assorter a_k and its Batchcomp counterpart A_k . Examine the minimum of a_k^{true} and the maximum of a_k^{rep} . Recall that assorters are always non-negative, and that w_k is defined as the maximum of a_k^{rep} across all batches. Thus, for any batch B_i :

$$A_k(B_i) = \frac{1}{2} + \frac{M_k + \overbrace{a_k^{true}(B_i)}^{\geq 0} - \overbrace{a_k^{rep}(B_i)}^{\leq w_k}}{2(w_k - M_k)} \geq \frac{1}{2} + \frac{M_k - w_k}{2(w_k - M_k)} = 0. \quad \triangleleft$$

▷ **Claim 6.** For any assorter a_k and its conversion to a Batchcomp assorter A_k , we have $a_k^{true}(B) > \frac{1}{2}$ iff $A_k(B) > \frac{1}{2}$.

Proof. By the definition of A_k and M_k (Definition 4):

$$\begin{aligned} A_k(B) &= \frac{1}{2} + \frac{M_k + a_k^{true}(B) - a_k^{rep}(B)}{2(w_k - M_k)} \\ &= \frac{1}{2} + \frac{a_k^{rep}(B) - \frac{1}{2} + a_k^{true}(B) - a_k^{rep}(B)}{2(w_k - M_k)} \\ &= \frac{1}{2} + \frac{a_k^{true}(B) - \frac{1}{2}}{2(w_k - M_k)}. \end{aligned}$$

And since $w_k > M_k$, as $w_k \geq a_k^{rep}(B) > M_k$, this value is greater than $\frac{1}{2}$ iff $a_k^{true}(B) > \frac{1}{2}$. \triangleleft

The Batchcomp assorters A_1, \dots, A_ℓ can also be used by the ALPHA-Batch algorithm in place of the original assorters a_1, \dots, a_ℓ . This, however, does not lead to an increase in the audit's efficiency by itself, at least in the settings we simulated. Batchcomp attempts to improve on ALPHA-Batch's efficiency by combining these new assorters with the re-definition of u_k (see Section 2.3).

3.5 The Batchcomp Algorithm

1. Initialization:

- (a) Initialize $\mathcal{K} = [\ell]$, which holds the indexes of assertions we have yet to approve.
- (b) Initialize $\mathcal{B}^1 = (B_1, B_2, \dots, B_d)$ and $\mathcal{B}^0 = \emptyset$. As the algorithm progresses, \mathcal{B}^0 holds the batches which were already audited and \mathcal{B}^1 the batches that have yet to be audited.
- (c) For each $k \in \mathcal{K}$ initialize:

$$T_k := 1, \quad \mu_j := \frac{1}{2}, \quad \eta_k := \frac{1}{2} + \frac{M_k}{2(w_k - M_k)}, \quad U_k := \frac{1}{2} + \frac{M_k + \delta}{2(w_k - M_k)}.$$

For some $\delta > 0$. Appendix B examines how to choose δ . For definitions of M_k and w_k see Definition 4. Note that since $w_k > M_k > 0$ we have $U_k > \eta_k > \mu_k$.

2. Auditing Stage:

- As long as $\mathcal{B}^1 \neq \emptyset$, perform:
- (a) Sample a batch from \mathcal{B}^1 and denote it as B_i . Each batch B_j in \mathcal{B}^1 is sampled with probability proportional to its size: $\frac{|B_j|}{\sum_{B_t \in \mathcal{B}^1} |B_t|}$.
 - (b) Remove B_i from \mathcal{B}^1 and add it to \mathcal{B}^0 .
 - (c) For each $k \in \mathcal{K}$, update T_k by the same update rule as in ALPHA-Batch:

$$T_k \leftarrow T_k \left(\frac{A_k(B_i)}{\mu_k} \frac{\eta_k - \mu_k}{U_k - \mu_k} + \frac{U_k - \eta_k}{U_k - \mu_k} \right).$$

- (d) For each $k \in \mathcal{K}$, if $T_k > \frac{1}{\alpha}$, the k th assertion can be approved, so remove k from \mathcal{K} .

(e) For each $k \in \mathcal{K}$ update u_k, μ_k and η_k , in this order:

- $\mu_k \leftarrow \frac{\frac{1}{2}n - \sum_{B_j \in \mathcal{B}^0} |B_j| A_k(B_j)}{n - \sum_{B_j \in \mathcal{B}^0} |B_j|}$.
- $\eta_k \leftarrow \max \left\{ \frac{1}{2} + \frac{M_k}{2(w_k - M_k)}, \mu_k + \epsilon \right\}$.
- $U_k \leftarrow \max \{U_k, \eta_k + \epsilon\}$.

Where ϵ is some very small positive meant to ensure that $\mu_k < \eta_k < U_k$.

(f) If $\mu_k < 0$, The k th assertion is necessarily true, so remove k from \mathcal{K} .

(g) If $\mathcal{K} = \emptyset$, all assertions were approved, so approve the reported winners.

3. Output: If the audit hasn't approved after examining all batches, it can calculate the true winners of the elections.

Any initialization and update rule for the variables η_k and U_k that always maintains $\mu_k < \eta_k < U_k$ also yields a risk-limiting audit. The update rules shown here lead to increased efficiency when the batch-level tallies are accurate. η_k , the algorithm's guess for the value A_k would return on the next sampled batch, is set to the value A_k returns on each batch given that the reported batch-level tallies is accurate, as calculated in Section 3.4.

► **Theorem 7.** *Batchcomp fulfills the RLA guarantee (Definition 1).*

Proof. Batchcomp is a modified version of ALPHA-Batch, and fulfills the RLA guarantee for the same reasons as ALPHA-Batch. It makes two modifications to the ALPHA-Batch algorithm, which maintain it being risk-limiting:

1. For every $k \in [\ell]$, Batchcomp verifies that $A_k(B) > \frac{1}{2}$ while ALPHA-Batch verifies that $a_k^{true}(B) > \frac{1}{2}$. By Claim 6, verifying these two conditions is equivalent. ALPHA-Batch also relies on a_1, \dots, a_ℓ being non-negative. Switching to auditing A_1, \dots, A_ℓ requires them to be non-negative as well, which is proven in Claim 5.
2. Batchcomp uses a different initialization and update rule for U_k . While ALPHA-Batch defines U_k differently than Batchcomp, it only requires to have $U_k > \eta_k$ for every $k \in [\ell]$ for the audit to fulfill the RLA guarantee. Batchcomp's update rule for U_k and η_k (step 2e) always maintains $U_k > \eta_k$, meaning that it fulfills the guarantee as well. ◀

4 Israeli Knesset Elections RLA

This section describes how to perform an RLA to verify the results of the Israeli Knesset elections. The Knesset is the Israeli parliament and its sole legislative authority. It comprises of 120 members who are elected according to closed party-list proportional representation. The goal of this suggested Knesset RLA is to verify that each party won the correct number of seats, meaning that the correct Knesset members were elected.

This method can be used in Israel currently to verify the initial hand-count of the votes, which is not performed centrally - each polling place independently tallies its own ballots. It can also become useful if, in the future, the vote tallying will be done by some electronic means, such as an optical reader. In such cases, this method could confirm that the winners outputted by the electronic vote tabulation system are likely to be correct.

Before moving to explain the social choice function of the Knesset elections, we define some notation. Let P be the set of all parties running in the elections, and let $S := 120$ be the number of available seats. For every party $p \in P$, let $v^{true}(p)$ denote the true number of votes p received, according to the backup paper-ballots.

4.1 Knesset Election Method

Before each election cycle, each running party submits a ranked list of its candidates. On polling day, each voter votes for a single party, and parties receive seats in proportion to the share of the votes they received. The seats each party wins are given to the top-ranked candidates in the party's list. Allocating Knesset seats to the various parties is done as follows [8]:

Electoral Threshold: In the Knesset elections, only parties who receive a share of at least $t := 0.0325$ of the valid votes are eligible to win seats.

Seat allocation: The allocation of seats is done according to the D'Hondt method, a highest averages method, and can be formulated in multiple ways. We present a description of a general highest averages method which was suggested previously by Gallagher [5]. Each specific highest averages method is characterized by a unique monotonically increasing function $d : \mathbb{N} \rightarrow \mathbb{N}$ which is used during the seat allocation process. D'Hondt, the method used in the Israeli Knesset, uses $d(n) = n$. To find how many seats a party is awarded for a highest averages method with some function d :

1. Imagine a table with a row for each party which is above the threshold, and S columns. At column s in the row of party p , write $v^{true}(p)/d(s)$. All cells are initially unmarked.
2. Mark the S cells with the largest values in the table.
3. The number of marked cells a party has in its row is the number of seats it receives. Note that the values in each row are monotonically decreasing, as d is monotonically increasing, so each row would be fully marked up to a certain column, and unmarked for the rest of it.

Apparentment (Also Known as Electoral Alliances): Prior to election day, two parties may sign an apparentment agreement, which may allow one of them to gain an extra seat. If two parties sign an apparentment agreement, and only if both are above the threshold, they unite to a single allied party during the seat allocation stage. Then, the number of seats their alliance received is split between them according to the same seat allocation method (D'Hondt). If one of the parties in the apparentment is below the electoral threshold while using only its own votes, the apparentment is ignored. Each party may only sign a single apparentment agreement.

4.2 Knesset RLA Assorters

This section presents assorters that can be used to perform an RLA for the Knesset elections, using the SHANGRLA framework. We begin by presenting three conditions which all hold true iff the reported winners of the elections are correct. We then proceed to show assorters for these conditions, such that the assorters all have a mean greater than $\frac{1}{2}$ iff these conditions all hold true.

► **Theorem 8.** *Let $s^{rep}(p)$ and $s^{true}(p)$ be the reported and true number of seats that a party p won in a Knesset elections, respectively. We have it that $s^{rep}(p) = s^{true}(p)$ for every party $p \in P$, iff these 3 conditions all hold true:*

1. *Every party who is reportedly above the electoral threshold, is truly above it.*
2. *Every party who is reportedly below the electoral threshold, is truly below it.*
3. *For every two parties p_1, p_2 who are reportedly above the electoral threshold, the condition $(s^{rep}(p_1) \geq s^{true}(p_1)) \vee (s^{rep}(p_2) \leq s^{true}(p_2))$ is true.*

Proof. Fix some reported and true tallies for the elections, and calculate the number of seats each party reportedly and truly won according to these tallies. If the reported and true number of seats each party won are equal, then the 3 conditions above hold true trivially.

Otherwise, assume there is a discrepancy between the reported and true seat allocation. There must be at least one party who won more seats according to the reported results compared to the true results, which we denote as p_r , and at least one party who won less seats according to the reported results compared to the true results, which we denote as p_t .

We now show that at least one of the three conditions above are violated. If p_r is not truly above the electoral threshold, then Condition 1 is violated, as it receives seats according to the reported tally. Similarly, if p_t is below the threshold according to the reported tally, then Condition 2 is violated. Otherwise, both parties are truly above the threshold.

If both parties are reportedly above the electoral threshold, then p_t reportedly won less seats than it truly deserves, meaning that $s^{rep}(p_t) < s^{true}(p_t)$. Similarly, we have $s^{rep}(p_r) > s^{true}(p_r)$. This violates Condition 3 and concludes the proof. ◀

Above Threshold Assertion

The role of this assertion is to check that Condition 1 holds. Stark [11] has previously suggested a SHANGRLA assertion for this condition exactly. For every party p who reportedly is above the electoral threshold, we add a single SHANGRLA assorter to the set we audit:

► **Definition 9.** An above threshold assorter for a party p is defined as:

$$a_p^{above}(b) := \begin{cases} \frac{1}{2t} & \text{if } b \text{ is for party } p \\ \frac{1}{2} & \text{if } b \text{ is invalid} \\ 0 & \text{otherwise} \end{cases}$$

Below Threshold Assertion

This assertion verifies Condition 2. Confirming that a party is below the threshold is equivalent to verifying that all other parties received at least $1 - t$ of the valid votes. Therefore, we can use an assorter similar to the one above. For every party p who is reportedly below the electoral threshold, we add the following assorter to the set we audit:

► **Definition 10.** A below-threshold assorter for party p is defined as:

$$a_p^{below}(b) := \begin{cases} 0 & \text{if } b \text{ is for party } p \\ \frac{1}{2} & \text{if } b \text{ is invalid} \\ \frac{1}{2(1-t)} & \text{otherwise} \end{cases}$$

Move-Seat Assertion

This assertion is verifies that Condition 3. For any pair of parties p_1, p_2 , this essentially verifies that compared to the reported results, p_1 does not deserve extra seats at the expense of p_2 . An assertion for this condition was previously suggested by Blom et al. [1] (Section 5.2.) when auditing elections using highest averages methods. For every ordered pair of parties (p_1, p_2) who are reportedly above the electoral threshold, we add the following assorter to the set we audit:

► **Definition 11.** A move-seat assorter for two parties p_1, p_2 is defined as:

$$a_{p_1, p_2}^{move}(b) := \begin{cases} \frac{1}{2} + \frac{s^{rep}(p_1)+1}{2s^{rep}(p_2)} & \text{if } b \text{ is for } p_2 \\ 0 & \text{if } b \text{ is for } p_1 \\ \frac{1}{2} & \text{otherwise} \end{cases}$$

Handling Apparentments

The assertions above ignore the existence of apparentments. To handle them, we can simply treat each two allied parties who are reportedly above the electoral threshold as a united party when adding move-seat assertions. Additionally, to verify that the seat allocation between every two allied parties is correct, two move-seat assertions (one in each direction) are added to the audit for every two allied parties who are reportedly above the electoral threshold.

4.3 Simulations Based on Recent Elections

We describe the results of simulating the execution of a batch-level RLA over the real election results for the 24th Knesset in 2021. The partition of ballots to batches used in this simulation is done according to the real election results, and each batch contains ballots from a single polling place. The audit uses assertions as described in Section 4.2, converts their asserters to Batchcomp asserters as described in Section 3.4 and runs the Batchcomp method described in Section 3.5 to audit them.

We compare the performance of Batchcomp with the performance of the ALPHA-Batch algorithm described in section 4.2 of ALPHA [12]. ALPHA-batch uses the SHANGRLA assertions from Section 4.2 of this work, without converting their asserters to Batchcomp-asserters. For each assertion, we measure the number of ballots required to approve it by each algorithm, as a factor of the assertion’s margin (minimal number of ballots that would need to be altered, compared to the reported vote tally for the assertion to become false).

The results presented here assume that *all vote tallies are accurate*. Similar plots for results with small tabulation errors, as well as results for additional election cycles, are available in the paper’s GitHub repository. The election cycle described here is representative of the trends present in the other examined cycles.

Technical Details

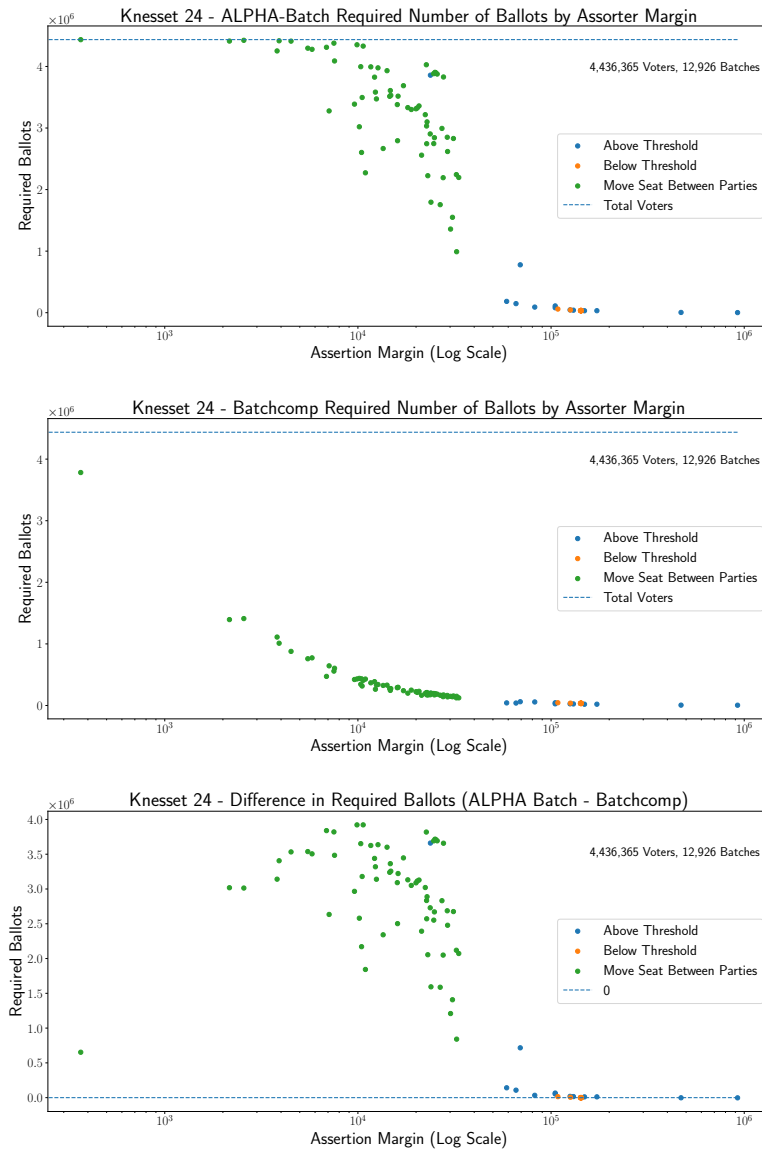
The simulated RLA uses a risk-limit of $\alpha = 0.05$ and $\delta = 10^{-10}$. The latter was determined after some experimentation - lower choices for δ do not improve efficiency when the reported results are accurate, while higher values reduce the audit’s efficiency.

The number of audited ballots by each method is averaged across 10 simulations. An examination of these simulations shows that the number of ballots required to approve each assertion by Batchcomp has very low standard deviation. The mean standard deviation across all assertions is 1,888, while the maximal standard deviation across all assertions is 5,291. The code used to generate these simulations was written in Python, and is available at the paper’s GitHub repository (see title page), along with plots for additional election cycles.

Results

Figure 1 and Table 1 show that approving the reported winners for this election cycle required auditing 85% of ballots by Batchcomp, while requiring virtually all ballots by ALPHA-Batch. If it wasn’t for a single assertion which had a very small margin (367 ballots), the Batchcomp audit would be done after auditing 32% of the ballots, while ALPHA-Batch would still require reading nearly all ballots.

The most glaring conclusion from this simulation, as well as ones we performed for additional election cycles, is that Knesset elections have very tight margins, which make them difficult to audit in a risk-limiting manner. If the election winners win with a margin



■ **Figure 1** The first two plots present the number of ballots required to approve each assertion during the audit, either by the ALPHA-Batch method or by our Batchcomp method. Each point in these plots represents a single assertion, where its value on the x axis is its margin in log-scale, and its value on the y axis is the number of ballots that the audit examined before approving the assertion. Each point in the plot is colored by the type of assertion it represents. The final plot presents the difference in ballots required per assertion between ALPHA-Batch and Batchcomp.

of below 0.01% of the total ballots, it’s unlikely that any RLA method could approve them without close to a full manual recount. Appendix D examines ways of relaxing the RLA’s guarantee to decrease the number of ballots the audit has to read.

While auditing the entire Knesset elections proves to be difficult, examining the number of ballots required to approve the various assertions shows that Batchcomp significantly outperforms ALPHA-Batch. Generally, assertions that had very small or fairly large margins required a similar number of ballots by both algorithms, while assertions with margins of

■ **Table 1** The last three assertions to be approved by the Batchcomp, including their margin and the number of ballots they required to be approved by each method.

Assertion	Margin (% of votes)	Batchcomp (% of votes)	ALPHA (% of votes)
Don't move a seat from Meretz to Labor	367 (0.008%)	3,782,269 (85%)	4,435,198 (\approx 100%)
Don't move a seat from The Joint List to Likud & Religious Zionist	2,567 (0.06%)	1,411,262 (32%)	4,424,877 (\approx 100%)
Don't move a seat from New Hope to Yamina	2,162 (0.05%)	1,394,595 (31%)	4,412,625 (99%)

between 0.01% and 2% were significantly easier to audit using Batchcomp. Some assertions which ALPHA-Batch could not approve without a nearly full manual recount were approved by Batchcomp while examining less than 20% of the backup paper-ballots.

5 The Census RLA

This section presents a risk-limiting audit method for a population census. It applies to nations which allocate political power to their constituencies or federal-states in proportion to their population according to a certain class of methods (highest averages), and who conduct a post-enumeration survey (PES) as recommended by United Nations guidelines [13]. By these guidelines, a PES is performed by randomly sampling a small number of households, re-running the census over this chosen sample, and then comparing the results to the original census. For consistency, throughout this section, we assume that this allocated political power is manifested as the number of representatives a region receives in parliament, and refer to these regions as the nation's federal-states.

The goal of our audit is to provide a clear statistical guarantee regarding the correctness of this census' resulting allocation of representatives. To achieve such a guarantee, we first need to define what allocation is considered correct. The allocation which results from the PES would not be sufficient here, since it may change based on the subset of households which were sampled. To avoid this potential issue, we view the true results of the census as the results the PES would have if it was to run over all households. This means that technically, a census RLA assumes that the PES surveyed all households. During the actual audit, however, it only asks for the information the PES collected on a small, randomly chosen sample of households, which is exactly the data that the PES actually has.

The census RLA is performed by sequentially sampling households and processing the census and PES information regarding them. Since the PES only runs over a small sample of households, the audit is limited in its length. Therefore, setting a risk-limit (probability of approving wrong results) ahead of the audit, as done in election RLAs, could be problematic. Were we to do so, then the audit might fail to approve a correct representative allocation even when using the entire PES sample, resulting in an inconclusive outcome.

The observation above leads us to slightly change the statistical guarantee that a census RLA provides: instead of setting the risk-limit and then running the audit, the census RLA runs over the entire PES and then returns the risk-limit with which it can approve the census representative allocation. If the risk-limit returned by census RLA is insufficient, a governing body may decide to conduct a second round of re-surveying, and to continue the audit on these newly re-surveyed households.

► **Definition 12.** *The census RLA guarantee:* For any $0 < \alpha \leq 1$, if running the PES over all households would lead to a different allocation of representatives than the census, then the probability that a census RLA returns a value α' such that $\alpha' \leq \alpha$ is at most α . α' is referred to as the audit's outputted risk-limit.

5.1 Post Enumeration Survey (PES)

A post enumeration survey is a process which measures the accuracy of a population census by conducting an independent population survey over a small portion of randomly chosen households. According to the guidelines published by the Department of Economic and Social Affairs of the United Nations [13], a PES begins by choosing a partial sample of the households in a nation, such that each household has an equal probability of being included in this sample. Afterwards, a new survey contacts each sampled household and asks them the exact same questions as the original census.

For our purposes, the only information of interest is the number of residents at each household. In reality, some countries may allocate representatives to federal-states according to the number of a specific sector of the population that they hold (e.g. eligible voters or citizens). In our model, we assume it is simply the number of residents, but our method applies in the same manner otherwise.

5.2 Model and Notation

In our model, a nation measures its population using a nation-wide census and then conducts a PES as described in the previous section. Denote the information given by the census as:

- H : A list of households that were surveyed.
- $g^{cen}(h)$: The number of residents a household $h \in H$ has according to the census.

And denote the information given by the PES as:

- H^{PES} : A list of households which were surveyed by the PES. Must be a subset of H .
- $g^{PES}(h)$: The number of residents a household $h \in H^{PES}$ holds according to the PES.

The nation then allocates R representatives to its federal-states, whose set we denote as \mathcal{S} , by using a *highest averages method*, as described in Section 4.1. Recall that each specific highest averages method is defined by a different monotonically increasing function $d: \mathbb{N} \rightarrow \mathbb{N}$.

Our model assumes each state also has a constant additive factor which is added to its census population count during the representative allocation process. We denote this constant as c_s for each $s \in \mathcal{S}$. Meaning, the value written at cell $[s, r]$ of the imaginary table used during the representative allocation, for $s \in \mathcal{S}$ and $r \in [R]$, is:

$$\frac{g^{cen}(s) + c_s}{d(r)}, \quad (5)$$

The additive factor, c_s , allows our model some added flexibility, meaning it can cover more political systems. In the United States, for example, we would want to exclude people living in group residence (e.g. homeless people, nursing home residents, etc') from the audit, since they are not covered by the PES. To do so, we can assume their number according to the census is accurate and run the audit over the rest of the population. This can be achieved by defining c_s to be the number of persons who live in a group residence in state s according to the census.

Our census RLA method relies on one simplifying assumption:

► **Assumption.** *In both the census and in the PES, the number of residents in a single household is upper-bounded by a known value, denoted as g^{max} .*

The value g^{max} must be set before the PES is conducted. Both the census and the PES must report that all households have g^{max} residents at most.

This assumption is necessary due to a critical difference between elections and censuses; In elections, a single ballot has very limited power. In a census, if it was not for this assumption, a single household could hold an arbitrarily large number of residents and completely swing the allocation of representatives to the states.

Finally, denote the number of representatives awarded to state $s \in \mathcal{S}$ according to the census as $r^{cen}(s)$.

5.3 Census RLA Overview

The following sections suggest a new method for census RLAs, which relies on the SHANGRLA framework. In the following section, we design SHANGRLA assertions for auditing the census' resulting allocation of representatives to the federal-states. While these assertions can be used as-is to perform a census RLA, they are only an intermediate step in the development of more efficient assertions. These more efficient assertions are used by a modified version of the ALPHA Martingale Test to perform a census RLA, as described in Section 5.5.

5.4 Census RLA Assorters

We begin by adapting the definition of assertions and assorters to the language of census RLAs. When auditing elections, an assorter is defined as a non-negative function over the set of possible ballots a voter may cast. When auditing a census, we define an assorter as a non-negative function over the set of all households, meaning $a: H \rightarrow [0, \infty)$. An assorter a satisfies the assertion $\frac{1}{|H|} \sum_{h \in H} a(h) > \frac{1}{2}$ iff some condition regarding the allocation of representatives to the federal states is true.

► **Definition 13.** *Census assorters are functions $a_1, \dots, a_\ell: H \rightarrow [0, \infty)$ with the following property: Given some census results, if the PES surveyed all households, the allocation of representatives according to the census and according to the PES match iff for all $k \in [\ell]$:*

$$\frac{1}{|H|} \sum_{h \in H} a_k(h) > \frac{1}{2}. \quad (6)$$

These ℓ inequalities are referred to as the **census assertions**.

The census assorters for our setting are developed by finding a set linear inequalities which all hold true iff a full PES leads to the same allocation of representatives as the census. These inequalities are then converted to SHANGRLA style assertions by the method described by Blom et al. [1] (see Section 2.1).

► **Theorem 14.** *Assume the PES surveyed all households. The allocation of representatives according to the census and according to the PES match, iff for any two states $s_1, s_2 \in \mathcal{S}$:*

$$\frac{\sum_{h \in H} g_{s_1}^{PES}(h) + c_{s_1}}{d(r^{cen}(s_1))} > \frac{\sum_{h \in H} g_{s_2}^{PES}(h) + c_{s_2}}{d(r^{cen}(s_2) + 1)}. \quad (7)$$

The proof of this theorem appears in Appendix A.1. By the method suggested by Blom et al. [1], confirming Equation (7) is equivalent to confirming the SHANGRLA style assertion:

$$\frac{1}{|H|} \sum_{h \in H} a_{s_1, s_2}^{PES}(h) > \frac{1}{2},$$

where:

► **Definition 15.** The census assorter a_{s_1, s_2}^{PES} is defined as:

$$a_{s_1, s_2}^{PES}(h) := \frac{g_{s_1}^{PES}(h)}{cd(r^{cen}(s_1))} + \frac{g^{max} - g_{s_2}^{PES}(h)}{cd(r^{cen}(s_2) + 1)},$$

where c denotes:

$$c := 2 \left(\frac{g^{max}}{d(r^{cen}(s_2) + 1)} - \frac{c_{s_1}}{|H|d(r^{cen}(s_1))} + \frac{c_{s_2}}{|H|d(r^{cen}(s_2) + 1)} \right).$$

► **Theorem 16.** Assume that the PES surveyed all households. The assorters $\{a_{s_1, s_2}^{PES} \mid s_1, s_2 \in \mathcal{S}, s_1 \neq s_2\}$ are all non-negative and satisfy the following condition: The allocation of representatives according to the census and the PES match iff for all $s_1, s_2 \in \mathcal{S}$:

$$\frac{1}{|H|} \sum_{h \in H} a_{s_1, s_2}^{PES}(h) > \frac{1}{2}. \quad (8)$$

Proof. The non-negativity of these assorters is due to the method by Blom et al., which generates non-negative assorters. Additionally, by this method, for any two states s_1, s_2 , verifying (8) is equivalent to verifying (7). By Theorem 14, verifying (7) for every two states is equivalent to verifying that the full PES leads to the same representative allocation as the census, concluding this proof. ◀

For each assorter a_{s_1, s_2}^{PES} , we now define a new assorter A_{s_1, s_2} which can also be used to audit the same census. A_{s_1, s_2} has a significant advantage over a_{s_1, s_2}^{PES} , which motivates us to use it instead. Each assorter a_{s_1, s_2}^{PES} essentially audits the number of residents per household according to the PES, without using the per-household census data. Meanwhile, A_{s_1, s_2} audits the discrepancy in the number of household members between the census and the PES. Since we typically expect this discrepancy to be small, this yields a more stable and efficient audit.

Before defining A_{s_1, s_2} , note that each assorter a_{s_1, s_2}^{PES} can also be defined over the census population counts instead of the PES counts. We denote this as a_{s_1, s_2}^{cen} :

► **Definition 17.** The value of an assorter a_{s_1, s_2}^{PES} as in Definition 15 over the census population count is defined as:

$$a_{s_1, s_2}^{cen}(h) := \frac{g_{s_1}^{cen}(h)}{cd(r^{cen}(s_1))} + \frac{g^{max} - g_{s_2}^{cen}(h)}{cd(r^{cen}(s_2) + 1)}.$$

Using this reported (by the census) and true (by the PES) resident counts, we define new assorters which audit the discrepancy between them. This is similar to the Batchcomp assorters from Section 3.4, which audit the batch-level discrepancy between the reported and true vote tallies.

► **Definition 18.** The census comparison assorter A_{s_1, s_2} for states $s_1, s_2 \in \mathcal{S}$ is defined as:

$$A_{s_1, s_2}(h) := \frac{1}{2} + \frac{m_{s_1, s_2} + a_{s_1, s_2}^{PES}(h) - a_{s_1, s_2}^{cen}(h)}{2(z_{s_1, s_2} - m_{s_1, s_2})},$$

where m_{s_1, s_2} is the margin of a_{s_1, s_2} according to the census population counts:

$$m_{s_1, s_2} := \frac{1}{|H|} \sum_{h \in H} a_{s_1, s_2}^{cen}(h) - \frac{1}{2},$$

and:

$$z_{s_1, s_2} := \max \left\{ \frac{g^{max}}{cd(r^{cen}(s_2) + 1)}, \frac{g^{max}}{cd(r^{cen}(s_1))}, 0 \right\}.$$

► **Theorem 19.** Assume that the PES surveyed all households. The assertors $\{A_{s_1, s_2} \mid s_1, s_2 \in \mathcal{S}\}$, as defined in Definition 18, are all non-negative and satisfy the following condition: the allocation of representatives according to the census and the PES match iff for all $s_1, s_2 \in \mathcal{S}$:

$$\frac{1}{|H|} \sum_{h \in H} A_{s_1, s_2}(h) > \frac{1}{2}.$$

The proof for this theorem is presented in Appendix A.2.

5.5 Census RLA Algorithm

The algorithm presented next is a slightly altered version of the ALPHA Martingale Test, when thinking of each household as a ballot whose content is the household's state and its number of residents. We denote the households surveyed by the PES as $H^{PES} = (h_1, h_2, \dots, h_d)$ for some $d \in \mathbb{N}$, and assume that they are given in random order.

1. Initialization

(a) For each $(s_1, s_2) \in \mathcal{S} \times \mathcal{S}$ s.t. $s_1 \neq s_2$, initialize:

$$\begin{aligned} & \bullet T_{s_1, s_2} := 1. \\ & \bullet T_{s_1, s_2}^{max} := 1. \\ & \bullet \mu_{s_1, s_2} := \frac{1}{2}. \\ & \bullet \eta_{s_1, s_2} := \frac{1}{2} + \frac{m_{s_1, s_2}}{2(z_{s_1, s_2} - m_{s_1, s_2})}. \\ & \bullet U_{s_1, s_2} := \frac{1}{2} + \frac{m_{s_1, s_2} + \delta}{2(z_{s_1, s_2} - m_{s_1, s_2})}, \text{ where } \delta > 0. \end{aligned}$$

2. **Auditing Stage:** Iterate over the households in H^{PES} , for the i th household h_i perform for each ordered pair of states (s_1, s_2) :

(a) Update T_{s_1, s_2} and T_{s_1, s_2}^{max} :

$$\begin{aligned} & \bullet T_{s_1, s_2} \leftarrow T_{s_1, s_2} \left(\frac{A_{s_1, s_2}(h_i)}{\mu_{s_1, s_2}} \frac{\eta_{s_1, s_2} - \mu_{s_1, s_2}}{U_{s_1, s_2} - \mu_{s_1, s_2}} + \frac{U_{s_1, s_2} - \eta_{s_1, s_2}}{U_{s_1, s_2} - \mu_{s_1, s_2}} \right). \\ & \bullet T_{s_1, s_2}^{max} \leftarrow \max \{ T_{s_1, s_2}^{max}, T_{s_1, s_2} \}. \end{aligned}$$

(b) Update μ_{s_1, s_2} , η_{s_1, s_2} and U_{s_1, s_2} , in this order:

$$\begin{aligned} & \bullet \mu_{s_1, s_2} \leftarrow \frac{\frac{1}{2}|H| - \sum_{j=1}^i A_{s_1, s_2}(h_j)}{|H| - i}. \\ & \bullet \eta_{s_1, s_2} \leftarrow \max \left\{ \frac{1}{2} + \frac{m_{s_1, s_2}}{2(z_{s_1, s_2} - m_{s_1, s_2})}, \mu_{s_1, s_2} + \epsilon \right\}. \\ & \bullet U_{s_1, s_2} \leftarrow \max \{ U_{s_1, s_2}, \eta_{s_1, s_2} + \epsilon \}. \end{aligned}$$

Where ϵ is some very small positive meant to ensure that $\mu_{s_1, s_2} < \eta_{s_1, s_2} < U_{s_1, s_2}$.

(c) For each s_1, s_2 , if $\mu_{s_1, s_2} < 0$, the corresponding assertion must be true, so set $T_{s_1, s_2}^{max} = \infty$.

3. **Output:** The result of the audit is the maximal risk-limit across all assertions:

$$\max_{s_1, s_2 \in \mathcal{S}} \left\{ \frac{1}{T_{s_1, s_2}^{max}} \right\}.$$

► **Theorem 20.** The census RLA fulfills the census RLA guarantee (Definition 12).

Proof. The census RLA is essentially the ALPHA Martingale Test, with four modifications. We explain why these modifications maintain the risk-limiting nature of the ALPHA Martingale Test:

- Instead of sampling and examining ballots, the census RLA samples and examines households. This does not effect the fact that the ALPHA Martingale Test is risk-limiting - a census RLA can be viewed as a classical election RLA where every ballot correspond to a household, and holds that household's state and number of residents.

- The census RLA doesn't sample households at random, it iterates over the households sampled by the PES. Despite this, since the PES surveys randomly selected households, the algorithm audits a previously unsampled household selected uniformly at random in each iteration. This is just as the ALPHA Martingale Test requires.
- Instead of pre-setting the risk-limit, the risk-limit with which the census representative allocation can be approved is outputted after iterating over all PES households. This outputted risk-limit is already available as part of the ALPHA Martingale Test. In election RLAs, the audit approves the reported winners when this running risk-limit drops below the pre-set risk-limit. Here it outputs it after examining all PES households.
- The census RLA defines U_{s_1, s_2} (which corresponds to u_k in the ALPHA Martingale Test) differently. As mentioned previously, it always maintains $U_{s_1, s_2} > \eta_{s_1, s_2}$, so the audit remains risk-limiting. ◀

5.6 Simulations

This section simulates the suggested census RLA on the Cypriot census and its resulting allocation of representatives to districts in the House of Representatives of Cyprus. Our original intention was to simulate the suggested census RLA method on the US census and its resulting allocation of representatives in the US House of Representatives to the states. This turned out to be infeasible, however, as the audit outputted an insufficient risk-limit. This is a result of the relatively large number of states (50) and representatives (435) in the American system. Allocating many representatives to many states increases the probability of there being a single representative whose allocation is determined by a very small number of state residents.

To show that the census RLA is useful in other settings, we chose to simulate the audit on the House of representatives of Cyprus, where 56 representatives are allocated to 5 districts. This should be viewed as a pet-setting for testing the census RLA method, and not as a ready-as-is implementation for the Cypriot system.

The House of Representatives of Cyprus

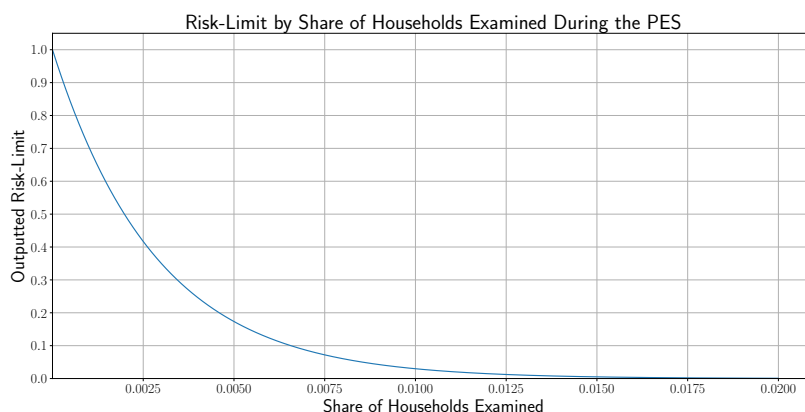
The House of representatives of Cyprus is its sole legislating body, and holds 56 occupied seats. An additional 24 seats are reserved for the Turkish Cypriot community, who withdrew from the political decision-making process in 1964, leaving their house seats vacant [4].

The remaining 56 seats of the house are allocated to 5 districts. Currently, the allocation of seats to the districts is amended by law when found necessary, and does not change automatically following a census according to a set method. A census RLA could be useful when performing these amendments, to ensure that the resulting allocation of seats to districts is sufficiently reliable.

Technical Details

We allocated representatives to districts using the D'Hondt method. D'Hondt was chosen since it's currently used in the Cypriot elections to allocate seats to political parties. The audit was run assuming that each household holds 15 residents at most, and with $\delta = 10^{-10}$.

The census data used in the simulation is based on the results of the 2021 Cypriot census. For more details regarding the census data generation, see Appendix C. The simulation's code is available at the paper's GitHub repository (see title page).



■ **Figure 2** The census RLA output when the census and PES fully agree on the number of residents in each household, as a factor of the share of households that were surveyed by the PES.

Results

To examine the census RLA method, we present in Figure 2 the outputted risk-limit of the census RLA as a factor of the size of the PES. This simulation assumes that the census and the PES agree on the number of residents in each PES-surveyed households. Results with small census and PES disagreement, which are largely similar to the ones presented here, are available at the paper’s GitHub repository.

Under the specified conditions, a PES which samples 0.66% of households is sufficient for a risk limit of 0.1, and a sample of 0.85% is sufficient for a risk-limit of 0.05. A PES often surveys around 1% of households [7], meaning that our census RLA can confidently approve the census’ resulting allocation of representatives to districts under these conditions.

These results show that the census RLA method is applicable in some settings, when the number of representatives and federal-states to allocate them to is relatively small. When there are many representatives and federal-states, even a small error in the census can lead to a wrongful allocation of representatives, and auditing the census results requires a larger PES sample.

6 Discussion and Further Research

Throughout this work, we can observe that an election’s social choice function and setting can severely limit the efficiency of their RLAs. Systems like the Israeli Knesset elections and the US House of Representatives’ allocation of representatives to states are very sensitive to enumeration errors, making it difficult to audit them efficiently.

The simulation of the Batchcomp method on the Israeli Knesset elections (Section 4.3) indicates that Batchcomp provides a noticeable improvement over ALPHA-Batch in the limited settings that were tested. Despite this relative success, we cannot definitively say it outperforms existing methods without a clear, rigorous way of analyzing their efficiency.

The census RLA method appears to be useful in some limited settings, and can be implemented using existing post-enumeration surveys. In systems where our method is currently not sufficient, a census RLA could perhaps aim for a weaker guarantee - that the number of representatives each state should receive according to the PES is close to the number it has according to the census. This option is discussed in Appendix D.

The work raises many open questions and potential research directions:

Applying RLAs in Additional Settings: Generally speaking, RLAs can be applied whenever one wishes to verify the computation of some function over a large number of inputs obtained through potentially error-prone processes. While political elections provide a natural environment for their application, we advocate for their use in a wider range of settings to ensure reliable results.

As an example of such settings, RLAs could potentially be used to verify that decisions taken based on datasets which were altered in order to satisfy differential privacy are correct according to the real data. This could be achieved by running an RLA in a protected environment (enclave) which holds a subsample of the original, noiseless data. In this setting, the noisy, (differential private) dataset is seen as the reported result, while the noiseless dataset is the true results. An RLA can verify that the results of some computation over the differential private dataset and over the original noiseless dataset are likely to be identical, based on a (hopefully) small random sample from the original dataset. One challenge is to make sure that the very fact that the data passed the test does not hurt the desired differential privacy property.

Analytical Analysis of the Efficiency of RLAs: Most recent literature in the field, including this work, focuses on suggesting new RLA algorithms and fitting them to additional electoral systems and settings. There is little to no analytical analysis of the *efficiency* and capabilities of many RLA methods. Without a more rigorous analysis, it is not possible to definitively determine which RLA methods are better for which settings. Such analysis could help, for instance, to argue analytically whether Batchcomp is indeed preferable over ALPHA-Batch.

Connection Between RLAs and Computational Models: Thus far, advancements in the field of RLAs were done mostly independently and without connection to computational models. Finding such connections may inspire new RLA algorithms, or suggest new methods for analyzing the capabilities and efficiency of existing methods. As an example of these connections, RLAs can essentially be viewed as randomized decision trees, where each branch represents a different sequence of paper-ballots that can be uncovered during the audit. Viewing RLAs in this manner may allow us to analyze their query complexity (number of ballots examined) or instance complexity (best possible performance over specific election results) and to apply existing results from other fields onto RLAs.

As a potential example for this, viewing RLAs as randomized decision trees may allow us to find lower bounds for the query-complexity of RLAs by analyzing the randomized unlabeled certificate complexity of the social choice function they operate on, as defined by Grossman, Komargodski and Naor [6]. Randomized unlabeled certificate complexity is a complexity measure of a function over some specific input. It's defined roughly as the minimal number of queries, in expectation, that any randomized decision tree which computes this function has to perform over the specified input, given a permuted version of it as a certificate. This notion could be relevant for RLAs since they are essentially randomized decision trees which calculate a social choice function's output (the true winners) while using the reported election results as a certificate.

References

- 1 Michelle Blom, Jurlind Budurushi, Ronald L. Rivest, Philip B. Stark, Peter J. Stuckey, Vanessa Teague, and Damjan Vukcevic. Assertion-based approaches to auditing complex elections, with application to party-list proportional elections. In *International Joint Conference on Electronic Voting*, pages 47–62. Springer, 2021.

- 2 Michelle Blom, Philip B Stark, Peter J Stuckey, Vanessa Teague, and Damjan Vukcevic. Auditing hamiltonian elections. In *Financial Cryptography and Data Security. FC 2021 International Workshops: CoDecFin, DeFi, VOTING, and WTSC, Virtual Event, March 5, 2021, Revised Selected Papers*, pages 235–250. Springer, 2021.
- 3 United States Census Bureau. Historical households tables. <https://www.census.gov/data/tables/time-series/demo/families/households.html>, 2022. Table HH-4.
- 4 Giorgos Charalambous. The house of representatives. *The Politics and Government of Cyprus. Oxford: Peter Lange*, pages 143–168, 2008.
- 5 Michael Gallagher. Proportionality, disproportionality and electoral systems. *Electoral studies*, 10(1):33–51, 1991.
- 6 Tomer Grossman, Ilan Komargodski, and Moni Naor. Instance complexity and unlabeled certificates in the decision tree model. In *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.
- 7 Guihua Hu, Ting Wen, and Yuhuan Liu. Determining the sample size of a post-enumeration survey: The case of china, 2020. *Mathematical Population Studies*, pages 1–31, 2022.
- 8 The Knesset Lexicon. The distribution of knesset seats (bader-ofer method). <https://m.knesset.gov.il/en/about/lexicon/pages/seats.aspx>. This page wrongly states that the electoral threshold is 2.0%. It was since changed to 3.25%.
- 9 Mark Lindeman and Philip B. Stark. A gentle introduction to risk-limiting audits. *IEEE Security & Privacy*, 10(5):42–49, 2012.
- 10 Statistical Service of the Republic of Cyprus. Census of population and housing 2021: Preliminary results. <https://www.pio.gov.cy/en/press-releases-article.html?id=27965>, 2022.
- 11 Philip B. Stark. Sets of half-average nulls generate risk-limiting audits: Shangrila. In *International Conference on Financial Cryptography and Data Security*, pages 319–336. Springer, 2020.
- 12 Philip B. Stark. Alpha: Audit that learns from previously hand-audited ballots. *The Annals of Applied Statistics*, 17(1):641–679, 2023.
- 13 Department of Economics United Nations Secretariat and Statistics Division Social Affairs. Post enumeration surveys operational guidelines. https://unstats.un.org/unsd/demographic/standmeth/handbooks/Manual_PESen.pdf, 2010.
- 14 Ian Waudby-Smith, Philip B. Stark, and Aaditya Ramdas. Rilacs: Risk limiting audits via confidence sequences. In *International Joint Conference on Electronic Voting*, pages 124–139. Springer, 2021.

A Proofs

A.1 Proof of Theorem 14

► **Theorem 14.** *Assume the PES surveyed all households. The allocation of representatives according to the census and according to the PES match, iff for any two states $s_1, s_2 \in \mathcal{S}$:*

$$\frac{\sum_{h \in H} g_{s_1}^{PES}(h) + c_{s_1}}{d(r^{cen}(s_1))} > \frac{\sum_{h \in H} g_{s_2}^{PES}(h) + c_{s_2}}{d(r^{cen}(s_2) + 1)}. \quad (7)$$

Proof. First, assume that the two allocations of representatives match. Examine the imaginary table with which representatives are allocated to states according to the PES. Recall that each state has exactly its first $r^{PES}(s)$ cells marked. Since we assume that for any $s \in \mathcal{S}$, $r^{PES}(s) = r^{cen}(s)$, we have it that for any $s_1, s_2 \in \mathcal{S}$, the cell at index $[s_1, r^{cen}(s_1)]$ is marked, while the cell at $[s_2, r^{cen}(s_2) + 1]$ is not. Since the marked cells are the ones which hold the largest values in the table, the cell at $[s_1, r^{cen}(s_1)]$ has a larger value than the cell at $[s_2, r^{cen}(s_2) + 1]$. Writing these values out results exactly in (7)- the larger term is the value at $[s_2, r^{cen}(s_2) + 1]$, and the smaller is the value at $[s_1, r^{cen}(s_1)]$.

Towards proving the other direction of the equivalence, we show that if (7) is true for any $s_1, s_2 \in \mathcal{S}$, then a certain condition (9) holds for any s_1, s_2 . We then show that if this condition is true, then the allocation of representatives according to the census and according to the PES match.

▷ **Claim 21.** Let $r^{PES}(s)$ be the number of representatives a state s is allocated according to the full PES results. For any $s_1, s_2 \in \mathcal{S}$, if (7) is true then:

$$(r^{PES}(s_1) \geq r^{cen}(s_1)) \vee (r^{PES}(s_2) \leq r^{cen}(s_2)). \quad (9)$$

Proof. Assume towards contradiction that for some $s_1, s_2 \in \mathcal{S}$, the condition (9) is false, meaning that $(r^{PES}(s_1) < r^{cen}(s_1)) \wedge (r^{PES}(s_2) > r^{cen}(s_2))$ is true.

Examine the table used to allocate representatives to states according to the PES results. According to this table, s_2 is awarded $r^{PES}(s_2)$ representatives. Since $r^{PES}(s_2) > r^{cen}(s_2)$, and since the row s_2 has exactly its first $r^{PES}(s_2)$ cells marked, the cell at $[s_2, r^{cen}(s_2) + 1]$ is marked. Additionally, since s_1 was awarded exactly $r^{PES}(s_1)$ seats and since $r^{PES}(s_1) < r^{cen}(s_1)$, the cell at $[s_1, r^{cen}(s_1)]$ is not marked.

By the paragraph above, if $(r^{PES}(s_1) \geq r^{cen}(s_1)) \vee (r^{PES}(s_2) \leq r^{cen}(s_2))$ is false, then the cell at $[s_2, r^{cen}(s_2) + 1]$ is marked while the cell at $[s_1, r^{cen}(s_1)]$ is not. Since the marked cells are the ones which hold the largest values, it follows that the cell at $[s_2, r^{cen}(s_2) + 1]$ has a larger value than the cell at $[s_1, r^{cen}(s_1)]$, meaning that:

$$\frac{\sum_{h \in H} g_{s_1}^{PES}(h) + c_{s_1}}{d(r^{cen}(s_1))} \leq \frac{\sum_{h \in H} g_{s_2}^{PES}(h) + c_{s_2}}{d(r^{cen}(s_2) + 1)}.$$

The larger term in this inequality is the value at index $[s_2, r^{cen}(s_2) + 1]$ and the smaller one is the value at index $[s_1, r^{cen}(s_1)]$. This contradicts (7), and thereby proves this claim. ◁

▷ **Claim 22.** If (9) is true for any $s_1, s_2 \in \mathcal{S}$, then the allocation of representatives according to the census and according to the full PES are identical.

Proof. Assume towards contradiction that the two allocations are not identical. Therefore, there must be at least one state s with $r^{PES}(s) \neq r^{cen}(s)$. If $r^{PES}(s) > r^{cen}(s)$, since the number of total representatives is constant, there must be another state s' with $r^{PES}(s') < r^{cen}(s')$. Similarly, if $r^{PES}(s) < r^{cen}(s)$, there must be another state s' with $r^{PES}(s') > r^{cen}(s')$. Either way, (9) is false. Thus, if (9) is true for every pair of states, then the two allocations must be identical, completing the proof. ◁

Using these two claims, we can now complete the proof of this theorem. Assume (7) is true for any pair of states. By Claim 21, (9) is also true for any pair of states, and by Claim 22, this makes the allocation of representatives according to the census and according to the PES identical. This proves the other direction of the equivalence and concludes the proof. ◀

A.2 Proof of Theorem 19

► **Theorem 19.** Assume that the PES surveyed all households. The assorters $\{A_{s_1, s_2} \mid s_1, s_2 \in \mathcal{S}\}$, as defined in Definition 18, are all non-negative and satisfy the following condition: the allocation of representatives according to the census and the PES match iff for all $s_1, s_2 \in \mathcal{S}$:

$$\frac{1}{|H|} \sum_{h \in H} A_{s_1, s_2}(h) > \frac{1}{2}.$$

Proof. We show that A_{s_1, s_2} is non-negative and that the required equivalence holds.

▷ Claim 23. For any $s_1, s_2 \in \mathcal{S}$, A_{s_1, s_2} is non-negative.

Proof. Fix two states $s_1, s_2 \in \mathcal{S}$. Recall the definition of A_{s_1, s_2} :

$$A_{s_1, s_2}(h) := \frac{1}{2} + \frac{m_{s_1, s_2} + a_{s_1, s_2}^{PES}(h) - a_{s_1, s_2}^{cen}(h)}{2(z_{s_1, s_2} - m_{s_1, s_2})}.$$

By the definition of a_{s_1, s_2}^{PES} and a_{s_1, s_2}^{cen} , the value of the nominator in the expression above is:

$$m_{s_1, s_2} + a_{s_1, s_2}^{PES}(h) - a_{s_1, s_2}^{cen}(h) = m_{s_1, s_2} + \frac{g_{s_1}^{PES}(h) - g_{s_1}^{cen}(h)}{cd(r^{cen}(s_1))} + \frac{g_{s_2}^{cen}(h) - g_{s_2}^{PES}(h)}{cd(r^{cen}(s_2) + 1)}.$$

h is either from s_1 , from s_2 or from neither of them. If it's from neither, this expression equals m_{s_1, s_2} . If it's from s_1 , then:

$$m_{s_1, s_2} + \frac{\overbrace{g_{s_1}^{PES}(h) - g_{s_1}^{cen}(h)}^{\geq 0}}{cd(r^{cen}(s_1))} + \frac{\overbrace{g_{s_2}^{cen}(h) - g_{s_2}^{PES}(h)}^{=0}}{cd(r^{cen}(s_2) + 1)} \geq m_{s_1, s_2} - \frac{g^{max}}{cd(r^{cen}(s_1))}$$

where g^{max} is the maximal number of residents a single household may have. If h is from s_2 , then:

$$m_{s_1, s_2} + \frac{\overbrace{g_{s_1}^{PES}(h) - g_{s_1}^{cen}(h)}^{=0}}{cd(r^{cen}(s_1))} + \frac{\overbrace{g_{s_2}^{cen}(h) - g_{s_2}^{PES}(h)}^{\geq 0}}{cd(r^{cen}(s_2) + 1)} \geq m_{s_1, s_2} - \frac{g^{max}}{cd(r^{cen}(s_2) + 1)}.$$

So for any $h \in H$:

$$\begin{aligned} & m_{s_1, s_2} + a_{s_1, s_2}^{PES}(h) - a_{s_1, s_2}^{cen}(h) \\ & \geq \min \left\{ m_{s_1, s_2}, m_{s_1, s_2} - \frac{g^{max}}{cd(r^{cen}(s_2) + 1)}, m_{s_1, s_2} - \frac{g^{max}}{cd(r^{cen}(s_1))} \right\}. \end{aligned} \quad (10)$$

By (10) and by the definition of z_{s_1, s_2} (Definition 18):

$$m_{s_1, s_2} + a_{s_1, s_2}^{PES}(h) - a_{s_1, s_2}^{cen}(h) \geq m_{s_1, s_2} - z_{s_1, s_2}. \quad (11)$$

Meaning that for any $h \in H$:

$$A_{s_1, s_2}(h) = \frac{1}{2} + \frac{m_{s_1, s_2} + a_{s_1, s_2}^{PES}(h) - a_{s_1, s_2}^{cen}(h)}{2(z_{s_1, s_2} - m_{s_1, s_2})} \geq \frac{1}{2} + \frac{m_{s_1, s_2} - z_{s_1, s_2}}{2(z_{s_1, s_2} - m_{s_1, s_2})} = 0,$$

proving the claim. \triangleleft

▷ Claim 24. Assume the PES surveyed all households. The allocation of representatives according to the census and the PES match iff for all $s_1, s_2 \in \mathcal{S}$:

$$\frac{1}{|H|} \sum_{h \in H} A_{s_1, s_2}(h) > \frac{1}{2}.$$

Proof. By Theorem 16, the allocation of representatives according to the census and the PES match iff for all $s_1, s_2 \in \mathcal{S}$:

$$\frac{1}{|H|} \sum_{h \in H} a_{s_1, s_2}^{PES}(h) > \frac{1}{2}.$$

Therefore, to prove this claim, it suffices to prove that for every $s_1, s_2 \in \mathcal{S}$:

$$\left(\frac{1}{|H|} \sum_{h \in H} A_{s_1, s_2}(h) > \frac{1}{2} \right) \iff \left(\frac{1}{|H|} \sum_{h \in H} a_{s_1, s_2}^{PES}(h) > \frac{1}{2} \right).$$

Fix any two federal-states $s_1, s_2 \in \mathcal{S}$. We show that the two inequalities above are equivalent:

$$\begin{aligned} & \frac{1}{|H|} \sum_{h \in H} A_{s_1, s_2}(h) > \frac{1}{2} \\ \iff & \frac{1}{|H|} \sum_{h \in H} \left(\frac{1}{2} + \frac{m_{s_1, s_2} + a_{s_1, s_2}^{PES}(h) - a_{s_1, s_2}^{cen}(h)}{2(z_{s_1, s_2} - m_{s_1, s_2})} \right) > \frac{1}{2} \\ \iff & \frac{1}{|H|} \sum_{h \in H} \frac{m_{s_1, s_2} + a_{s_1, s_2}^{PES}(h) - a_{s_1, s_2}^{cen}(h)}{2(z_{s_1, s_2} - m_{s_1, s_2})} > 0. \end{aligned}$$

Now, using the definition of m_{s_1, s_2} and re-arranging the summation yields the desired equivalence:

$$\begin{aligned} \iff & \frac{1}{|H|} \sum_{h \in H} \frac{\frac{1}{|H|} \sum_{h' \in H} a_{s_1, s_2}^{cen}(h') - \frac{1}{2} + a_{s_1, s_2}^{PES}(h) - a_{s_1, s_2}^{cen}(h)}{2(z_{s_1, s_2} - m_{s_1, s_2})} > 0 \\ \iff & \frac{\frac{1}{|H|} \sum_{h \in H} a_{s_1, s_2}^{PES}(h) - \frac{1}{2}}{2(z_{s_1, s_2} - m_{s_1, s_2})} > 0 \\ \iff & \frac{1}{|H|} \sum_{h \in H} a_{s_1, s_2}^{PES}(h) > \frac{1}{2}. \end{aligned}$$

The last transition relies on the fact that $z_{s_1, s_2} > m_{s_1, s_2}$, which is true since $z_{s_1, s_2} \geq \max_{h \in H} a_{s_1, s_2}^{cen}(h) \geq m_{s_1, s_2}$ (see Definition 18). This concludes the proof of this claim. \triangleleft

The combination of these two claims completes this theorem's proof. \blacktriangleleft

B Batchcomp – Choosing δ

As seen in Section 3.5, for every assorter a_k and its Batchcomp counterpart A_k we initialize:

$$U_k = \frac{1}{2} + \frac{M_k + \delta}{2(w_k - M_k)},$$

where $\delta > 0$. Different choices for δ all maintain the RLA guarantee, but under certain conditions, certain values of δ yield more efficient audits. This section attempts to give intuition regarding the ideal choice of δ . Generally, the more we expect the reported vote tallies of the different batches to be accurate, the smaller δ should be. We show this by comparing μ_k to the expected value of a Batchcomp assorter on the next batch to be sampled.

\triangleright **Claim 25.** During a Batchcomp RLA, if the next sampled batch B_i satisfies $A_k(B_i) \geq \mu_k$ for some batch-assorter A_k , then choosing a smaller U_k increases the audit's efficiency, and vice-versa; if $A_k(B_i) < \mu_k$, then setting a larger U_k increases the audit's efficiency.

Proof. Examine some Batchcomp assorter A_k . Approving its assertion requires fewer ballots the more significantly T_k grows per batch. This is because the audit approves assertion k when $T_k > \frac{1}{\alpha}$. Therefore, it suffices to show that if $A_k(B_i) \geq \mu_k$, then T_k grows more significantly when U_k is small, and vice-versa.

Towards this purpose, denote the next audited batch as B_i . To prove this claim, we take the derivative by U_k of the update rule of T_k in step 2c of the Batchcomp algorithm:

$$T_k \leftarrow T_k \left(\frac{A_k(B_i)}{\mu_k} \frac{\eta_k - \mu_k}{U_k - \mu_k} + \frac{U_k - \eta_k}{U_k - \mu_k} \right).$$

Taking its derivative by U_k results in:

$$\begin{aligned} & T_k \left(-\frac{A_k(B_i)}{\mu_k} \frac{\eta_k - \mu_k}{(U_k - \mu_k)^2} + \frac{1}{U_k - \mu_k} - \frac{U_k - \eta_k}{(U_k - \mu_k)^2} \right) \\ &= \frac{T_k}{(U_k - \mu_k)^2} \left(-\frac{A_k(B_i)}{\mu_k} (\eta_k - \mu_k) + U_k - \mu_k - U_k + \eta_k \right) \\ &= \frac{T_k}{(U_k - \mu_k)^2} \left(-\frac{A_k(B_i)}{\mu_k} (\eta_k - \mu_k) - \mu_k + \eta_k \right) \\ &= \underbrace{T_k \frac{\eta_k - \mu_k}{(U_k - \mu_k)^2}}_{>0} \left(1 - \frac{A_k(B_i)}{\mu_k} \right). \end{aligned}$$

Where the term above the under-brace is positive since T_k is positive, and since we always have $U_k > \eta_k > \mu_k \geq 0$. We can observe that if $A_k(B_i) > \mu_k$, this derivative is negative, meaning that choosing a smaller value for U_k causes T_k to increase more significantly. If $A_k(B_i) < \mu_k$, then the opposite is true. This concludes the proof of this claim. \triangleleft

According to this claim, if we expect to have $A_k(B_i) > \mu_k$ for all batch-assorters and batches, we should choose a smaller δ , and vice versa. When using a Batchcomp assorter, we have:

$$A_k(B_i) = \frac{1}{2} + \frac{M_k + a_k^{true}(B_i) - a_k^{rep}(B_i)}{2(w_k - M_k)}.$$

And $w_k > M_k > 0$ by the definition of M_k . Therefore, as long as the batch-level discrepancies between the reported and true vote counts are small, we expect to consistently have $A_k(B_i) \geq \mu_k$, meaning we should choose a smaller δ . To get $A_k(B_i) < \mu_k$, we would need to have $a_k^{true}(B_i) - a_k^{rep}(B_i) > M_k$, meaning that the discrepancy in vote counts, as it relates to the assorter a_k , is greater than its reported margin. If the margin isn't extremely small, and the errors in the vote count are uncorrelated and rare, this is very unlikely to happen. We believe that this should encourage choosing a very small value for δ , since it would only make the audit inefficient if it's likely that the vote counting was malicious.

► **Conclusion** (informal). *A Batchcomp RLA is more efficient when $\delta > 0$ is very small, as long as the vote tallying is not done maliciously.*

C Census RLA – Data Generation

The data used to perform this simulation is based on the population census conducted in 2021 [10]. The Statistical Service of Cyprus publicly reports the total number of residents in every district, but not the individual household data, which the census RLA requires. To generate this data, we assumed that the number of residents per household distributes as it does in the United States, as reported by its census [3]. We additionally assumed that 1% of households do not respond to the census and are counted as if they have no residents. The per-household data used in these simulations was generated as follows:

1. The number of households per district was calculated by dividing the district's population by the expected number of residents per household.

2. The number of residents in each household was drawn from the distribution specified in the US census [3].
3. Due to the randomness involved in the previous step, the real census and our generated one might disagree on the population of the districts. To balance this, the constant of each district (c_s in (5) at Section 5.2) was set as the difference between the population of the district according to the real census and according to our generated one. With this definition, the allocations of representatives to districts by the real census and by our generated one are necessarily identical.

D Weakening the RLA Guarantee

When conducting a SHANGRLA based RLA, a single assertion may be the difference between reading relatively few or a relatively many ballots or households to approve the reported outcome. As an example of this, in the Knesset elections examined in Section 4.3, a single assertion causes the Batchcomp audit to read 85% of ballots, instead of only 32% without it. In such cases, the auditing body may decide in advance that a certain assertion is too difficult to audit, and forgo approving it. This decision can be taken based on the assertion's margin, or by simulating the audit in advance and checking the number of ballots required per assertion.

For RLAs which approve an allocation of parliament seats to different political parties or federal-states, tight assertions can be altered to verify that the reported allocation of seats is nearly accurate. For example, in the Knesset elections, if an assertion which involves some specific party drastically increases the number of ballots the audit reads, an RLA can approve that the number of seats that this party wins according to the reported results is at most ± 1 from its true number. This would result in a shorter audit, at the expense of a slightly weaker guarantee.

To achieve this, when designing move-seat assertions which involve some difficult-to-audit party p , we alter the number of seats it reportedly won. For every assertion which verifies that seats shouldn't be moved *from* p to some other party p' , we imagine p reportedly won one seat less than it actually did. Similarly, when verifying that seats shouldn't be moved *from* p' to p , we imagine p has reportedly won one seat more than it did. The same notion also applies for census RLAs and their asserters.


Alternatively, if some assertions are too difficult to audit, the auditing body can decide to verify that certain blocks of parties or federal-states get the correct number of seats. For parliamentary elections, this is achieved by partitioning the parties to electoral blocks, and verifying that no seats should be moved between every two parties who belong to different blocks.

Bidding Strategies for Proportional Representation in Advertisement Campaigns

Inbal Livni Navon ✉ 
Stanford University, CA, USA

Charlotte Peale ✉ 
Stanford University, CA, USA

Omer Reingold ✉ 
Stanford University, CA, USA

Judy Hanwen Shen ✉ 
Stanford University, CA, USA

Abstract

Many companies rely on advertising platforms such as Google, Facebook, or Instagram to recruit a large and diverse applicant pool for job openings. Prior works have shown that equitable bidding may not result in equitable outcomes due to heterogeneous levels of competition for different types of individuals. Suggestions have been made to address this problem via revisions to the advertising platform. However, it may be challenging to convince platforms to undergo a costly re-vamp of their system, and in addition it might not offer the flexibility necessary to capture the many types of fairness notions and other constraints that advertisers would like to ensure. Instead, we consider alterations that make no change to the platform mechanism and instead change the bidding strategies used by advertisers. We compare two natural fairness objectives: one in which the advertisers must treat groups equally when bidding in order to achieve a yield with group-parity guarantees, and another in which the bids are not constrained and only the yield must satisfy parity constraints. We show that requiring parity with respect to both bids and yield can result in an arbitrarily large decrease in efficiency compared to requiring equal yield proportions alone. We find that autobidding is a natural way to realize this latter objective and show how existing work in this area can be extended to provide efficient bidding strategies that provide high utility while satisfying group parity constraints as well as deterministic and randomized rounding techniques to uphold these guarantees. Finally, we demonstrate the effectiveness of our proposed solutions on data adapted from a real-world employment dataset.

2012 ACM Subject Classification Theory of computation → Theory and algorithms for application domains

Keywords and phrases Algorithmic fairness, diversity, advertisement auctions

Digital Object Identifier 10.4230/LIPIcs.FORC.2023.3

Supplementary Material *Software (Source Code)*: <https://github.com/heyjjudes/bidding-strategies-for-proportional-representation>
archived at `swh:1:dir:f530386b342b1173b7ef36fd3ae2fa1cbf8ff7b9`

Funding *Inbal Livni Navon*: Supported by the Sloan Foundation Grant 2020-13941 and the Zuckerman STEM Leadership Program.

Charlotte Peale: Supported by the Simons Foundation Collaboration on the Theory of Algorithmic Fairness.

Omer Reingold: Supported by the Simons Foundation Investigators Award 689988.

Judy Hanwen Shen: Supported by the Simons Foundation Collaboration on the Theory of Algorithmic Fairness.



© Inbal Livni Navon, Charlotte Peale, Omer Reingold, and Judy Hanwen Shen;
licensed under Creative Commons License CC-BY 4.0
4th Symposium on Foundations of Responsible Computing (FORC 2023).
Editor: Kunal Talwar; Article No. 3; pp. 3:1–3:22



Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

For many institutions, hiring a diverse workforce is crucial to achieving and retaining an equitable environment. While there are many strategies that can be employed to ensure that each stage of the hiring process, from initial resume screening to a final hiring decision, can be done equitably, even the best of attempts may fall short if the initial pool of applicants lacks sufficient diversity. As a result, many companies rely on online advertising platforms such as Google, Facebook, or Instagram to recruit a wider applicant pool for job openings.

Advertising platforms sell slots to advertisers through auction mechanisms. Toward the goal of yielding a diverse applicant pool, advertisers are able to create recruitment and marketing campaigns to target users of different demographic groups. This specific but salient setting of job advertisements is bound by policy oversight from different government entities. In the United States, the Equal Employment Opportunity Commission enforces discrimination laws that prohibit employers from “*publishing a job advertisement that shows a preference for or discourages someone from applying for a job based on his or her race, color, religion, sex, national origin, age, disability or genetic information*”¹. However, it is unclear whether this guidance refers to bidding equally on individuals from different demographics, or to achieving a proportional yield for all demographics regardless of protected class status.

Prior work has observed that these two goals may not be equivalent. Due to differences in advertiser demand, the required costs to reach users of various demographic groups can be very different; women in particular may see fewer job ads due to competition from retail brands that do not target men [21]. As a result, setting the same bid value for all groups may still result in a disproportionate representation in downstream yield when there are different levels of competition for different groups of users on the platform [15, 10]. Existing work (such as [10, 12] with more discussed in Section 2) interprets this behavior as a failure of the mechanism due to composition, and suggests ways that ad auctions could be redesigned to guarantee fair outcomes despite these composition effects.

However, an advertising platform may be unlikely to implement a new auction mechanism for a number of reasons, even if the revenue of the new alternative can be shown to be close to that of the original mechanism. For instance, the costs necessary to research, deploy, and completely redesign the current auction system may make such a change undesirable. Moreover, a new auction might make the mechanism far more complex and difficult for advertisers to understand as well as offer less flexibility if it is designed with only a few specific types of constraints and objectives in mind. We discuss these concerns in more detail in Section 3.

Instead, we take the perspective that perhaps only requiring advertisers to bid values that are similar across different groups of interest may not be the most useful requirement for this context if the fairness of the system is judged by the outcomes of the auctions, and not the bids that are inputted. In fact, in Section 3.1, we demonstrate that requiring advertiser’s bids to be similar across groups may actually *hinder* achieving parity with respect to auction outcomes, and show that the utility of the optimal bidding strategy that satisfies parity constraints at both the bid and outcome level can be far lower than the optimal bidding strategy that requires group parity at the outcome level alone.

We use these arguments and examples to motivate an alternative approach to redesigning auctions, which is to consider the perspective of an individual advertiser and design bidding strategies that guarantee outcomes that meet the advertiser’s goals. This approach is often referred to as an “autobidder,” and the adoption of such technologies as a way to control

¹ <https://www.eeoc.gov/prohibited-employment-policiespractices>

spending and budget depletion is growing increasingly popular. There are a few works that consider autobidding for group parity goals [9, 23], but these approaches do not give formal guarantees about how closely the resulting bidding strategies meet the desired parity constraints.

We argue that due to their flexibility, practicality, and ease of implementation in existing systems, autobidding strategies that guarantee proportional representation across key subgroups are a key direction for research in equitable online ads. In this paper, we show how we can build on the autobidding framework of Aggarwal et al. [1] to develop an efficient algorithm to compute bidding strategies with provable proportional group representation guarantees in the offline setting. We additionally show how our constraints fit into the model studied by Castiglioni et al. [7] to provide efficient online bidding algorithms with sublinear regret. We focus on strategies for a single autobidder, though understanding market dynamics when many autobidders with fairness constraints are deployed is a natural next step for future work.

We supplement our arguments and algorithms with empirical evidence using data modified from the American Community Survey. By comparing single bid, gender-based bids, and our autobidder, we see that the autobidder achieves the best combination of utility and representation across different job sectors with different levels of representation.

1.1 Contributions

Motivations and Fairness Notions

Through examples and qualitative analysis, we consider different potential notions of fairness and group representation for ad auctions and make the case for using autobidders to achieve equitable ad exposure (Sections 3 and 4.1).

Optimal Randomized Bidding Strategy for Budget and Group Representation Constraints

Building on the autobidder with constraints framework suggested in [1], we add constraints on the representation of key subgroups in the set of clicks resulting from a series of auctions and show that it is possible to calculate an approximately optimal bidding strategy that stays within budget while satisfying group representation constraints in expectation. We consider two platform revenue schemes: one in which advertisers only pay for clicks and another where advertisers pay for impressions (Section 4).

Bidding Strategy with Deterministic Constraint Guarantees

Our model assumes that each individual i clicks on an ad with some probability ctr_i , therefore there is inherent randomness in the outcome, and it is not possible to have a deterministic promise on any of the constraints. However, we give a modification to the algorithm that results in slightly lower utility, but satisfies the constraints with high probability, and not only in expectation. The randomized rounding method assumes that there exists a large fraction of the population without representational constraints (Section 4.4).

Rounding for Deterministic Solutions When Groups are Disjoint

In the special case of disjoint groups and constraints on group representation with respect to impressions (rather than clicks), we show how to achieve a deterministic solution with utility that is close to optimal. This solution works also for a small number of intersection groups. For intersecting groups, it is possible to use the randomized rounding described above, that promises the constraints are met with high probability (Section 4.4).

Extension to Autobidding with Representation Constraints in the Online Setting

We show that our constraints can be satisfied by an online algorithm with sublinear regret using the online learning framework of Castiglioni et al. [7] (Section 4.5).

Empirical Data on Autobidder Performance

Using data adapted from the American Community Survey and the US Bureau of Labor Statistics, we simulate the results of different bidding strategies. We show the advantage of our proposed autobidder with proportional representation constraints and randomized rounding for achieving both equitable exposure and high advertiser utility (Section 5).

2 Related Work**2.1 Mechanism Design**

A number of existing works consider ways to design auctions that satisfy different choices of fairness guarantees. Chawla et al. [11] design truthful auctions that guarantee individually fair outcomes, Celis et al. [8] incorporate group parity constraints into an auction mechanism, and Kuo et al. [20] propose a deep learning approach to approximately optimal auctions while incorporating relaxed individual fairness constraints. Dwork et al. [15] observe that even when advertisers place bids that fulfill their personal fairness goals, competition from other advertisers may prevent the auction outcomes from satisfying advertisers' fairness constraints. Building on this observation, subsequent works design auction mechanisms whose outcomes satisfy individual fairness constraints for each advertiser, assuming that advertisers' bids satisfy individual fairness guarantees with respect to their personal metrics [10, 12].

2.2 Autobidding Strategies

A different approach for achieving advertising auction goals is to consider the problem from the point of view of the advertisers (bidders) and design bidding strategies that guarantee the desired properties. This approach is often termed “autobidding”. While a number of works explore autobidding strategies for a number of different budgets and spending-related goals [4, 6, 13, 22, 1], autobidding strategies that optimize for fairness guarantees are still relatively under explored. Nasr et al. [23] first suggest adding parity constraints specifically to bidding strategies using an unlimited budget. Celli et al. [9] explore autobidding strategies that incorporate parity constraints via a regularization term in the objective function. However, this approach does not allow for any formal guarantees about how well these fairness goals are achieved by the algorithm. More recently, Castiglioni et al. [7] consider more general autobidding strategies that can handle a number of different types of constraints. While they do not consider fairness guarantees in their main results, they note that fairness constraints could be considered as a future direction and suggest one potential formulation of fairness constraints. Unfortunately, the fairness constraints they suggest are not proven to yield an efficient algorithm because the solutions are not guaranteed to be feasible. We show that our constraints do satisfy feasibility requirements and yield an efficient online autobidding algorithm.

Another related line of work considers how to “learn to bid” or how to discover bidding strategies using feedback from the outcomes of repeated auctions [3, 16, 18, 19, 24, 25]. Our work mostly considers the offline full information setting in which the winning bids, items, and values are all known to the autobidding algorithm. In Section 4.5, we note that the

online learning framework developed by Castiglioni et. al in [7] can be extended to give online bidding algorithms that guarantee group proportionality constraints will be satisfied in the long-term when slots and their associated values are drawn from a stationary distribution at each step. However, further exploring how to learn fairness-aware bidding strategies in the repeated auction setting is an interesting direction for future work.

3 Motivation

We consider a two-part system, in which advertisers place bids on individual ad slots according to some bidding algorithm and a centralized auction mechanism decides which advertiser gets a particular slot, and how much they will pay. In reality, platforms like Google will often perform both parts of this process once an advertiser defines a campaign with a target population, total budget (B), and potentially some additional desiderata. Currently, most ad platforms use a standard second- or first-price auction to decide how ads are allocated, though there have been proposals for alternative options that guarantee a variety of different fairness objectives (See Section 2).

We observe that there are a number of different reasons why it makes sense to focus on designing a new bidding algorithm rather than implementing alterations to the auction mechanism itself when the system contains bidders who would like to ensure their ads result in an even spread of clicks from their target population.

Cost to Platform

Most alternative options do not consider the potential loss in revenue for the platform that would arise from implementing a new auction mechanism. Even when an alternative mechanism can provide near-optimal platform utility, significant costs associated with designing, implementing, and switching over to a new mechanism are likely to make such a switch impractical from the point of view of a platform.

Loss of Flexibility

There are many different objectives and constraints that advertisers would like to optimize for. While we focus on group representation constraints, some advertisers may be more focused on other objectives such as alternative notions of fairness, or goals outside fairness such as a limit on their rate of spending. Keeping the auction as a fixed mechanism allows advertisers to specify their own individual constraints and optimize their bids to match.

Decreased Comprehension

Ad platforms prioritize simplicity in their auction mechanisms. For this reason, many companies including Google have recently decided to switch from second- to first-price auctions, citing concerns around simplifying the ad-purchasing process for advertisers [17]. It, therefore, seems unrealistic to expect platforms to switch to the more complicated mechanisms required to enforce fairness guarantees.

Due to these reasons, we concentrate on the question of designing optimal bidding algorithms for advertisers with group representation constraints.

3.1 What Does it Mean to Bid Fairly in Second and First-Price Auctions?

Prior works have observed a composition problem that arises in standard auction settings [15]. As a simple example, we assume that some advertiser values individuals from groups A and B equally, and so bids the same value v on individuals from each group.

In a vacuum, such a strategy would result in having a proportional number of ads shown to both groups. However, other advertisers in the market may not have the same goals, and may specifically target one group by only bidding on individuals in group A . When composed together, the many bidding strategies used by all the different advertisers in the market may result in different winning bid values for the two groups. In particular, an individual from group B may require a winning bid of v , but individuals from A may require a higher bid of $2v$ due to increased demand. In this situation, our advertiser's strategy will result in ads shown only to group B , rather than to both groups proportionally.

The perspective of existing work is that in this example, our advertiser was “doing the right thing”, i.e. bidding on groups similarly, and it is a failure of the composition mechanism (the auction) that causes differential rates of ad exposure. Instead, we argue that bidding in a way that satisfies group parity constraints might not be the right notion for this context given that practical goals of diverse recruitment are judged based on the auction outcomes. In fact, there are three potential general types of fairness that could be considered here, defined by different parts of the bidding process. We discuss these three options below in terms of group parity guarantees. However, this framing applies to other notions such as individual fairness as well.

1. (*Bid Parity*) The advertiser is required to bid similarly across different groups of interest. This is the notion we considered in our above example, where we saw that bid parity alone does not guarantee that yields will satisfy any sort of group proportionality goals.
2. (*Outcome Parity*) Instead, we could explicitly require that an advertiser's yield (measured either in terms of clicks or exposures, depending on the setting) has representation of key groups that is proportional to their representation in the population. Here, we do not put any constraint on how advertisers must bid to achieve a proportional yield.
3. (*Bid-and-Outcome Parity*) Lastly, we could potentially consider a stricter notion that requires that both an advertiser's bids be similar across groups *and* the resulting yields be proportional.

If we care about the outcomes of ad auctions, it's natural to focus on either outcome parity or bid-and-outcome parity as goals for a bidding algorithm. On first glance, these might seem somewhat similar. Clearly, any strategy satisfying bid-and-outcome parity will also satisfy outcome parity, however, we can show that the opposite direction does not necessarily hold. In fact, a simple example demonstrates that strategies satisfying bid-and-outcome parity may result in arbitrarily large decreases in advertiser utility compared to strategies that are only required to satisfy outcome parity in both second and first-price auctions.

► **Example 1.** We consider a second-price auction² being run on a population partitioned into two groups, A and B .

² By similar reasoning, it's easily verified that a first-price auction run in the same setting would result in even larger gaps in utility, so we concentrate on second-price auctions for this example. We also only focus on yield in terms of exposure here for simplicity, but the example can be easily extended to work for yield that is measured in terms of clicks as well by incorporating click-through-rates.

We suppose that an advertiser has a budget of \$5 that it uses to bid on a population of 100 individuals $(G, w) \in P \subseteq \{A, B\} \times W$, where G corresponds to an individual's group, and w corresponds to the winning bid from a discrete set of bids $W = \{\$0.1, \$0.4, \$1\}$ (if an advertiser bids $b \geq w$, they win the auction and pay w , and do not win the auction and pay nothing otherwise).

Groups are distributed evenly across the population, so there are 50 individuals from group A and 50 individuals from group B. However, the distributions of winning bids are skewed slightly to the right (higher cost) for individuals from group A compared to group B, i.e. we have the following numbers of individuals with each winning bid:

	w = \$0.1	w = \$0.4	w = \$1
Group A	25	20	5
Group B	40	10	0

We consider an offline setting where these winning bids and numbers of individuals are all known to an advertiser beforehand and used to set a bidding strategy, and then these 100 individuals arrive in a random order and the bidding strategy is applied until the budget runs out. We consider two options for bidding strategies. First, a bid-constrained strategy is one where an advertiser must set a maximum bid b , and bid b on every individual that arrives until the budget runs out (this translates to bidding b on every individual with equal probability because the order is randomized). In Appendix A, we discuss why this is a natural definition of bid parity in this setting. The second option is to use a bid-unconstrained strategy. In this approach, an advertiser can set a unique bid for each individual.

We assume that an advertiser values all individuals equally, and thus its utility is equal to the number of individuals that are shown an ad. When the outcome is required to be proportional to the group sizes, an advertiser must bid in such a way that the expected number of ads shown to group A is equal to the number of ads shown to group B.

When bids are unconstrained and the advertiser can decide the bid amount for each individual separately as long as the outcomes are proportional to group sizes, it's optimal for an advertiser to bid \$0.1 on 25 individuals with $w = \$0.1$ from group A and 25 individuals with $w = \$0.1$ from group B, and bid $w = \$0$ on every other individual. This results in ads shown to 50 individuals total, which is equal to the optimal number that could be reached even when outcomes are unconstrained.

In contrast, when an advertiser must use a bid-constrained strategy, setting the maximum bid b to be any value smaller than \$1 cannot satisfy group proportionality constraints because the expected number of individuals shown ads from group B will always be larger than for group A. Thus, the only strategy that satisfies both bid and outcome parity is to bid the maximum-possible bid of \$1 on all individuals until the budget runs out.

This results in a strategy that shows ads to only 21.3 individuals in expectation, less than half the utility of the bid-unconstrained strategy. Moreover, note that this strategy provides the lowest utility of *any* of the potential bid-constrained strategies.

This example exhibits a setting in which requiring bid parity in addition to outcome parity may result in much lower utility for advertisers. We note that this example can be extended to even larger spreads of price distributions where the distribution of group A is slightly skewed right in comparison to group B, again requiring a bid-constrained strategy to bid the maximum possible winning bid of any individual to receive proportional outcomes, whereas a bid unconstrained strategy can satisfy outcome parity while matching the utility of the optimal unconstrained bidding strategy.

We conclude that requiring advertisers to bid in a way that respects parity constraints does not directly contribute to receiving group-proportional outcomes, and in some situations may actually make achieving such outcomes incredibly costly compared to strategies where bids are unconstrained. This motivates our interest in optimal bidding strategies that satisfy outcome parity, which we explore in the following sections.

4 Autobidder with Constraints on Subgroup Representation

Now that we have justified our perspective and proposed approach, we describe how we choose to model an ad auction from an advertiser’s point of view, and how to compute optimal bidding strategies for this setting.

4.1 Setup

We consider a large set of queries (or individuals) I , each of which has a single slot that can show an ad. For each query i , an auction determines which ad is shown as well as the cost-per-click (cpc_i) of the ad.

We consider a static setting in which we are trying to set the bid of single advertiser with full knowledge of the bids of other advertisers, i.e. there is a set cpc_i for each query, and the advertiser wins the ad if and only if their bid is above that value. In a first-price auction, the winning bidder pays their bid. In a second-price auction, the winning bidder will pay cpc_i . This is a practical assumption in larger markets since cpc_i remains stable. Because our model assumes that we know cpc_i for each individual, the optimal strategies for first-price and second-price auctions are equivalent, because there is no need to bid higher than whatever would be paid in a second-price auction.

Each query also has an associated click-through-rate ($ctr_i \in [0, 1]$) and value to the advertiser: $v_i \geq 0$. A bidder’s goal is to select the optimal set of queries I^* that maximize its expected value $\sum_{i \in I^*} v_i ctr_i$, subject to a set of budget constraints and representation constraints. Budget constraints ensure that the advertiser’s expected cost stays below some threshold. Here we will focus on the simplest type of budget constraint that just requires the total expected cost is within a budget B : $\sum_{i \in I^*} ctr_i cpc_i \leq B$. However, our approach can be extended to more complicated sets of budgetary constraints.

The second type of constraint we consider is a *group representation constraint*, which allows the advertiser to ensure that the clicks it receives contain sufficient representation from key demographic groups. We allow an advertiser to specify its goal via a set of constraints that require the proportion of clicks from a particular group $g \subseteq I$ to be at least some goal value μ_g , i.e. $\sum_{i \in I^* \cap g} ctr_i \geq \mu_g \sum_{i \in I^*} ctr_i$.

4.2 Optimal Ad Allocation as a Linear Program

We express the search for an optimal I^* as described above as an integer linear program, in which the variables x_i correspond to whether or not the advertiser should win the auction for the i th slot. We assume that the advertiser’s spending is limited by a budget B , and we are given a set of groups G , where each $g \in G$ is associated with a lower bound on the desired fraction of total clicks that come from group g , $\mu_g \in [0, 1]$. For a group g , denote $g_i := \mathbf{1}[i \in g]$ as a binary indicator variable for query i ’s membership in g .

$$\begin{aligned}
& \text{maximize} && \sum_i x_i \text{ctr}_i v_i \\
& \text{subject to} && \sum_i x_i \text{ctr}_i c p c_i \leq B \\
& && \sum_i x_i \text{ctr}_i (\mu_g - g_i) \leq 0, \forall g \in G \\
& && x_i \in \{0, 1\}, \forall i \in I.
\end{aligned} \tag{1}$$

We can relax the above program by allowing $0 \leq x_i \leq 1$, where fractional x_i s can represent the probability the advertiser should win the auction for slot i . We denote it as the relaxed ad allocation linear program.

► **Theorem 2.** *Let P be a relaxed ad allocation linear program. Let \mathcal{V} be an bound on the objective value, and for each constraint c , let V_c be an upper bound on the violation of constraint c . Then for every $\delta > 0$, Algorithm 3 outputs a solution $x \in [0, 1]^n$ with utility within $\delta\mathcal{V}$ of the optimal utility achievable by the relaxed linear program and violates the each of the constraints with up to δV_C additive error.*

On Lemma 5 we show that under certain conditions it is possible to have a randomized rounding algorithm satisfying all the constraints with high probability, and in Lemma 6 we show that for disjoint groups, there is a deterministic rounding algorithm satisfying the constraints with a small additive error.

We prove the theorem by adapting the multiplicative weights algorithm presented in [1], where it was used to solve a linear program with only budgetary constraints. We show that this approach can be modified to work for our setting as well.

At a high level, the algorithm from [1] assumes some known rough bounds on the maximal objective value of the linear program, \mathcal{V} , and rough bounds on the amount of violation of each constraint. It then searches for the optimal objective by considering candidate objective values V and for each V , searching for a solution whose objective value is equal to V . The search is done by a multiplicative weights algorithm that solves a series of one-dimensional problems. In this setting, the solution for each of these one-dimensional problems has a closed form in terms of a thresholds T_i . The algorithm runs in time $O(n^2/\delta^4|G|)$ to get a $\delta\mathcal{V}$ -approximate solution, where \mathcal{V} is a bound on the maximal possible utility value, i.e. $\sum_i x_i \text{ctr}_i v_i \leq \mathcal{V}$ for every x . The algorithm also uses bounds V_c on the constraint violations.

In Section 4.3 we show that there exists an equivalent threshold T_i for the linear program with fairness constraints. In Appendix B we write the approximation algorithm for fairness constraints and prove its correctness using the adapted threshold. Using our threshold, the multiplicative weights algorithm can solve the 1-dimensional problem for the linear program with fairness constraints.

► **Note 3.** It is important to note that when seeking integer solutions, certain choices of fairness and budget constraints may be so strict that the only feasible solution is one where no bids are made. This can happen even when fairness constraints would be feasible with an unlimited budget, but are too costly to implement with limited funds.

In such cases, it would be easy for an autobidding algorithm to notify an advertiser that an inputted constraint set is infeasible. There are many potential ways to relax the budget and/or fairness constraints to achieve a non-trivial feasible solution. However, we want to note that which relaxation an advertiser selects should be given careful consideration as to whether it still aligns with the advertiser’s goals and does not disproportionately affect any particular group. What constitutes a “fair” relaxation of a constraint set and how to find minimal relaxations with these guarantees is an interesting question for future work.

4.3 Solutions to the Linear Program

In this section we show that all optimal solutions to LPs of the type described in (1) have a specific structure.

As a first step, we write the linear program and its dual, allowing the solution x to be fractional.

$$\begin{array}{ll}
 \text{maximize } \sum_i x_i \text{ctr}_i v_i & \text{minimize } \sum_i \delta_i + \alpha B \\
 \text{s.t. } \sum_i x_i \text{ctr}_i \text{cpc}_i \leq B & \text{s.t.} \\
 \sum_i x_i \text{ctr}_i (\mu_g - g_i) \leq 0, \forall g \in G & \delta_i + \alpha \text{ctr}_i \text{cpc}_i + \sum_g \beta_g \text{ctr}_i (\mu_g - g_i) \geq \text{ctr}_i v_i \\
 0 \leq x_i \leq 1, \quad \forall i \in I & \alpha, \delta_i, \beta_g \geq 0, \quad \forall g \in G, i \in I
 \end{array} \tag{2} \tag{3}$$

We show that there is an optimal bidding threshold T_i such that if $x_i^* = 1$ in the optimal solution to the LP above, we have $T_i \geq \text{cpc}_i$, and if $x_i^* = 0$, we have $T_i \leq \text{cpc}_i$.

Note that if these inequalities were strict (i.e. $T_i < \text{cpc}_i$ and not \leq), T_i would provide an optimal bidding formula whose outcomes would match that of the optimal solution. For a second-price auction, the bids would consist of exactly T_i , while for a first-price, the advertiser should bid cpc_i or $\text{cpc}_i + \epsilon$ if this is the winning bid) whenever $T_i > \text{cpc}_i$. Because the inequalities are not strict, these thresholds are only used as an intermediate step in the algorithm used to solve the linear program (see Appendix B).

► **Theorem 4.** *Let \mathbf{x}^* be the optimal solution to 2, and for each $i \in I$, let T_i be*

$$T_i := \frac{v_i - \sum_{g \in G} \beta_g (\mu_g - g_i)}{\alpha}. \tag{4}$$

Then, $x_i^ = 0$ implies that $T_i \leq \text{cpc}_i$, and $x_i^* = 1$ implies $T_i \geq \text{cpc}_i$, with the latter inequality strict whenever $\delta_i > 0$.*

We prove the theorem via analyzing the complementary slackness conditions of the primal and dual LPs. The proof appears on Appendix C.

4.4 Rounding the Solution

The solution to this linear program is a vector $x \in [0, 1]^n$ that maximizes the objective subject to the given constraints. In this section, we show how to round a fractional solution into an integer solution satisfying the constraints and achieving nearly optimal objective value.

Randomized Rounding

One way to interpret the fractional solution $x \in [0, 1]^n$ is as a probabilistic solution. That is, for every individual $i \in [n]$, bid cpc_i with probability x_i , and else bid 0. Let $y \in \{0, 1\}^n$ be a vector corresponding to a run of this random process, i.e. for every i , $y_i \sim \text{Ber}(x_i)$ independently. Let $r_i \in \{0, 1\}^n$ be the vector indicating whether an individual clicked on the ad, i.e. for all i , $r_i \sim \text{Ber}(\text{ctr}_i)$. By definition, it means that for all i , $\mathbb{E}[y_i] = x_i$ and $\mathbb{E}[r_i] = \text{ctr}_i$.

Since each r_i is a random variable decided by individual i , there is an inherent randomness in the outcome and constraint values. Even if we had a deterministic rounding algorithm generating y from x , the uncertainty in r does not disappear and we do not get a deterministic expression for the objective and constraints. This does not mean that the advertiser would not prefer a stronger guarantee from the solution y . For example, the advertiser might want

to never exceed the budget. Given a fractional solution x satisfying certain conditions, we show a randomized rounding algorithm that generates y satisfying all of the constraints with high probability, while only reducing the expected utility by a small factor.

For ease of notation, we say that an ad allocation linear program (2) and a solution $x \in [0, 1]^n$ are γ -flexible if the set $S_0 = \left\{ i \in [n] \mid \sum_{g \in G} g_i = 0 \right\}$ satisfies $\sum_{i \in S_0} x_i \text{ctr}_i \geq \gamma \sum_{i \in [n]} x_i \text{ctr}_i$ and $\sum_{i \in [n]} x_i \text{cpc}_i \text{ctr}_i \geq \gamma n$. We remark that if an individual i has $g_i = 1$ only for groups g such that $\mu_g = 0$, then effectively it is not in any constraint and therefore can be added to S_0 .

Our rounding algorithm only works for γ -flexible solutions. We remark that some flexibility in the constraints is required for any rounding algorithm, as can be seen from the following example. Suppose $G = \{g_1, g_2\}$ and that we have two constraints requiring that exactly $1/2$ of the clicks should be from individuals $i \in g_1$ and $1/2$ from $i \in g_2$. Then, because of the inherent randomness in the clicks created by $r_i \sim \text{Ber}(\text{ctr}_i)$, it is not possible to promise that both constraints are satisfied with high probability.

■ **Algorithm 1** Randomized rounding algorithm for γ -flexible linear program and solution.

Input: $x \in [0, 1]^n, S_0 \subset [n], \epsilon > 0$

Output: $y \in \{0, 1\}^n$

for $i = 1$ **to** n **do**

$$\left[\begin{array}{l} x'_i \leftarrow \begin{cases} (1 - \epsilon)x_i & i \in S_0 \\ (1 - \epsilon/2)x_i & i \notin S_0. \end{cases} \\ y_i \leftarrow \text{Ber}(x'_i) \end{array} \right.$$

► **Lemma 5.** *Let P be an ad allocation linear program, and let $x \in [0, 1]^n$ be a fractional solution such that P, x are γ -flexible and x matches the representation constraint values of the optimal solution to P up to a multiplicative error at most 2. Then for every constant $\epsilon > \gamma$ Algorithm 1 outputs a solution $y \in \{0, 1\}^n$, satisfying the following. Let r be the vector representing the individuals clicks, and $\mu = \min_{g \in G} \{\mu_g\}$. Then with probability $1 - \exp(-\mu \epsilon^2 \gamma^3 n)$ over the randomness of y, r we have*

$$\sum_i y_i r_i \text{cpc}_i \leq \sum_i x_i \text{ctr}_i \text{cpc}_i, \quad (5)$$

$$\sum_i y_i r_i (\mu_g - g_i) \leq \sum_i x_i \text{ctr}_i (\mu_g - g_i) \quad \forall g \in G, \quad (6)$$

$$\mathbb{E} \left[\sum_i y_i r_i v_i \right] \geq (1 - \epsilon) \sum_i x_i \text{ctr}_i v_i. \quad (7)$$

The lemma implies that if $x \in [0, 1]^n$ satisfies the constraints, then with high probability y satisfies them also. If x approximately satisfies the constraints and has some small error δ , then with high probability y approximately satisfies the constraints with the same error. The proof appears on Appendix C.

Deterministic Rounding

An interesting variant of our autobidding problem is one in which the advertiser pays for individuals to *view* the ad, rather than clicking on it. This can be modeled by the bid allocation LP in (1) by setting $\text{ctr}_i = 1$ for every $i \in [n]$. In this setting there is no random variable r representing the clicks and thus no inherent randomness in the outcome. Therefore, we have motivation to discuss a deterministic rounding procedure.

3:12 Bidding Strategies for Proportional Representation in Advertisement Campaigns

We focus on the special case of disjoint groups, where each individual i has $g(i) = 1$ for exactly one $g \in G$ (some groups might not have constraints). The rounding procedure we present results in a deterministic solution that nearly satisfies all constraints and guarantees approximately optimal utility for the advertiser. Our rounding algorithm works for every solution $x \in [0, 1]^n$ satisfying the following condition

$$\forall g \in G, i, i' \in g \text{ such that } x_i, x_{i'} \in (0, 1), \quad v_i > v_{i'} \implies cpc_i > cpc_{i'}. \quad (8)$$

We remark that from the complimentary slackness, the optimal solution satisfies this condition. Furthermore, for $i, i' \in g$ on which the condition does not hold, i is strictly better than i' , so we can increase x_i and reduce $x_{i'}$ and get a better solution. More formally, suppose $x \in [0, 1]^n$ is a solution that does not satisfy Equation (8) for some g and $i, i' \in g$, then by changing x_i to $\min\{1, x_i + x_{i'}\}$ and $x_{i'}$ to $\max\{0, x_i + x_{i'} - 1\}$ we receive a new solution satisfying all constraints as the original solution, and has at least as good objective. Since checking this condition is efficient, we can easily turn every solution into one satisfying the above without hurting the guarantees.

► **Lemma 6.** *Let P be an ad allocation linear program with disjoint groups G and $ctr_i = 1$ for all $i \in [n]$, and let $v_{max} = \max_{i \in I} \{v_i\}$. For every $g \in G$, let $S_g = \{i \in [n] \mid g_i = 1, x_i \in (0, 1)\}$. For every fractional solution $x \in [0, 1]^n$ satisfying the constraints of P and Equation (8), Algorithm 2 applied on every set S_g outputs a solution $y \in \{0, 1\}^n$ such that*

$$\sum_{i \in [n]} y_i cpc_i \leq \sum_{i \in [n]} x_i cpc_i \leq B \quad (9)$$

$$\sum_{i \in [n]} y_i g_i + 1 \geq \mu_g \sum_{i \in [n]} y_i \quad \forall g \in G, \quad (10)$$

$$\sum_{i \in [n]} y_i v_i \geq \sum_{i \in [n]} x_i ctr_i v_i - |G| v_{max}. \quad (11)$$

At a high level, the rounding algorithm round down each group separately. That is, if $y \in \{0, 1\}^n$ are the rounded values, then for every group g and every value v we have $\sum_{i \in g, v_i \geq v} y_i \leq \sum_{i \in g, v_i \geq v} x_i$. See the proof on Appendix C for more details.

■ **Algorithm 2** Deterministic rounding for a single group S .

Input: $S = \{i_1, \dots, i_t\}, x \in [0, 1]^n, v \in \mathbb{R}^n$

Output: $y \in \{0, 1\}^S$

Assume that the elements in S are ordered according to v , i.e. $v_{i_1} \leq v_{i_2} \dots \leq v_{i_t}$ and in case of equality by cpc_i .

for $j = t$ **to** 1 **do**

if $x_{i_j} + \sum_{l > j} (x_{i_l} - y_{i_l}) \geq 1$ then
$y_{i_j} \leftarrow 1;$
else
$y_{i_j} \leftarrow 0;$

We remark that Algorithm 2 can also be applied in the case of a few not-disjoint set of groups G . In this case, we should run it separately over each possible intersection of the groups, i.e. for every $h \in \{0, 1\}^{|G|}$ run is on $S_h = \{i \in [n] \mid \forall g \in G, h_g = g_i\}$. In this case, instead of violating each constraint by an additive factor of 1, we have an additive error of $2^{|G|}$, the loss to the objective value can be $v_{max} 2^{|G|}$. Therefore, it only makes sense to apply this algorithm for either disjoint, or very few groups G .

4.5 Extension to Online Bidding

Thus far, our autobidder formulation follows prior work which examines an offline setting [1]. For a large enough advertisement market, generating bids in an offline setting is sufficient due to the high volume and frequency of slots. However, in settings where advertisement slots may be more sparse and there is a fixed time horizon, generating bids that respect budget and representation constraints can be modeled as an online stochastic optimization problem. We assume cpc_t , ctr_t , $\{g_t\}_{g \in G}$, and v_t are stochastic, meaning that at each time step t , a tuple consisting of these values are drawn i.i.d from some stationary distribution.

We can then define an objective $f_t(x_t)$ and constraints $c_{t,g}^{(0)}, \dots, c_{t,g}^{(3)}$ for each group $g \in G$ to give the optimization problem at the t th step

$$\begin{aligned} f_t(x_t) &= x_t ctr_t v_t \\ c_{t,g}^{(0)}(x_t) &= x_t ctr_t (\mu_g - g_t) \leq 0, \quad \forall g \in G \\ c_{t,g}^{(1)}(x_t) &= x_t ctr_t cpc_t - \rho \leq 0 \\ c_{t,g}^{(2)}(x_t) &= x_t - 1 \\ c_{t,g}^{(3)}(x_t) &= -x_t \end{aligned}$$

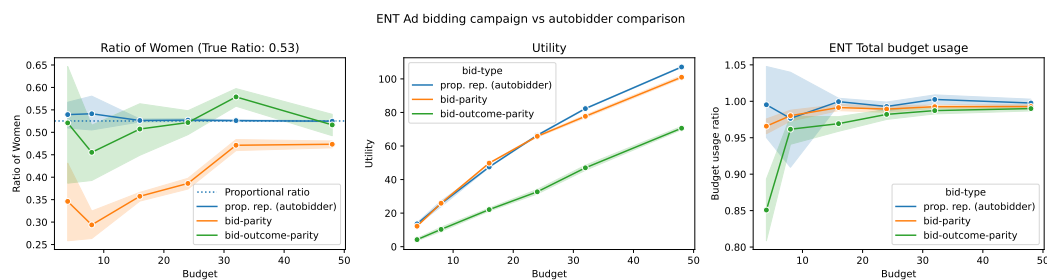
where $\rho = \frac{B}{T}$ is the goal amount of budget used at every step and T is the time horizon in consideration (i.e. campaign duration).

Using the algorithm for this problem proposed by [7] guarantees an approximate cumulative constraint satisfaction of $\frac{1}{T} \sum_{t=1}^T c_{t,g}^{(i)}(x_t) \leq \tilde{O}(T^{-1/4})$ for all $i \in [3]$ and $g \in G$. This means that across T steps, our group representation goals can be approximately achieved. Further, this algorithm also gives an upper bound of $\tilde{O}(T^{-1/4})$ on the regret. While [7] also proposed quota-based fairness constraints, they were unable to apply their algorithm because they could not assume the existence of a feasible solution. In contrast, our ratio-based representation constraints always yield a feasible solution: the zeros vector. Moreover, the existence of a strictly feasible solution implies even better guarantees on the cumulative constraint satisfaction and regret.

5 Experiments

To simulate the problem of an employer looking to advertise to a diverse set of candidates, we use data from the US Bureau of Labor Statistics and the American Community Survey. The American Community Survey is a yearly survey given to a sample of the United States population in order to determine how federal and state funds should be distributed. The survey collects information about employment, housing, education, demographic information, and other topics³. Using 2021 records of individuals in California from this survey [14], we construct cost-per-click based on an individual's income and estimate advertiser value by assigning a higher value for individuals in the same occupational category. To model the higher cost of advertising to women observed by prior works [21], we add an additional bump uniformly to the cost per click for women such that the average cost-per-click for women is 10% higher than men. We define click-through rates by assuming an individual is more likely to click on an ad if there are more people similar to themselves in the current occupation. This modeling assumption corresponds to stereotype threat [5]; the negative

³ <https://www.census.gov/programs-surveys/acs/about.html>



■ **Figure 1** Men and women each represent half of the workforce among *entertainment occupations* workers; we compare the consequences of different fairness objectives in a second price auction. When women cost more to reach than men, using an approach that enforces bid parity guarantees that ads will be shown disproportionately to men; this underrepresentation is particularly stark at a lower budget. Using an auto-bidder with constraints achieves proportional representation while maintaining higher utility than a strategy satisfying bid-and-outcome parity.

experience caused by being judged based on a negative group stereotype. Using Labor force summary statistics from 2021⁴, we use the gender and race distributions of occupational categories to approximate the click-through rates for an individual query. To account for the variance across income, demographic, and job categories, we add Gaussian noise to value (v_i), cost per click (cpc_i), and click-through-rate (ctr_i), and clip values to a small range.

In our experiment setting, we consider a larger pool of viewers both within and outside the target job industry. We set the values of individuals within an industry to be 1.0 and values for individuals in other occupations to be zero. For each budget, bids are estimated using a disjoint sample from the population that maximizes budget use. We approximate the parity-satisfying bid by finding the cpc threshold in the disjoint population where the budget would become exhausted. For the bid satisfying bid-and-outcome parity, we compare the cumulative distributions of cost-per-click for men and women respectively and find a non-zero intersection point. The cost-per-click at this point reaches a proportional number of men and women. And thus is both bid and outcome fair. While results from previous sections apply to both first and second-price auctions, this set of experiments will be based on second-price auctions. It is easy to see that if we looked at first-price auctions, the bid-parity and bid-and-outcome-parity strategies would be even less efficient in utility with the same budget.

Figure 1 compares the bid-parity and bid-and-outcome-parity strategies achieved by a single max bid threshold against our autobidder with proportional group representation constraints in the entertainment industry. This scenario in *entertainment occupations* is motivated by our original example from the introduction, where showing ads to men and women have different costs but men and women appear in the workforce in equal proportion. We see that focusing on bid parity yields a low ratio of women; this effect is especially stark when the total budget is lower. When bid-and-outcome parity is enforced, better representation can be achieved but the utility is strictly lower than the strategy satisfying bid parity. This is because requiring both bid-and-outcome parity results in inefficiency. We apply our autobidder with randomized rounding with parity constraints since parity is equivalent to proportional representation in this industry and plot autobidder candidates for the entertainment industry only. Since the autobidder will use all of the available budget,

⁴ <https://www.bls.gov/cps/cpsaat11.htm>

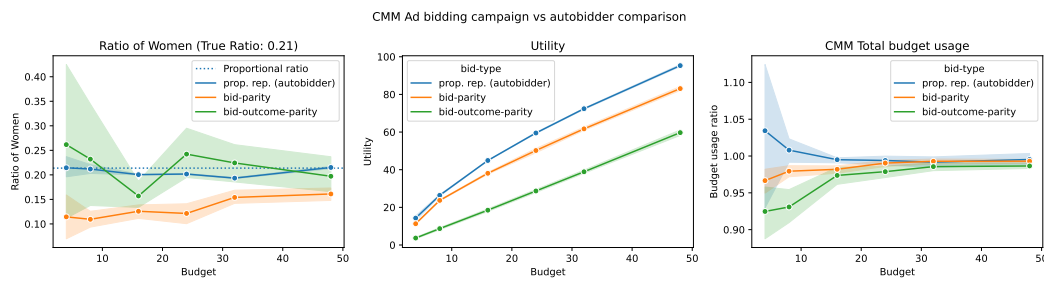


Figure 2 Women only represent 21% of the workforce in the *computer and mathematical* occupation; we again compare the consequences of various fairness strategies in a second price auction. A bid-parity strategy yields very low female representation but high utility. In contrast, a bid-and-outcome-parity strategy yields proportional representation but lower utility. Meanwhile using an autobidder with proportional constraints yields both good representation and high utility. For a large enough budget, the bid parity and bid-and-outcome parity bids are the same and achieve similar utility and representation.

female candidates not in the entertainment industry may also be selected. Thus while the total number of women candidates is exactly proportional, the number of women in the entertainment industry might be slightly less than proportional. However, our simulations show that the autobidder still achieves representation closer to proportional and yields higher utility than solutions satisfying bid-and-outcome parity.

Next, we turn to *computer and mathematical* occupations where women only represent 21% of workers in our sampled data. Repeating the same process for finding the optimal bid for strategies satisfying bid and bid-and-outcome parity, we can again compare these approaches to our autobidder with proportional representation constraints. Since workers in this industry have much higher incomes, we adjust the minimum cost per click to be slightly higher. In Figure 2, we observe that both our autobidder and the bid-and-outcome-parity strategy achieve better representation than the bid-parity strategy. Comparing utility, we once again observe a significant gap between autobidder and bid-and-outcome-parity utility where employing the autobidder achieves much higher utility. We once again see that the autobidder has higher utility than the bid-parity strategy for the same reason as previously mentioned. Utility-wise, for both occupations, the autobidder always matches or surpasses the bid-parity strategy since some individuals under the threshold may not be the most efficient choices; the autobidder might find a different combination of individuals which maximizes utility that a single threshold cannot achieve.

In both industries with vastly different baseline demographic compositions, we see that using our autobidder with proportional representation constraints achieves both high levels of representation and utility. For any underrepresented group or intersectional group, we can repeat these examples with similar expected results. If the required level of yield is beyond the population proportion, we can also adjust the target ratio accordingly.

6 Discussion

Even in the specific setting of group fairness, there are many definitions of fairness and parity that can arise in the advertisement auction and bidding process. We give examples to motivate three potential objectives that have been scattered throughout prior work. We discuss what different strategies for achieving each of these goals might look like and give examples of when one notion of fairness (i.e. in bids) might contradict other notions of

fairness (e.g. in yield outcome). Our experiments verify the observation from prior work that a strategy satisfying bid parity may result in a lack of diversity when some subgroups are more expensive to advertise to than others. Turning to the bid-and-outcome parity objective, where proportional group representation must be achieved via a bidding strategy that satisfies parity constraints, we show that these additional constraints require much higher bid values to ensure that all populations can be reached. In our simulations, bid-and-outcome parity does achieve better proportional representation than the bid-parity strategy but at the cost of significant utility loss.

Motivating the case for strategies that satisfy outcome parity, we extend on an existing autobidding framework to include group representation constraints based on the desired ratio of individuals from different groups. Since we use a probabilistic model of cost that is based on click-through rates, we also further modify the autobidder algorithm to satisfy budget and representation constraints with high probability, rather than just in expectation. Incorporating our proposed randomized rounding method that complements our autobidder solution, we show in our experiments that we achieve better outcome fairness than the bid-parity strategy and better utility than the bid-and-outcome-parity strategy.

In our simplified framework, we assumed that an individual's value to an advertiser can be easily derived based on information about the individual's occupational record. In a real advertising scenario, platforms might have only estimates of viewer employment. Furthermore, there might be systematic biases in terms of missing features like current occupation and income. Designing mechanisms to achieve outcome parity as well as other notions such as individual fairness in the presence of real world data challenges is a promising direction for future work. Furthermore, advertising for job recruitment is just one aspect of recruitment. In reality, a pool of candidates can come from a variety of sources including recruitment events, referrals, job search engines, and direct applications. Each stream of candidates involves different recruitment costs and yield groups with different levels of diversity and skill levels. Exploring composition effects across different sources of recruitment and the underlying network effects that affect which audiences are reached is another interesting direction for future research.

References

- 1 Gagan Aggarwal, Ashwinkumar Badanidiyuru, and Aranyak Mehta. Autobidding with constraints. In *International Conference on Web and Internet Economics*, pages 17–30. Springer, 2019.
- 2 Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of computing*, 8(1):121–164, 2012.
- 3 Santiago Balseiro, Negin Golrezaei, Mohammad Mahdian, Vahab Mirrokni, and Jon Schneider. Contextual bandits with cross-learning. *Advances in Neural Information Processing Systems*, 32, 2019.
- 4 Santiago R Balseiro and Yonatan Gur. Learning in repeated auctions with budgets: Regret minimization and equilibrium. *Management Science*, 65(9):3952–3968, 2019.
- 5 Maya A Beasley and Mary J Fischer. Why they leave: The impact of stereotype threat on the attrition of women and minorities from science, math and engineering majors. *Social Psychology of Education*, 15:427–448, 2012.
- 6 Christian Borgs, Jennifer Chayes, Nicole Immorlica, Kamal Jain, Omid Etesami, and Mohammad Mahdian. Dynamics of bid optimization in online advertisement auctions. In *Proceedings of the 16th international conference on World Wide Web*, pages 531–540, 2007.

- 7 Matteo Castiglioni, Andrea Celli, Alberto Marchesi, Giulia Romano, and Nicola Gatti. A unifying framework for online optimization with long-term constraints. *arXiv preprint*, 2022. [arXiv:2209.07454](https://arxiv.org/abs/2209.07454).
- 8 Elisa Celis, Anay Mehrotra, and Nisheeth Vishnoi. Toward controlling discrimination in online ad auctions. In *International Conference on Machine Learning*, pages 4456–4465. PMLR, 2019.
- 9 Andrea Celli, Riccardo Colini-Baldeschi, Christian Kroer, and Eric Sodomka. The parity ray regularizer for pacing in auction markets. In *Proceedings of the ACM Web Conference 2022*, pages 162–172, 2022.
- 10 Shuchi Chawla, Christina Ilvento, and Meena Jagadeesan. Multi-category fairness in sponsored search auctions. *arXiv preprint*, 2019. [arXiv:1906.08732](https://arxiv.org/abs/1906.08732).
- 11 Shuchi Chawla and Meena Jagadeesan. Individual Fairness in Advertising Auctions Through Inverse Proportionality. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*, volume 215 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 42:1–42:21, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. [doi:10.4230/LIPIcs.ITCS.2022.42](https://doi.org/10.4230/LIPIcs.ITCS.2022.42).
- 12 Shuchi Chawla, Rojin Rezvan, and Nathaniel Sauerberg. Individually-fair auctions for multi-slot sponsored search. *arXiv preprint*, 2022. [arXiv:2204.04136](https://arxiv.org/abs/2204.04136).
- 13 Vincent Conitzer, Christian Kroer, Eric Sodomka, and Nicolas E. Stier-Moses. Multiplicative pacing equilibria in auction markets. *Oper. Res.*, 70(2):963–989, March 2022. [doi:10.1287/opre.2021.2167](https://doi.org/10.1287/opre.2021.2167).
- 14 Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. Retiring adult: New datasets for fair machine learning. *Advances in Neural Information Processing Systems*, 34, 2021.
- 15 Cynthia Dwork and Christina Ilvento. Fairness under composition. *arXiv preprint*, 2018. [arXiv:1806.06122](https://arxiv.org/abs/1806.06122).
- 16 Zhe Feng, Chara Podimata, and Vasilis Syrgkanis. Learning to bid without knowing your value. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, EC '18, pages 505–522, New York, NY, USA, 2018. Association for Computing Machinery. [doi:10.1145/3219166.3219208](https://doi.org/10.1145/3219166.3219208).
- 17 Google. Faqs about adsense moving to a first-price auction. <https://support.google.com/adsense/answer/10858748?hl=en>. URL: <https://support.google.com/adsense/answer/10858748?hl=en>.
- 18 Yanjun Han, Zhengyuan Zhou, Aaron Flores, Erik Ordentlich, and Tsachy Weissman. Learning to bid optimally and efficiently in adversarial first-price auctions. *arXiv preprint*, 2020. [arXiv:2007.04568](https://arxiv.org/abs/2007.04568).
- 19 Yanjun Han, Zhengyuan Zhou, and Tsachy Weissman. Optimal no-regret learning in repeated first-price auctions. *arXiv preprint*, 2020. [arXiv:2003.09795](https://arxiv.org/abs/2003.09795).
- 20 Kevin Kuo, Anthony Ostuni, Elizabeth Horishny, Michael J Curry, Samuel Dooley, Ping-yeh Chiang, Tom Goldstein, and John P Dickerson. Proportionnet: Balancing fairness and revenue for auction design with deep learning. *arXiv preprint*, 2020. [arXiv:2010.06398](https://arxiv.org/abs/2010.06398).
- 21 Anja Lambrecht and Catherine Tucker. Algorithmic bias? an empirical study of apparent gender-based discrimination in the display of stem career ads. *Management science*, 65(7):2966–2981, 2019.
- 22 Brendan Lucier, Sarath Pattathil, Aleksandrs Slivkins, and Mengxiao Zhang. Autobidders with budget and roi constraints: Efficiency, regret, and pacing dynamics. *arXiv preprint*, 2023. [arXiv:2301.13306](https://arxiv.org/abs/2301.13306).
- 23 Milad Nasr and Michael Carl Tschantz. Bidding strategies with gender nondiscrimination constraints for online ad auctions. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 337–347, 2020.
- 24 Ashwinkumar Badanidiyuru Varadaraja, Guru Prashanth Guruganesh, and Zhe Feng. Learning to bid in contextual first price auctions. In *The Proceedings of the ACM Web Conference 2023*, 2023. [arXiv:2109.03173](https://arxiv.org/abs/2109.03173).

- 25 Jonathan Weed, Vianney Perchet, and Philippe Rigollet. Online learning in repeated auctions. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 1562–1583, Columbia University, New York, New York, USA, 23–26 June 2016. PMLR. URL: <https://proceedings.mlr.press/v49/weed16.html>.

A Remark on Definitions of Bid Parity

Throughout this paper, we define a strategy satisfying bid parity as one that selects a single maximum bid b_{max} and bids this value on every member of the target population until the budget runs out. We use this definition because it captures the standard setting in which advertisers can specify their preferences to online advertising platforms by creating a campaign parameterized by a budget, target population, and maximum bid. Moreover, natural relaxations to this strict notion of parity may result in notions that don't guarantee parity with respect to outcomes even in the absence of composition effects. We consider two potential relaxations here to illustrate.

A.1 Parity with Respect to Average Bids

We could imagine a situation in which rather than requiring advertisers bid the same bid with the same probability on all key subgroups, they are instead only required to have the same average bid for each group.

We show that even in the simplest case where we have two disjoint groups A and B of equal size making up the population and every individual has the same winning bid w , only requiring parity with respect to average bids can lead to outcomes where the representation of A and B is far from proportional.

In particular, consider a strategy that bids w on all individuals from group A , while bidding $w - \epsilon$ for some small $\epsilon > 0$ on 90% of individuals from group B , and bidding $w + 9\epsilon$ on the remaining 0.1%. For small ϵ the difference in bids is extremely small, but such a strategy will result in 10x the number of individuals from group A shown ads compared to group B .

A.2 Approximate Parity

Similar to above, we might loosen our definition to only require that bids on individuals be close to each other, i.e. for all individuals i and j , we have $|b_i - b_j| < \epsilon$ for some $\epsilon > 0$.

However, as in our example above, such a constraint can still result in outcomes that are far from proportional even for arbitrarily small values of ϵ . To see how this can occur, consider our example from above where all individuals in A and B have a winning bid of ϵ . One potential strategy in this setting would be to bid w on all individuals from A and $w - \epsilon$ on all individuals from B . This results in a strategy that satisfies approximate bid parity constraints, but never shows an ad to an individual from B .

B Algorithm for Solving the Linear Program

The bidding algorithm from [1] can be extended to work with additional group representation constraints. In this section, we explain the bidding algorithm and prove its correctness when there are additional representation constraints.

In the algorithm δ is the approximation parameter, \mathcal{V} is an upper bound on the objective and V_B, V_G are bounds on the value of the budget and group representation constraints.

At a high level, the algorithm iterates over all possible objective values V , and for each value tries to solve the following problem: “is there an x that satisfies the constraints and has utility V ?”. This problem can be equivalently restated in matrix form, to ask whether there is an x such that $Ax \geq u$ for the values of A, u described in the algorithm. We use the multiplicative weights algorithm to solve each of these sub-problems. In the update step, the problem is reduced to a problem in 1-dimension: “is there an x such that $p^T Ax \geq p^T u$, where p is the weights vector?”. For the 1-dimensional problem, the optimal threshold described on Section 4.3 is an optimal solution, and therefore can be used for the update.

■ **Algorithm 3** Finding the optimal strategy.

Input: $\delta > 0, \mathcal{V}, V_B, V_g \forall g \in G$
Output: $\hat{x}_1, \dots, \hat{x}_n \in \{0, 1\}$

/* \mathcal{V}, V_g, V_B , are bounds on the objective value and constraints violations. */
 $T_1 \leftarrow c/\delta$;
 $T_2 \leftarrow c/\delta^3$;
 $\hat{x} \leftarrow 0^n$;// output init
for $i = 1, \dots, T_1$ **do**
 $V \leftarrow i\delta\mathcal{V}$; // V is the current objective we are trying to reach.
 $A \leftarrow \begin{pmatrix} ctr_1 v_1/\mathcal{V} & ctr_2 v_2/\mathcal{V} & \dots & ctr_n v_n/\mathcal{V} \\ -ctr_1 c p c_1/V_B & \dots & \dots & -ctr_n c p c_n/V_B \\ ctr_1(g_1 - \mu_g)/V_g & \dots & \dots & ctr_n(g_n - \mu_g)/V_g \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}$; $u \leftarrow \begin{pmatrix} V/\mathcal{V} \\ -B/V_B \\ 0 \\ \vdots \end{pmatrix}$;
 /* MW algorithm solving: is there $x, 0 \leq x_l \leq 1$ such that $Ax \geq u$? */
 $FAIL \leftarrow 0$;
 $w \leftarrow 1^{2+|G|}$; // Initialize weights
 for $t = 1, \dots, T_2$ **do**
 /* Each iteration solving 1-dim problem: is there $x, 0 \leq x_l \leq 1$ such that
 $w^T Ax \geq w^T u$? */
 $\alpha \leftarrow \frac{w_2 \mathcal{V}}{V_B w_1}$;
 $\beta_g \leftarrow \frac{\mathcal{V} v_B w_g}{w_1 w_2 V_G}$; // for g the j 'th group, $w_g = w_{j+2}$
 $b(l) \leftarrow \frac{v_l - \sum_{g \in G} \beta_g (\mu_g - g_l)}{\alpha}$;
 $x_l^{(t)} \leftarrow \mathbf{1}(b(l) \geq c p c_l) \quad \forall l \in [n]$; // $x_l^{(t)}$ is the optimal solution to the 1-dim
 problem.
 if $w^T A x^{(t)} < w^T u$ **then**
 $FAIL \leftarrow 1$;
 else
 $w_j \leftarrow \begin{cases} w_j \cdot (1 - \epsilon)^{A_j x^{(t)} - u_j} & A_j x^{(t)} - u_j \geq 0 \\ w_j \cdot (1 + \epsilon)^{-A_j x^{(t)} + u_j} & A_j x^{(t)} - u_j < 0 \end{cases}, \quad \forall j \in [2 + |G|]$;
 if $FAIL = 0$ **then**
 $\hat{x} = \sum_{t=1}^{T_2} x^{(t)}$;

We state a more formal statement of Theorem 2 and prove it.

► **Lemma 7.** *Let P be a relaxed ad allocation linear program, (2). Let \mathcal{V} be an upper bound on the objective value of (2) and V_B, V_G be upper bounds on the amount of violation of the budget and representation constraints. Then for every $\delta > 0$, Algorithm 3 runs in time and $O(n^2/\delta^4|G|)$ and outputs a solution $x \in [0, 1]^n$ such that*

$$\begin{aligned} \sum_i x_i \text{ctr}_i v_i &\geq \mathbf{OPT} - \delta \mathcal{V} \\ \sum_i x_i \text{ctr}_i \text{cpc}_i &\leq B + \delta V_B \\ \sum_i x_i \text{ctr}_i (\mu_g - g_i) &\leq \delta V_G \quad \forall g \in G. \end{aligned}$$

Proof. To prove the correctness of the algorithm, it is enough to prove that the $x_i^{(t)}$ assigned is indeed the optimal solution for the 1-dimensional problem. The rest is implied from the correctness of the multiplicative weight algorithm, see [2]. Therefore, we prove that $x_i^{(t)}$ is the optimal solution to the 1-dimensional problem $\max_x \{w^T Ax - w^T u\}$.

$$\begin{aligned} w^T Ax - w^T u &= w_1 \sum_{l=1}^n \frac{v_l}{\mathcal{V}} x_l - w_2 \sum_{l=1}^n \frac{\text{ctr}_l \text{cpc}_l}{V_B} x_l \\ &\quad + \sum_{j=3}^{|G|+2} w_j \sum_{l=1}^n \frac{\text{ctr}_l (g_l - \mu_g)}{V_g} x_l - w_1 \frac{V}{\mathcal{V}} + w_2 \frac{B}{V_B} \\ &= \sum_{l=1}^n x_l \left(w_1 \frac{v_l}{\mathcal{V}} - w_2 \frac{\text{ctr}_l \text{cpc}_l}{V_B} + \sum_{j=3}^{|G|+2} w_j \frac{\text{ctr}_l (g_l - \mu_g)}{V_g} \right) - w_1 \frac{V}{\mathcal{V}} + w_2 \frac{B}{V_B}. \end{aligned}$$

Denote $C_l = w_1 \frac{v_l}{\mathcal{V}} - w_2 \frac{\text{ctr}_l \text{cpc}_l}{V_B} + \sum_{j=3}^{|G|+2} w_j \frac{\text{ctr}_l (g_l - \mu_g)}{V_g}$. The maximal value of $w^T Ax - w^T u$ is given when in every l such that $C_l \geq 0$ we have $x_l = 1$, and for the rest we have $x_l = 0$.

Notice that after switching α, β_g (which we can think about just as renaming of w_j) we have that $c_l \geq 0$ is equivalent to $b(l) \geq \text{cpc}_l$, as

$$\begin{aligned} C_l &= w_1 \frac{\text{ctr}_l v_l}{\mathcal{V}} - \frac{w_2}{V_B} \text{ctr}_l \text{cpc}_l + \sum_g \frac{w_g}{V_g} \text{ctr}_l (g_l - \mu_g) \geq 0 \iff \\ \text{cpc}_l &\leq \frac{V_B}{w_2} \left(\frac{w_1 v_l}{\mathcal{V}} + \sum_g \frac{w_g}{V_g} (g_l - \mu_g) \right) \end{aligned}$$

If we denote $\alpha = \frac{w_2}{V_B} \frac{\mathcal{V}}{w_1}$ and $\beta_g = \frac{\mathcal{V} v_B w_g}{w_1 w_2 V_G}$ then we have that this is the same as $b(l) \geq \text{cpc}_l$. The algorithm solves the 1-dimensional problem $(1/\delta^4)$ times, each takes $O(|G|n^2)$ time. ◀

C Proofs of Theorems and Lemmas

Proof of Theorem 4. First, suppose we have an i such that $x_i^* = 0$. By the slackness conditions of the LPs, this implies that $\delta_i = 0$. Substituting this fact into constraint 3 of the dual tells us that we must have

$$\alpha \text{ctr}_i \text{cpc}_i + \sum_{g \in G} \beta_g \text{ctr}_i (\mu_g - g_i) \geq \text{ctr}_i v_i$$

Rearranging the terms of this inequality (and assuming $\text{ctr}_i \neq 0$), we get

$$\text{cpc}_i \geq \frac{v_i - \sum_{g \in G} \beta_g (\mu_g - g_i)}{\alpha} = T_i$$

as desired. For the other direction, suppose that $x_i^* = 1$. Again applying complementary slackness, we know that constraint 3 must be tight, and thus

$$\delta_i + \alpha \text{ctr}_i \text{cpc}_i + \sum_{g \in G} \beta_g \text{ctr}_i (\mu_g - g_i) = \text{ctr}_i v_i.$$

Again rearranging to solve for cpc_i , we get:

$$\text{cpc}_i = \frac{v_i - \sum_{g \in G} \beta_g (\mu_g - g_i)}{\alpha} - \frac{\delta_i}{\text{ctr}_i \alpha} = T_i - \frac{\delta_i}{\text{ctr}_i \alpha}.$$

We can conclude that this guarantees $T_i \geq \text{cpc}_i$, and if $\delta_i > 0$, then $T_i > \text{cpc}_i$. \blacktriangleleft

Proof of Lemma 5. Given a γ -flexible solution x , let y be the output of Algorithm 1, and let $x' \in [0, 1]^n$ be as in Algorithm 1. We show that all of the constraints hold with high probability. We denote the realization of clicks from each individual as r , i.e. $r_i \sim \text{Ber}(\text{ctr}_i)$.

For the budget constraint, we show that (5) holds with high probability,

$$\Pr_{y,r} \left[\sum_{i \in [n]} y_i r_i \text{cpc}_i \geq B \right] \leq \Pr_{y,r} \left[\sum_{i \in S} y_i r_i \text{cpc}_i \geq \left(1 + \frac{\epsilon}{2}\right) \sum_{i \in S} x'_i \text{ctr}_i \text{cpc}_i \right] \leq e^{-\epsilon^2 \gamma^2 (1 - \frac{\epsilon}{2})^2 \frac{n}{4}},$$

where the last inequality is due to Hoeffding's inequality.

For the representation constraints, (6), we have that for every $g \in G$,

$$\Pr_{y_i, r_i} \left[\sum_{i \in [n]} g_i y_i r_i \leq \left(1 - \frac{\epsilon}{2} - \frac{\gamma \epsilon}{4}\right) \sum_{i \in [n]} x_i \text{ctr}_i g_i \right] \quad (12)$$

$$\leq \Pr_{y_i, r_i} \left[\sum_{i \in [n]} g_i y_i r_i \leq \left(1 - \frac{\gamma \epsilon}{4}\right) \sum_{i \in [n]} x'_i \text{ctr}_i g_i \right] \leq e^{-\frac{\gamma^2 \epsilon^2}{32} \sum_{i \in [n]} g_i x_i \text{ctr}_i}. \quad (13)$$

$$\Pr_{y,r} \left[\sum_{i \in [n]} y_i r_i \geq \left(1 - \gamma \epsilon - (1 - \gamma) \frac{\epsilon}{2} + \frac{\gamma \epsilon}{4}\right) \sum_{i \in [n]} x_i \text{ctr}_i \right] \quad (14)$$

$$\leq \Pr_{y,r} \left[\sum_{i \in [n]} y_i r_i \geq \left(1 + \frac{\gamma \epsilon}{4}\right) \sum_{i \in [n]} x'_i \text{ctr}_i \right] \leq e^{-\frac{\gamma^2 \epsilon^2}{32} \sum_{i \in [n]} x_i \text{ctr}_i}. \quad (15)$$

The solution x satisfies the constraint up to a constant error of 2, so $\sum_{i \in [n]} g_i x_i \text{ctr}_i \geq 1/2 \cdot \mu_g \sum_{i \in [n]} x_i \text{ctr}_i$. Therefore the bound in both (12) and (14) is at most $\exp(-\gamma^3 \epsilon^2 \mu_g n)$. If the events in (12) and (14) do not hold, then the representation constraint on group g is satisfied, as we have that

$$\mu_g \sum_{i \in [n]} y_i r_i \leq \left(1 - \frac{\epsilon}{2} - \frac{\gamma \epsilon}{4}\right) \mu_g \sum_{i \in [n]} x_i \text{ctr}_i, \quad \sum_{i \in [n]} g_i y_i r_i \geq \left(1 - \frac{\epsilon}{2} - \frac{\gamma \epsilon}{4}\right) \sum_{i \in [n]} g_i x_i \text{ctr}_i$$

By union bound over all group representation constraints for $g \in G$ and over the budget constraint, with probability $1 - \exp(-\mu \gamma^2 \epsilon^2 n)$ all constraints hold.

We are left with showing that the objective is not reduced by much. We notice that $\forall i \in [n], x'_i \geq (1 - \epsilon)x_i$, so from the linearity of expectation we get (7). \blacktriangleleft

Proof of Lemma 6. Let $S = S_g$ for some $g \in G$. Let i_1, \dots, i_t be the order of the elements in S used by the algorithm. From (8), this order is also an order by cpc_i .

From the algorithm, we have that for every $j \in [t]$,

$$\sum_{l \geq j} x_{i_l} - 1 \leq \sum_{l \geq j} y_{i_l} \leq \sum_{l \geq j} x_{i_l}. \quad (16)$$

For the budget constraint, (9), we claim that for every $j \in [t]$,

$$\sum_{l \geq j} cpc_{i_l}(x_{i_l} - y_{i_l}) \geq cpc_{i_j} \sum_{l \geq j} (x_{i_l} - y_{i_l}). \quad (17)$$

We prove it by induction on j , starting from $j = t$. The basis is implied from (16). The step,

$$\begin{aligned} \sum_{l \geq j} cpc_{i_l}(x_{i_l} - y_{i_l}) &= cpc_{i_j}(x_{i_j} - y_{i_j}) + \sum_{l > j} cpc_{i_l}(x_{i_l} - y_{i_l}) \\ &\geq cpc_{i_l}(x_{i_l} - y_{i_l}) + cpc_{i_{j+1}} \sum_{l > j} (x_{i_l} - y_{i_l}) && \text{(Inductive step)} \\ &\geq cpc_{i_l}(x_{i_l} - y_{i_l}) + cpc_{i_j} \sum_{l > j} (x_{i_l} - y_{i_l}). \end{aligned}$$

Where in the last inequality we use the facts that $cpc_{i_j} \leq cpc_{i_{j+1}}$ and $\sum_{l > j} (x_{i_l} - y_{i_l}) \geq 0$. Applying (17) with $j = 1$ and using (16) implies that $\sum_{i \in S} y_i cpc_i \leq \sum_{i \in S} x_i cpc_i$, and in general $\sum_{i \in [n]} y_i cpc_i \leq \sum_{i \in [n]} x_i cpc_i$, proving (9).

For the representation constraint, we have from (16) that for every $g \in G$, $\sum_{i \in [n]} y_i g_i \geq \sum_{i \in [n]} x_i g_i - 1$. By summing up on all S , we get that $\sum_{i \in [n]} y_i \leq \sum_{i \in [n]} x_i$. Together with the fact that x satisfy the representation constraint we get

$$\sum_{i \in [n]} y_i g_i + 1 \geq \sum_{i \in [n]} x_i g_i \geq \mu_g \sum_{i \in [n]} x_i \geq \mu_g \sum_{i \in [n]} y_i.$$

Therefore, y satisfy (10) for every group g .

For the objective value, (11), we fix a set S and let i_1, \dots, i_t , be the order used in the algorithm. To simplify the proof, we “split” elements in S and divide their x_i in the following way: if we have $y_{i_j} = 1$ because $x_{i_j} + \sum_{l > j} (x_{i_l} - y_{i_l}) > 1$, then we split i_j to two elements i, i' with $x_i = 1 - \sum_{l > j} (x_{i_l} - y_{i_l})$ and $x_{i'} = x_{i_j} - x_i$. This “splitting” is for analysis only, and we abuse notation by denoting $S = \{i_1, \dots, i_t\}$ also after the splitting. After the splitting we have that if $y_{i_j} = 1$ then $\sum_{l \geq j} x_l = \sum_{l \geq j} y_l$.

Let $j_1, \dots, j_k \in [t]$ be the indices in which $y_j = 1$. We have that for every $m \in [k]$, $\sum_{l=j_m}^{j_{m+1}-1} x_l = 1$, and also $\sum_{l \geq j_k} x_l = 1$ and $\sum_{l < j_1} x_l < 1$. Therefore,

$$\begin{aligned} \sum_{l \in [t]} v_{i_l} x_{i_l} &= \sum_{l=1}^{j_1-1} v_{i_l} x_{i_l} + \sum_{l=j_1}^{j_2-1} v_{i_l} x_{i_l} + \dots + \sum_{l=j_m}^t v_{i_l} x_{i_l} \\ &\leq v_{i_{j_1}} \sum_{l=1}^{j_1-1} x_{i_l} + v_{i_{j_2}} \sum_{l=j_1}^{j_2-1} x_{i_l} + \dots + v_{i_t} \sum_{l=j_m}^t x_{i_l} && (v_i \text{ are increasing}) \\ &\leq v_{i_{j_1}} + v_{i_{j_2}} + \dots + v_{i_t} \leq v_{i_t} + \sum_{l \in [t]} y_{i_l} v_{i_l}, \end{aligned}$$

which proves (11). ◀

Multiplicative Metric Fairness Under Composition

Milan Mossé 

Department of Philosophy, University of California at Berkeley, CA, USA

Abstract

Dwork, Hardt, Pitassi, Reingold, & Zemel [6] introduced two notions of fairness, each of which is meant to formalize the notion of similar treatment for similarly qualified individuals. The first of these notions, which we call additive metric fairness, has received much attention in subsequent work studying the fairness of a system composed of classifiers which are fair when considered in isolation [3, 4, 7, 8, 12] and in work studying the relationship between fair treatment of individuals and fair treatment of groups [6, 7, 13]. Here, we extend these lines of research to the second, less-studied notion, which we call multiplicative metric fairness. In particular, we exactly characterize the fairness of conjunctions and disjunctions of multiplicative metric fair classifiers, and the extent to which a classifier which satisfies multiplicative metric fairness also treats groups fairly. This characterization reveals that whereas additive metric fairness becomes easier to satisfy when probabilities of acceptance are small, leading to unfairness under functional and group compositions, multiplicative metric fairness is better-behaved, due to its scale-invariance.

2012 ACM Subject Classification Mathematics of computing → Probability and statistics

Keywords and phrases algorithmic fairness, metric fairness, fairness under composition

Digital Object Identifier 10.4230/LIPIcs.FORC.2023.4

Acknowledgements Many thanks to Omer Reingold and Li-Yang Tan for their generous guidance and support with this project. Thanks to James Evershed, Wes Holliday, Niko Kolodny, Gabrielle Candès, and three anonymous reviewers for helpful comments.

1 Introduction

We study the fairness of a decision-maker, modeled as a *classifier* C , which takes as input an individual and outputs a label 1 or 0, each with some probability. For example, C could take as input an individual applying for a loan and output 1 if it decides that they will receive the loan and 0 if not, and C could have high likelihood of approving application of individuals with high credit scores and a low likelihood of approving applications of individuals with low credit scores.

One plausible constraint on a fair decision-maker requires that it treat similarly qualified individuals similarly. Dwork, Hardt, Pitassi, Reingold, & Zemel [6] introduced two notions of fairness, each meant to formalize this constraint. The first of these, additive metric fairness, has received much attention in subsequent work [3, 4, 7, 8, 12, 13]:

► **Definition 1** (Additive metric fairness). *Let \mathcal{U} denote a set of individuals. A classifier C is additive metric fair with respect to a metric $d : \mathcal{U} \times \mathcal{U} \rightarrow [0, 1]$ if for all $u, v \in \mathcal{U}$,*

$$|\Pr[C(u) = 1] - \Pr[C(v) = 1]| \leq d(u, v).$$

The difference in two individuals' treatment is modeled as the additive difference in their likelihoods of acceptance by the classifier, and the difference in their qualifications is given by a metric. Additive metric fairness thus requires that two individuals' difference in treatment not exceed their difference in qualifications. For example, where $\Pr[C(u) = 1]$ is the likelihood that the loan application of u is approved, $d(u, v)$ could be the normalized difference between the credit scores of u and v .

Additive metric fairness becomes easy to satisfy when the probabilities $\Pr[C(u) = 1]$ are small:



© Milan Mossé;

licensed under Creative Commons License CC-BY 4.0

4th Symposium on Foundations of Responsible Computing (FORC 2023).

Editor: Kunal Talwar; Article No. 4; pp. 4:1–4:11

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

4:2 Multiplicative Metric Fairness Under Composition

► **Example 2** (An unfair lottery). Suppose that every pair of individuals u, v differs in qualifications by at least some amount δ . Then provided that for all individuals u , the likelihood $\Pr[C(u) = 1]$ is at most some sufficiently small value ϵ , the classifier C will be additive metric fair:

$$\begin{aligned} |\Pr[C(u) = 1] - \Pr[C(v) = 1]| &\leq \max(\Pr[C(u) = 1], \Pr[C(v) = 1]) \\ &\leq \epsilon \leq \delta \leq d(u, v). \end{aligned}$$

For example, C could be a highly selective university, so that $C(u) = 1$ means that u is accepted; an investment with a low likelihood of return, so that $C(u) = 1$ means that u received a return on the investment; or a lottery, so that $C(u) = 1$ means that u had a winning lottery ticket.

As a result, additive metric fairness is compatible with the following kinds of unfairness:

► **Example 3** (Unfairness for groups). Suppose that there are two groups A and B of investors. If those in group B invested a cent more than those in group A , we may set $d(u, v) = .01$ for $u \in A$ and $v \in B$. A classifier C can satisfy additive metric fairness by giving those in group A no chance of receiving a return on their sizable investment while giving those in group B some sufficiently small chance ϵ of receiving a return on their similarly-sized investment. However, this is manifestly unfair to those in group A .

► **Example 4** (Unfairness under functional composition). Suppose u and v each apply to several universities C_1, \dots, C_k , such that at each university C_i , the likelihood that u is accepted is 0 while $\Pr[C_i(v) = 1] = \epsilon$. Then the likelihood that v is accepted by at least one university may approach 1, while u has no chance of acceptance at any university. Even if the universities satisfy additive metric fairness when considered in isolation, because the likelihoods of acceptance are sufficiently small, they compose to create system which fails to treat similarly qualified applicants similarly.

Thus additive metric fairness is easier to satisfy when probabilities of acceptance are small, and this can lead to unfairness for groups and under functional composition. In this paper we find that the second, scale-invariant notion of fairness introduced by Dwork, Hardt, Pitassi, Reingold, & Zemel, *multiplicative metric fairness*, is better-behaved in its treatment of groups and under functional composition:

► **Definition 5** (Multiplicative metric fairness). *A classifier C is multiplicative metric fair with respect to a metric $d : \mathcal{U} \times \mathcal{U} \rightarrow \mathbb{R}^{\geq 0}$ if for all $u, v \in \mathcal{U}$,*

$$\Pr[C(u) = 1] \leq \Pr[C(v) = 1] \cdot \exp(d(u, v)).$$

Multiplicative metric fairness models the difference in treatment between two individuals not as an additive difference, but as a ratio; it does not become easy to satisfy when probabilities are small. In order to state our results, we now introduce the relevant notions of group fairness and of fairness under functional composition.

Group fairness

We propose the following notion of group fairness:

► **Definition 6** (Geometric Metric Fairness). *Fix a collection of protected attributes $\mathcal{A} \subseteq 2^{\mathcal{U}}$ (e.g. races, ages, genders, etc.). A classifier C satisfies geometric metric fairness with respect to a metric $d : \mathcal{A} \times \mathcal{A} \rightarrow \mathbb{R}^{\geq 0}$ when for all $A, B \in \mathcal{A}$,*

$$p_{\Pi}(A) \leq p_{\Pi}(B) \cdot \exp(d(A, B)),$$

where $p_{\Pi}(A) = \prod_{u \in A} \Pr[C(u) = 1]^{1/|A|}$ is the geometric mean likelihood of acceptance.

In the above definition, a metric quantifies differences in qualifications between groups, just as in Definitions 1 and 5, a metric quantifies differences in qualifications between individuals. For example, suppose that every job applicant $u \in A$ can be paired with some unique applicant $v \in B$ who is equally qualified, and vice versa. Then even if individual applicants within each group differ in their qualifications, there is no difference in qualifications between the groups: $d(A, B) = 0$. In this case, geometric metric fairness amounts to the constraint that

$$p_{\Pi}(A) = p_{\Pi}(B).$$

This contrasts with a well-studied notion of group fairness:

► **Definition 7 (Conditional Parity).** Fix $\mathcal{Q} \subseteq 2^{\mathcal{U}}$ and a collection of protected attributes $\mathcal{A} \subseteq 2^{\mathcal{U}}$ (e.g. races, ages, genders, etc.). A classifier C satisfies conditional parity if for all $Q \in \mathcal{Q}$, $A, B \in \mathcal{A}$,

$$p_{\Sigma}(A \cap Q) = p_{\Sigma}(B \cap Q)$$

where $p_{\Sigma} = \frac{1}{|A \cap Q|} \sum_{u \in A \cap Q} \Pr[C(u) = 1]$ is the arithmetic mean likelihood of acceptance.

Conditional parity was introduced by Ritov, Sun, & Zhao [14] and plays a central role in Dwork & Ilvento's study of fairness under composition [7]. Conditional parity generalizes other group notions of fairness. For example, one recovers *parity* by setting $\mathcal{Q} = \{\mathcal{U}\}$; one recovers *equalized odds* by setting $\mathcal{Q} = \{\{u : Y(u) = y\} : y \in \{0, 1\}\}$, where $Y(u)$ denotes the true label of u ; and one recovers *equal opportunity* by setting $\mathcal{Q} = \{\{u : Y(u) = 1\}\}$ [11, 15]. In general, we think of \mathcal{Q} as a collection of sets of individuals who are similarly qualified for the purposes of classification.

Plausibly, one should not be able to “make up for” mistreatment of some individuals within a group by treating other individuals within the group better; a radical departure from the mean treatment for any sub-group should register as unfair. However, because conditional parity only constrains the arithmetic mean probability of acceptance across members of a group, it allows for large variance in treatment of individuals within a group. In 2010, this feature of the arithmetic mean led the United Nations to change its way of calculating the Human Development Index (HDI):

In 2010, the geometric mean was introduced to compute the HDI [which was previously computed with the arithmetic mean]. Poor performance in any dimension is directly reflected in the geometric mean. In other words, a low achievement in one dimension is not linearly compensated for by a higher achievement in another dimension. The geometric mean reduces the level of substitutability between dimensions and at the same time ensures that a 1 percent decline in the index of, say, life expectancy has the same impact on the HDI as a 1 percent decline in the education or income index. Thus, as a basis for comparisons of achievements, this method is also more respectful of the intrinsic differences across the dimensions than a simple average.

Just as the geometric mean index value is thought to better respect differences and non-substitutability across the dimensions of the HDI, the geometric mean likelihood of acceptance across a group might be thought to also better respect differences and non-substitutability across individuals within a group; this motivates the constraint of geometric metric fairness.¹

¹ When any factor of the geometric mean is 0, of course the geometric mean is itself 0, and it becomes trivial to ensure geometric metric fairness; one merely has to assign $\Pr[C(u) = 1] = \Pr[C(v) = 1] = 0$ for one person $u \in A$ and another person $v \in B$. For this reason, the geometric mean (and the associated definition of fairness) is most meaningful when probabilities are nonzero.

Fairness under functional composition

We focus on the two kinds of functional composition introduced (with the following examples) by Dwork & Ilvento [7]:

- *AND*. Suppose that, considered in isolation from one another, a university's admissions and financial aid committees treat every pair of similarly qualified prospective students similarly. To what degree do similarly qualified students have similar likelihoods of receiving admission *and* financial aid offers?
- *OR*. Suppose that, considered in isolation from one another, several universities' admissions committees treat every pair of similarly qualified prospective students similarly. To what degree do two similarly qualified individuals have very different overall likelihoods of being accepted by at least one university?

More formally, we can define the AND and OR compositions of several classifiers:

► **Definition 8.** Fix classifiers C_1, \dots, C_k . Where u is an individual to be classified, define the classifiers

$$C_{AND}(u) = \bigwedge_{i \in [k]} C_i(u)$$

$$C_{OR}(u) = \bigvee_{i \in [k]} C_i(u).$$

In other words, C_{AND} accepts individual u if and only if each of C_1, \dots, C_k accepts u , and C_{OR} accepts u if and only if at least one of the classifiers C_1, \dots, C_k accepts u .

Supposing C_i is (additive or multiplicative) metric fair with respect to $d_i(u, v)$ for $i \in [k]$, in fairness under functional composition, we ask: With respect to what metric are C_{AND} and C_{OR} (additive or multiplicative) metric fair?

Our results

Having introduced the relevant definitions, we can state the paper's results:

► **Theorem 9 (Groups).** If C is multiplicative metric fair with respect to d , then it is geometric metric fair with respect to $\text{EMD}_d(A, B)$, the earth-mover distance between uniform distributions on A and B , with $d(u, v)$ giving the cost of moving a unit of probability from u to v . Further, this is tight: for any metric d on \mathcal{U} , there exists a classifier C which is multiplicative metric fair with respect to d and for which

$$p_{\Pi}(A) = p_{\Pi}(B) \cdot \exp(\text{EMD}_d(A, B)).$$

► **Theorem 10 (Functions).** If C_i is multiplicative metric fair with respect to $d_i(u, v)$ for $i \in [k]$, then:

- C_{AND} is multiplicative metric fair with respect to $d_{\Sigma}(u, v) = \sum_i d_i(u, v)$.
- C_{OR} is multiplicative metric fair with respect to $d_{\max}(u, v) = \max_i d_i(u, v)$.

Further, these results are tight: for each of the above forms of composition and for any choices of d_i for $i \in [k]$, there exist classifiers C_i for $i \in [k]$ which are multiplicative metric fair with respect to d_i and whose composition is multiplicative metric fair with respect to no metric smaller than the one stated above.

The rest of the paper is organized as follows. §1.1 discusses related work. §1.2 summarizes the notation used in the paper. §2 discusses the relationship between notions of metric fairness and the above notions of group fairness, proving the paper's first main result. §3 discusses how notions of metric fairness behave under functional composition, proving the paper's second main result.

1.1 Related work

Several recent works in algorithmic fairness studies how the fairness of classifiers locally relates to that of the classifiers' global behavior composed over many decisions, or to that of a classifier that in some way composes the decisions of the individual classifiers [6, 2, 7, 9, 10, 8, 12, 3, 4].

The need for work in this area is underscored by the fact that in practice, classifiers are often trained separately and without communication, so that any guarantees on their global behavior must rest solely on the decisions the designers of the classifiers are able to make in isolation. In a recent survey of work on fairness in machine learning, Chouldechova and Roth make exactly this point, calling for work exploring fairness under composition [5].²

Experience from differential privacy suggests that graceful degradation under composition is key to designing complicated algorithms satisfying desirable statistical properties, because it allows algorithm design and analysis to be modular. Thus, it seems important to find satisfying fairness definitions and richer frameworks that behave well under composition.

Much recent work on fairness under composition focuses in particular on the behavior of additive metric fairness under various kinds of composition [7, 8, 12, 3]. There are two papers which relate especially closely to this one. The first, written by Dwork, Hardt, Pitassi, Reingold, & Zemel [6], introduced the notion of additive metric fairness and characterized its relation to conditional parity. The second, written by Dwork & Ilvento [7], introduced the kinds of functional composition (AND and OR) studied in this paper and made progress in showing that additive metric fairness is not always well-behaved under these kinds of composition; we summarize some of this work in §3.1. Our results are meant to complement this line of research, by showing that multiplicative metric fairness is better-behaved in treatment of groups and under functional composition.

1.2 Notation

A classifier $C : \mathcal{U} \times \{0, 1\}^* \rightarrow \{0, 1\}$ is a (possibly randomized) Boolean-valued map, defined on a universe \mathcal{U} of individuals $u \in \mathcal{U}$. We denote by $1 - C$ the classifier C' which accepts precisely the individuals rejected by C : $C'(u) := \neg C(u)$. We say that a classifier *accepts* an individual when it assigns them a label of 1 and *rejects* an individual when it assigns them a label of 0. Throughout, d is a metric on \mathcal{U} . For classifiers C_1, \dots, C_k , we use d_1, \dots, d_k to denote their corresponding metrics. We define $d_\Sigma(u, v) = \sum_i d_i(u, v)$ and $d_{\max}(u, v) = \max_i d_i(u, v)$.

Subsets $A, B \subseteq \mathcal{U}$ denote protected groups. For a classifier C , we denote by $p_\Pi(A)$ the geometric mean likelihood of acceptance and by $p_\Sigma(A)$ the arithmetic mean likelihood of acceptance. When there are multiple classifiers C_1, \dots, C_k , we assume the randomness of the classifiers C_i to be mutually independent, and we use $p_i(u)$ to denote $\Pr[C_i(u) = 1]$. We define $p_{AND}(u) = \Pr[C_{AND}(u) = 1]$ and $p_{OR}(u) = \Pr[C_{OR}(u) = 1]$, where C_{AND} is the AND composition and C_{OR} the OR composition of some classifiers C_1, \dots, C_k .

² Dwork & Ilvento [7] point out an important difference between differential privacy and fairness under composition: "Comparing functional composition to differential privacy, it is important to understand that each component satisfying individual fairness separately (and for different metrics) is not analogous to the composition properties of differential privacy. With differential privacy, we assume a single privacy loss random variable which evolves gracefully with each release of information, increasing in expectation over time. However, with fairness, we may see that fairness loss increases or decreases (depending on the number and type of compositions) in idiosyncratic ways. Moreover, we may need to simultaneously satisfy many different task-specific 'fairness budgets,' and a bounded increase in distance based on one task may be catastrophically large for another."

2 Treatment of groups

The relation of (conditional) parity to additive metric fairness has garnered recent interest [1, 6].³ Dwork, Hardt, Pitassi, Reingold, & Zemel [6] give a tight characterization of the relationship between additive metric fairness and parity, using the following notion of earth-mover's distance:

► **Definition 11** (Earth-Mover's Distance). *Fix sets $A, B \subseteq \mathcal{U}$ and a collection of associated costs $d(u, v) \geq 0$ for each $u \in A, v \in B$. The earth-mover's distance $\text{EMD}_d(A, B)$ is the minimum amount of work required to transform a uniform distribution on A into one on B , where the amount of work required to move a unit of probability from individual u to individual v is given by $d(u, v)$. Formally,*

$$\text{EMD}_d(A, B) = \sum_{u \in A, v \in B} f_{u,v} \cdot d(u, v),$$

where the variables $f_{u,v}$ give an optimal solution to the following linear program (LP):

$$\begin{aligned} \min \quad & \sum_{u \in A, v \in B} f_{u,v} \cdot d(u, v) \\ & f_{u,v} \geq 0 \\ & f_{u,v} = 0 \text{ if } u \notin A \text{ or } v \notin B \\ & \sum_{v \in B} f_{u,v} = \frac{1}{|A|}, \sum_{u \in A} f_{u,v} = \frac{1}{|B|}, \sum_{u \in A, v \in B} f_{u,v} = 1 \end{aligned}$$

Dwork, Hardt, Pitassi, Reingold, & Zemel [6] prove the following by LP duality, applied to the LP in Definition 11:

► **Theorem 12.** *If C is additive metric fair with respect to d , then for all $A, B \subseteq \mathcal{U}$,*

$$|p_\Sigma(A) - p_\Sigma(B)| \leq \text{EMD}_d(A, B).$$

Further, this is tight: for all metrics d and choices of A, B , there exists a classifier C that is additive metric fair with respect to d , such that the above inequality is an equality.

The above result says that if C is additive metric fair, then the earth-mover distance gives a tight characterization of the extent to which C satisfies conditional parity. The same authors observe that an identical upper bound holds if we instead assume that C and $1 - C$ are multiplicative metric fair:

► **Corollary 13.** *If C and $1 - C$ are multiplicative metric fair with respect to d , then for all $A, B \subseteq \mathcal{U}$,*

$$|p_\Sigma(A) - p_\Sigma(B)| \leq \text{EMD}_d(A, B).$$

³ We observe that there is no deep difference between parity (i.e. conditional parity where $\mathcal{Q} = \mathcal{U}$) and conditional parity. It is clear that conditional parity is a generalization of parity. Conversely, conditional parity is a version of parity where we stipulate that A and B are equally qualified.

Thus the relationship between additive metric fairness and parity is well-understood, and it is known already that multiplicative metric fairness performs “at least as well” as additive metric fairness, in the sense that one can only get closer to satisfying parity in the multiplicative case.

However, because Theorem 12 only provides a bound on the difference between *arithmetic mean* conditional probabilities of acceptance, the guarantee can still hold when sub-groups are treated very differently, so long as advantages and disadvantages of different sub-groups are traded off in a way that maintains conditional parity. We now show Theorem 9, according to which multiplicative metric fairness, in contrast to additive metric fairness, provides a bound on the ratio of the geometric mean probabilities of acceptance:

Proof. We first show the upper bound and next show the lower bound. Fix a classifier C which is multiplicative metric fair with respect to d , and fix any flow $\{f_{u,v}\}_{u \in A, v \in B}$ solving the earth-mover LP. Using the multiplicative metric fairness constraint, note that for all $u \in A, v \in B$, we have

$$\Pr[C(u) = 1]^{f_{u,v}} \leq e^{d(u,v) \cdot f_{u,v}} \cdot \Pr[C(v) = 1]^{f_{u,v}}.$$

Taking the product on both sides over all $u \in A, v \in B$ gives

$$\prod_{u \in A} \Pr[C(u) = 1]^{\sum_{v \in B} f_{u,v}} \leq \prod_{u \in A, v \in B} e^{d(u,v) \cdot f_{u,v}} \cdot \prod_{v \in B} \Pr[C(v) = 1]^{\sum_{u \in A} f_{u,v}}.$$

Note that for $u \in A$, we have $\sum_{v \in B} f_{u,v} = 1/|A|$, while for $v \in B$, we have $\sum_{u \in A} f_{u,v} = 1/|B|$. Thus

$$p_{\Pi}(A) \leq \exp\{\text{EMD}_d(A, B)\} \cdot p_{\Pi}(B)$$

Now, we show the lower bound. Fix any metric d . Let c be a constant with $1 \leq c$ such that $d(u, v) \leq c$ for all $u, v \in \mathcal{U}$. Define the metric $d'(u, v) = d(u, v)/c$, so that $d'(u, v) \in [0, 1]$ for all $u, v \in \mathcal{U}$. By Theorem 12, there exists a classifier C' which is additive metric fair with respect to d' and for which

$$|p'_{\Sigma}(A) - p'_{\Sigma}(B)| = \text{EMD}_{d'}(A, B). \quad (1)$$

Define C by

$$\Pr[C(u) = 1] = \exp\{-\Pr[C'(u) = 1] \cdot c\}.$$

Because C' is additive metric fair with respect to d' , it follows that C is multiplicative metric fair with respect to d :

$$\ln \left[\frac{\Pr[C(u) = 1]}{\Pr[C(v) = 1]} \right] = -\Pr[C'(u) = 1] \cdot c + \Pr[C'(v) = 1] \cdot c \leq d'(u, v) \cdot c = d(u, v).$$

Suppose without loss of generality that $p'_{\Sigma}(B) \geq p'_{\Sigma}(A)$. Let us restate Equation 1 using the definition of C :

$$\begin{aligned} \frac{1}{c} \cdot \sum_{u \in B} \frac{-\ln \Pr[C(u) = 1]}{|B|} - \frac{1}{c} \cdot \sum_{v \in A} \frac{-\ln \Pr[C(v) = 1]}{|A|} &= \text{EMD}_{d'}(A, B) \\ &= \frac{\text{EMD}_d(A, B)}{c}. \end{aligned}$$

Eliminating the factor $1/c$ and making both sides the exponent of e , we obtain

$$\frac{\prod_{u \in B} \Pr[C(u) = 1]^{-1/|B|}}{\prod_{u \in A} \Pr[C(u) = 1]^{-1/|A|}} = \frac{p_{\Pi}(A)}{p_{\Pi}(B)} = \exp\{\text{EMD}_d(A, B)\},$$

as desired. ◀

3 Functional composition

We first overview known results for additive metric fairness under functional composition; this will serve to illustrate the contrast with multiplicative metric fairness.

3.1 Additive metric fairness under functional composition

Here, we rehearse known limitations and positive results for additive metric fairness of AND and OR compositions, with an eye to explaining some of the difficulties that arise.

We start with AND fairness. Given the following result, it is tempting to conjecture that C_{AND} is additive metric fair with respect to the maximum of the individual metrics:

► **Proposition 14** (Dwork & Ilvento [7]). *Fix nontrivial metrics d_1, d_2 and let d be any metric. If there exist $u, v \in \mathcal{U}$ such that*

- $d(u, v) \leq d_1(u, v), d_2(u, v)$, and
- $d_1(u, v), d_2(u, v) > 0$,

there exist C_1, C_2 , fair with respect to d_1, d_2 , such that C_{AND} is unfair with respect to d .

dxBut in fact even picking d_{\max} does not guarantee additive metric fairness:

► **Example 15.** Let C_1 and C_2 be copies of the same classifier: $p_i(u) = 1, p_i(v) = 1/2$, and $d_i(u, v) = 1/2$ for $i = 1, 2$. Then the classifiers C_i are individually additive metric fair with respect to $d_i(u, v)$, but their composition is not fair with respect to $d_{\max}(u, v) = \max_i d_i(u, v)$.

In a sense, when probabilities are small, the choice of metric for the AND composition in the additive case does not matter: as long as for each u , there exists some i with $p_i(u) \leq d(u, v)$, a fortiori $p_{AND}(u) \leq d(u, v)$. Since without loss of generality $p_{AND}(u) \geq p_{AND}(v)$, we have $|p_{AND}(u) - p_{AND}(v)| \leq d(u, v)$, giving fairness with respect to the arbitrary metric d . In other words, if probabilities are small enough, additive metric fairness for the AND composition trivializes.

We turn now to OR fairness. Dwork & Ilvento [7] observe that in the case of OR fairness, it is natural to suppose that the metrics are identical; returning to an earlier example, if the individual classifiers are admissions committees for different universities, it is natural to suppose that the admissions committees compare candidates using similar metrics. In this case the problem just discussed of picking a metric against which to compare the composition is more tractable: one can pick the composition metric to be the same as the metrics of the individual classifiers. Dwork & Ilvento's results imply the following:

► **Proposition 16** (Dwork & Ilvento [7]). *Fix classifiers C_1, \dots, C_k that are additive metric fair with respect to d . Consider two cases. If for all u , we have*

$$\Pr[C_{OR}(u) = 1] \geq \frac{1}{2},$$

then for any classifier C_{k+1} with $\Pr[C_{k+1}(u) \geq 1/2]$ for all $u \in \mathcal{U}$, the OR composition of C_1, \dots, C_{k+1} is additive metric fair with respect to d . If instead the above condition fails for some u, v with nontrivial distance ($d(u, v) > 0$), then there exist two classifiers C_{k+1}, C_{k+2} , additive metric fair with respect to d , such that the OR composition of C_1, \dots, C_{k+2} is not additive metric fair with respect to d .

In other words, the first, positive part of the above result says that if an initial collection of classifiers is more likely than not to accept every individual, adding a classifier that shares this property makes the entire collection's OR composition fair. The second, negative part

of the result says that if there are even two (nontrivially different) individuals the initial collection is more likely to reject than accept, one can add two fair classifiers that make the OR composition of the entire collection unfair. We earlier found that when the probabilities $p_i(u)$ are small enough, additive fairness for the AND composition trivializes; we now find that when the probabilities are small, we have no positive result for the additive metric fairness of the OR composition.

3.2 Multiplicative metric fairness under functional composition

We now show Theorem 10, which provides substantive fairness guarantees even when probabilities of acceptance are small:⁴

Proof. For $u, v \in \mathcal{U}$

$$p_{AND}(u) = \prod_i p_i(u) \leq \prod_i p_i(v) \cdot e^{\sum_i d_i(u,v)} = p_{AND}(v) \cdot e^{\sum_i d_i(u,v)}.$$

This shows that C_{AND} is multiplicative metric fair with respect to d_Σ . To see that the result is tight, one simply picks classifiers such that $p_i(u) = e^{d_i(u,v)} \cdot p_i(v)$, so that indeed $p_{AND}(u) = e^{d_\Sigma(u,v)} p_{AND}(v)$.

We next show that C_{OR} is multiplicative metric fair with respect to d_{\max} and show that this is tight. We only consider the case where $k = 2$, since iterating the argument then gives the result for general k . Suppose without loss of generality that $p_{OR}(u) \geq p_{OR}(v)$. By assumption C_1, C_2 are multiplicative metric fair with respect to d_{\max} , so it suffices to show the first inequality:

$$\frac{p_{OR}(u)}{p_{OR}(v)} \leq \max \left[\frac{p_1(u)}{p_1(v)}, \frac{p_2(u)}{p_2(v)} \right] \leq e^{d_{\max}(u,v)}.$$

To show the first inequality, we suppose $p_{OR}(u)/p_{OR}(v) > p_1(u)/p_1(v)$ and show that it follows that

$$\frac{p_{OR}(u)}{p_{OR}(v)} < \frac{p_2(u)}{p_2(v)}.$$

Noting that $p_{OR}(u) = p_1(u) + p_2(u) - p_1(u)p_2(u)$, let us rephrase $\frac{p_{OR}(u)}{p_{OR}(v)} > p_1(u)/p_1(v)$ after clearing denominators:

$$p_1(v)[p_1(u) + p_2(u) - p_1(u)p_2(u)] > p_1(u)[p_1(v) + p_2(v) - p_1(v)p_2(v)].$$

After removing $p_1(v)p_1(u)$ from both sides and factoring, the above says that

$$p_2(u)p_1(v)(1 - p_1(u)) > p_2(v)p_1(u)(1 - p_1(v)).$$

In other words,

$$\frac{p_2(u)}{p_2(v)} > \frac{p_1(u)}{p_1(v)} \cdot \frac{1 - p_1(v)}{1 - p_1(u)}.$$

⁴ Dwork, Hardt, Pitassi, Reingold & Zemel [6] introduce the constraint equivalent to multiplicative metric fairness for C and $1 - C$. Theorem 10 illustrates why this paper has separated their definition into two components: multiplicative metric fairness of $1 - C_i$ for $i \in [k]$ does not yield multiplicative metric fairness for $1 - C_{AND}$, where C_{AND} is the AND composition of C_1, \dots, C_k , but instead yields multiplicative metric fairness for the AND composition of $1 - C_1, \dots, 1 - C_k$.

4:10 Multiplicative Metric Fairness Under Composition

We will later show that $\frac{1-p_1(v)}{1-p_1(u)} \geq \frac{1-p_2(u)}{1-p_2(v)}$, but let us finish the proof on this assumption. Combining this with the above inequality gives

$$\frac{p_2(u)}{p_2(v)} > \frac{p_1(u)}{p_1(v)} \cdot \frac{1-p_2(u)}{1-p_2(v)}.$$

Clearing denominators, the above says that

$$p_2(u)[p_1(v) - p_1(v)p_2(v)] > p_2(v)[p_1(u) - p_1(u)p_2(u)]$$

Add $p_2(u)p_2(v)$ to both sides. Then the above says that

$$p_2(u)[p_1(v) + p_2(v) - p_1(v)p_2(v)] > p_2(v)[p_1(u) + p_2(u) - p_1(u)p_2(u)],$$

or in other words, $p_2(u)/p_2(v) > p_{OR}(u)/p_{OR}(v)$, as desired.

It remains for us to show that

$$\frac{1-p_1(v)}{1-p_1(u)} \geq \frac{1-p_2(u)}{1-p_2(v)}.$$

Since by assumption $p_{OR}(u) \geq p_{OR}(v)$, of course $1-p_{OR}(v) \geq 1-p_{OR}(u)$. Noting that $p_{OR}(v) = 1 - (1-p_1(v))(1-p_2(v))$, we can rephrase $1-p_{OR}(v) \geq 1-p_{OR}(u)$ as

$$(1-p_1(v))(1-p_2(v)) \geq (1-p_1(u))(1-p_2(u)) \iff \frac{1-p_1(v)}{1-p_1(u)} \geq \frac{1-p_2(u)}{1-p_2(v)}.$$

We now show that the result for OR is tight. It again suffices to consider the case for $k=2$. Fix any metric $d_1(u, v)$ and put $d_2(u, v) = d_1(u, v) - \alpha$ for an arbitrarily small $\alpha > 0$. We claim there exist classifiers C_1, C_2 such that:

- The classifiers C_1, C_2 are (respectively) multiplicative metric fair with respect to d_1, d_2 .
- C_{OR} is not multiplicative metric fair with respect to $d_2(u, v)$.

Let $\beta_1 \in (0, e^{-d_1(u, v)}], \beta_2 \in (0, e^{-d_2(u, v)})$ be parameters to be chosen later and define

$$p_1(u) = \beta_1 \cdot \exp[d_1(u, v)]$$

$$p_1(v) = \beta_1$$

$$p_2(u) = \beta_2 \cdot \exp[d_2(u, v)]$$

$$p_2(v) = \beta_2.$$

Then the classifiers C_1, C_2 defined by the above probabilities are multiplicative metric fair with respect to d_1, d_2 (respectively). We claim that for $\beta_2 < \frac{\beta_1 \cdot \alpha}{\exp[d_2(u, v)] \cdot (1-\beta_1)}$, we have

$$p_{OR}(u) \geq p_1(u) > \exp[d_2(u, v)] \cdot p_{OR}(v),$$

so that C_{OR} is indeed not multiplicative metric fair with respect to d_2 . It suffices to show the inequality on the right, which says that

$$p_1(u) > \exp[d_2(u, v)][\beta_1 + \beta_2 - \beta_1\beta_2] = \exp[d_2(u, v)] \cdot \beta_1 + \beta_2 \cdot \exp[d_2(u, v)] \cdot (1-\beta_1).$$

Subtracting $\exp[d_2(u, v)] \cdot \beta_1$ from both sides, this says that

$$\beta_1 \cdot \alpha = p_1(u) - \exp[d_2(u, v)] \cdot \beta_1 > \beta_2 \cdot \exp[d_2(u, v)] \cdot (1-\beta_1),$$

which holds by our choice of β_2 . ◀

References

- 1 Reuben Binns. On the apparent conflict between individual and group fairness. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 514–524, 2020.
- 2 Amanda Bower, Sarah N Kitchen, Laura Niss, Martin J Strauss, Alexander Vargas, and Suresh Venkatasubramanian. Fair pipelines. *arXiv preprint*, 2017. [arXiv:1707.00391](#).
- 3 Shuchi Chawla and Meena Jagadeesan. Fairness in ad auctions through inverse proportionality. *arXiv preprint*, 2020. [arXiv:2003.13966](#).
- 4 Shuchi Chawla, Rojin Rezvan, and Nathaniel Sauerberg. Individually-fair auctions for multi-slot sponsored search. In *3rd Symposium on Foundations of Responsible Computing*, page 1, 2022.
- 5 Alexandra Chouldechova and Aaron Roth. The frontiers of fairness in machine learning. *arXiv preprint*, 2018. [arXiv:1810.08810](#).
- 6 Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226, 2012.
- 7 Cynthia Dwork and Christina Ilvento. Fairness under composition. *arXiv preprint*, 2018. [arXiv:1806.06122](#).
- 8 Cynthia Dwork, Christina Ilvento, and Meena Jagadeesan. Individual fairness in pipelines. *arXiv preprint*, 2020. [arXiv:2004.05167](#).
- 9 Vitalii Emelianov, George Arvanitakis, Nicolas Gast, Krishna Gummadi, and Patrick Loiseau. The price of local fairness in multistage selection. *arXiv preprint*, 2019. [arXiv:1906.06613](#).
- 10 Swati Gupta and Vijay Kamble. Individual fairness in hindsight. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pages 805–806, 2019.
- 11 Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In *Advances in neural information processing systems*, pages 3315–3323, 2016.
- 12 Christina Ilvento, Meena Jagadeesan, and Shuchi Chawla. Multi-category fairness in sponsored search auctions. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 348–358, 2020.
- 13 Michael P Kim, Omer Reingold, and Guy N Rothblum. Fairness through computationally-bounded awareness. *arXiv preprint*, 2018. [arXiv:1803.03239](#).
- 14 Ya’acov Ritov, Yuekai Sun, and Ruofei Zhao. On conditional parity as a notion of non-discrimination in machine learning. *arXiv preprint*, 2017. [arXiv:1706.08519](#).
- 15 Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*, pages 962–970. PMLR, 2017.

Setting Fair Incentives to Maximize Improvement*

Saba Ahmadi ✉

Toyota Technological Institute at Chicago, IL, USA

Hedyeh Beyhaghi ✉

Carnegie Mellon University, Pittsburgh, PA, USA

Avrim Blum ✉

Toyota Technological Institute at Chicago, IL, USA

Keziah Naggita ✉

Toyota Technological Institute at Chicago, IL, USA

Abstract

We consider the problem of helping agents improve by setting goals. Given a set of target skill levels, we assume each agent will try to improve from their initial skill level to the closest target level within reach (or do nothing if no target level is within reach). We consider two models: the *common* improvement capacity model, where agents have the same limit on how much they can improve, and the *individualized* improvement capacity model, where agents have individualized limits. Our goal is to optimize the target levels for social welfare and fairness objectives, where *social welfare* is defined as the total amount of improvement, and we consider fairness objectives when the agents belong to different underlying populations. We prove algorithmic, learning, and structural results for each model.

A key technical challenge of this problem is the non-monotonicity of social welfare in the set of target levels, i.e., adding a new target level may decrease the total amount of improvement; agents who previously tried hard to reach a distant target now have a closer target to reach and hence improve less. This especially presents a challenge when considering multiple groups because optimizing target levels in isolation for each group and outputting the union may result in arbitrarily low improvement for a group, failing the fairness objective. Considering these properties, we provide algorithms for optimal and near-optimal improvement for both social welfare and fairness objectives. These algorithmic results work for both the common and individualized improvement capacity models. Furthermore, despite the non-monotonicity property and interference of the target levels, we show a placement of target levels exists that is approximately optimal for the social welfare of each group. Unlike the algorithmic results, this structural statement only holds in the common improvement capacity model, and we illustrate counterexamples of this result in the individualized improvement capacity model. Finally, we extend our algorithms to learning settings where we have only sample access to the initial skill levels of agents.

2012 ACM Subject Classification Theory of computation → Algorithmic mechanism design; Theory of computation → Machine learning theory

Keywords and phrases Algorithmic Fairness, Learning for Strategic Behavior, Incentivizing Improvement

Digital Object Identifier 10.4230/LIPIcs.FORC.2023.5

Related Version *Full Version*: <https://arxiv.org/pdf/2203.00134.pdf>

Funding This work was supported in part by the National Science Foundation under grants CCF-1733556 and CCF-1815011 and by the Simons Foundation under the Simons Collaboration on the Theory of Algorithmic Fairness.

* Authors are ordered alphabetically.



© Saba Ahmadi, Hedyeh Beyhaghi, Avrim Blum, and Keziah Naggita;
licensed under Creative Commons License CC-BY 4.0

4th Symposium on Foundations of Responsible Computing (FORC 2023).

Editor: Kunal Talwar; Article No. 5; pp. 5:1–5:22



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

One of the principal practices in policy making is setting reasonable expectations for the groups or individuals involved in the policy. Whether it is in the context of public policy, education, or career development, a too low expectation (i.e., an easy-to-achieve goal) may be a cause for not improving at one’s capacity, while a too high expectation (i.e., an out-of-reach goal) may be discouraging to even make an attempt. To accommodate different levels of participants’ abilities, the policy-maker may consider different levels of expectations and design goals at various levels so that all (or most of) the participants have a goal within their reach but not too easily achieved. At the other side from the policy-maker are the participants who may view their goal as a mere requirement for accessing other benefits, e.g., increasing their chances of promotion or gaining freedom to pursue other opportunities. In this case, the individuals may choose an easy-to-achieve goal rather than aim for maximal improvement. Examples of such expectations include reading goals for youth, language proficiency goals for applicants, outreach activities for employers, etc.

In this work, we consider the problem of helping agents improve by setting goals. Given a set of target “skill levels”, we assume each agent will try to improve from their initial skill level to the closest target level within reach (or do nothing if no target level is within reach). The designer’s goal is to maximize the total improvement both with and without fairness considerations.

Mathematically, we formulate this problem as follows. There are n agents belonging to g distinct groups. Agent i has an initial skill level, $p_i \in \mathbb{Z}_{\geq 0}$, and can increase their skill by at most Δ_i which is called its “improvement capacity”. Given a set of target levels $\mathcal{T} \subset \mathbb{Z}_{\geq 0}$, agent i improves to the closest target $\tau \in \mathcal{T}$ such that $\tau > p_i$ and $\tau \leq p_i + \Delta_i$ if such target exists; otherwise it stays at p_i .¹

This problem formulation gives rise to multiple challenges. First, optimizing improvement for a set of agents may conflict with another set. Consider a beginner-level agent (skill level B) and an intermediate-level (skill level I). Agent I finds any level up to τ_I within reach. Therefore, we need to design a project at level τ_I for this agent to improve maximally. On the other hand, B has the capacity to improve until τ_B , where $I < \tau_B < \tau_I$ – See Figure 1a. Now, consider both target levels τ_B and τ_I . Since agent I now has a closer target of τ_B , this agent no longer achieves its maximum improvement, and only reaches skill level τ_B . Secondly, there is non-monotonicity in the placement of target levels, i.e., adding a new target to the current placement may decrease the total amount of improvement. Consider a beginner-level (B) and an intermediate-level (I) agent and a target, τ , achievable by both agents – See Figure 1b. Designing a new project at level τ' between B and τ decreases the total amount of improvement since one agent (if $B < \tau' \leq I$) or both agents (if $I < \tau' < \tau$) switch from improving to τ to improving to τ' , which requires less improvement.



■ **Figure 1** Challenges in designing optimal target levels.

¹ We assume the policy-maker can disallow agent i from choosing a target $\tau \leq p_i$. For example, in designing reading goals for grade-school students, the fifth graders are not allowed to choose materials from the second-grade level, although the reverse is allowed. Setting the base for each agent at their true skill level is an abstraction of our mathematical model.

Main Results. In this work, we consider algorithmic, fairness, and learning-theoretic formulations, where a set of optimal target levels must be found in the presence of effort-bounded agents. We use *social welfare* as the notion of efficiency and define it as the total amount of improvement. Also, we define *social welfare for a given group* as the amount of improvement that group achieves. We consider two models: (1) the common improvement capacity model, where agents have the same limit Δ on how much they can improve, and (2) the individualized improvement capacity model, where agents have individualized limits Δ_i .

The main results of the paper are:

1. An efficient algorithm for placement of target levels to maximize social welfare. (Section 3)
2. An efficient algorithm for outputting the Pareto-optimal outcome for the social welfare of multiple groups. In particular, this can output the max-min fair solution that maximizes the minimum total improvement across groups. (Section 4)
3. A structural result on Pareto-optimal solutions: there exists a placement of target levels that simultaneously is approximately optimal for each group. More explicitly, when there are a constant number of groups, the total improvement for each group is a constant-factor approximation of the maximum improvement that we could provide that group if it were the only group under consideration. **This is our main contribution.** (Section 5)
4. An efficient learning algorithm for near-optimal placement of target levels. (Section 6)

The algorithmic results work for both the common and individualized improvement capacity models. However, the structural result only holds in the common improvement capacity model, and we illustrate examples where achieving any nontrivial fraction of optimal for all groups is not possible in the individualized capacity model.

Related work

Our work broadly falls under two general research areas: social welfare maximization in mechanism design and algorithmic fairness. Specifically, the closest topics to our paper are designing portfolios for consumers to minimize loss of returns [13], designing badges to steer users' behavior [4], and the literature on strategic classification.

Closest to our work is Diana et al.[13] who consider a model where each agent has a risk tolerance, observed as a real number, and must be assigned to a portfolio with risk lower than what they can tolerate. The goal of the mechanism designer is to design a small number of portfolios that minimizes the sum of the differences between the agent's risk tolerance and the risk of the portfolio they take; in other words, it minimizes the loss of returns. Since this is a minimization problem where each agent selects the closest target (portfolio) below their risk tolerance, adding any new target can only help with the objective function. Therefore, unlike our model, there is no conflict between targets, and the objective function is monotone in the set of targets.

Designing targets to incentivize agents to take specific actions is also a common feature of online communities and social media sites. In these platforms, there is a mechanism for rewarding user achievements based on a system of *badges* (similar to targets in our model) [4, 15, 5, 11, 12]. Among such papers, the closest to ours is Anderson et al.[4] who investigate how to optimally place badges in order to induce particular user behaviors, among other things. They consider a dynamic setting with a single user type interested in a particular distribution of actions and a mechanism designer whose objective is to set badges to motivate a different distribution of actions. Compared to our work, their model is more general in the sense that users can spend effort on different actions (improve in multiple dimensions), but also more specific, in the sense that there is only one user type; therefore, unlike our model there is no conflict between different users and adding more badges for the desired action always helps with steering the users in that direction (it is a monotone setting).

5:4 Setting Fair Incentives to Maximize Improvement

Another line of work that is relevant is strategic classification. In most cases, agents are fraudulently strategic, that is to say, game the decision-making model to get desired outcomes (see [18, 14, 20, 23, 1, 10, 16, 9, 26] among others). In other cases, in addition to actions only involving gaming the system, agents can also perform actions that truthfully change themselves to become truly qualified (see [21, 19, 3, 28, 22, 17, 7, 25, 8, 2] among others). In this paper, we assume agents only truthfully change themselves and, therefore, focus on incentivizing agents to improve as much as they can.

Within the combinatorial optimization literature, our work is related to the *uncapacitated facility location* (UFL) problem (see Chapter 4.5 of [27] for the problem definition). The main distinction between our problem and UFL is that the objective in UFL is to minimize the total distance traveled by the clients to reach their closest facility; whereas in our problem, the goal is to maximize the total distance traveled by agents to their closest target within reach.

Organization of the Paper

Section 2 formally introduces the general model settings and definitions used in the paper, and Section 3 provides an efficient algorithm for the problem of maximizing total improvement. Section 4 provides algorithms that output Pareto optimal solutions for groups' social welfare, including a solution that maximizes the minimum improvement per group. Section 5 provides an algorithm that finds the best simultaneously approximately optimal improvement per group and show it provides a constant approximation when the number of groups is constant. In Section 6, we provide efficient learning algorithms which generalize the previous results to a setting where there is only sample access to agents, and Appendix E provides further extensions to our main problems. All missing proofs are deferred to the appendix.

2 Model and Preliminaries

There are n agents $1, \dots, n$. Agent i is associated with two quantifiers: initial skill level, p_i , and *improvement capacity*, Δ_i , which determines the maximum amount agent i can improve its skill. For the majority of the paper, we assume p_i and Δ_i belong to $\mathbb{Z}_{\geq 0}$; however, some of our results hold more generally for real numbers. We consider two different models. The *common* and the *individualized* improvement capacity models. In the first model, all agents have the same improvement capacity, i.e., Δ_i are equal across agents; we substitute Δ_i with Δ in this case. The second model is a generalization where Δ_i may have different values. We use $\Delta_{max} = \max\{\Delta_1, \dots, \Delta_n\}$. Our solution is a finite set of target levels $\mathcal{T} \subset \mathbb{Z}_{\geq 0}$. We assume we are given a maximum number of allowed target levels k (if $k = n$, this is equivalent to allowing an unbounded number of target levels).

Agents behavior. Given target levels $\mathcal{T} \subset \mathbb{Z}_{\geq 0}$, agent i aims for the closest target above its initial skill if it can reach to that target given its improvement capacity. More formally, agent i aims for $\min\{\tau \in \mathcal{T} : p_i < \tau \leq p_i + \Delta_i\}$ if such τ exists and improves from p_i to τ . If no such target exists, agent i does not improve and its final skill level remains the same as the initial skill level p_i .

We use *social welfare* (SW) as our notion of efficiency and define it as the total amount of improvement of agents.

Groups and fairness notion. Each agent belongs to one of g distinct groups G_1, \dots, G_g . Given any set of target levels, the social welfare of group ℓ , SW_ℓ , is defined as the total amount of improvement for agents in that group.² We are interested in Pareto-optimal solutions for groups' social welfare. A solution \mathcal{T} is Pareto-optimal (is on the Pareto frontier) if there does not exist \mathcal{T}' in which all groups gain at least as much social welfare, and one group gains strictly higher. In particular, the Pareto frontier includes the max-min solution that maximizes the minimum social welfare across groups. In this paper, we focus on two natural fairness notions: one is the max-min solution described above, and the other is the notion of simultaneous approximate optimality given below.

► **Definition 1** (Simultaneous α -approximate optimality). *A solution with at most k targets is simultaneously approximately optimal for each group with approximation factor $0 \leq \alpha \leq 1$ if, for each group ℓ , the social welfare of group ℓ is at least an α fraction of the maximum social welfare achievable for group ℓ using at most k targets.*

Observation 2 determines the potential positions of the targets in an optimal solution.

► **Observation 2.** *Without loss of optimality, the targets in an optimal solution are either at positions $p_i + \Delta_i$ or p_i for some $i \in \{1, 2, \dots, n\}$. Consider a solution where target τ does not satisfy this condition. By shifting τ to the right as long as it does not cross $p_i + \Delta_i$ or p_i for any i , the total amount of improvement weakly increases: This transformation does not change the sets of agents that reach each target, and only increases the improvement of agents aiming for τ .*

► **Definition 3** (\mathcal{T}_p). *The set of potential optimal target levels, \mathcal{T}_p , is $\bigcup_{i=1}^n \{p_i, p_i + \Delta_i\}$.*

3 Maximizing Total Improvement

Algorithm 1 provides an efficient dynamic programming algorithm for finding a set of k target levels that maximizes total improvement for a collection of n agents. The recursion function $T(\tau, \kappa)$ finds the best set of at most κ target levels for agents on or to the right of τ . Recall that any target τ only affects the agents on its left, and agent i such that $p_i < \tau$ never selects $\tau' > \tau$ in presence of τ . In the recursive step (item 3 in Algorithm 1), $T(\tau, \kappa)$ is optimized by picking a target $\tau' > \tau$ that maximizes the total improvement achieved by τ' plus the total improvement achieved in the subproblem $T(\tau', \kappa - 1)$.

► **Algorithm 1.** *Run dynamic program based on function T , defined below, that takes $\cup_i \{p_i\}$ and k as input and outputs $T(\tau_{\min}, k)$, as the optimal improvement, and $S(\tau_{\min}, k)$, as the optimal set of targets; where $\tau_{\min} = \min\{\tau \in \mathcal{T}_p\}$ and $\tau_{\max} = \max\{\tau \in \mathcal{T}_p\}$. $T(\tau, \kappa)$ captures the maximum improvement possible for agents on or to the right of $\tau \in \mathcal{T}_p$ when at most κ target levels can be selected. Function T is defined as follows.*

- 1) For any $\tau \in \mathcal{T}_p$, $T(\tau, 0) = 0$.
- 2) For any $1 \leq \kappa \leq k$, $T(\tau_{\max}, \kappa) = 0$.
- 3) For any $\tau \in \mathcal{T}_p$, $\tau < \tau_{\max}$ and $1 \leq \kappa \leq k$:

$$T(\tau, \kappa) = \max_{\tau' \in \mathcal{T}_p \text{ s.t. } \tau' > \tau} \left(T(\tau', \kappa - 1) + \sum_{\tau \leq p_i < \tau' \text{ s.t. } \tau' - p_i \leq \Delta_i} (\tau' - p_i) \right)$$

$S(\tau, \kappa)$ keeps track of the optimal set of targets corresponding to $T(\tau, \kappa)$.

² Although the results are presented for the *total* improvement objective, they also hold for the *average* improvement objective.

► **Theorem 4.** *Algorithm 1 finds a set of targets that achieves the optimal social welfare (maximum total improvement) that is feasible using at most k targets given n agents. The algorithm runs in $\mathcal{O}(n^3)$.*

4 Pareto Optimality and Maximizing Minimum Improvement

Algorithm 2 provides a dynamic programming algorithm that constructs the Pareto frontier for groups' social welfare. By iterating through all Pareto-optimal solutions, we can find the solution that maximizes minimum improvement across all groups in pseudo-polynomial time. In contrast to Algorithm 1 where the algorithm only needs to store an optimal solution for each subproblem, here for each subproblem the algorithm stores a set containing *all* g -tuples of groups' improvements (I_1, I_2, \dots, I_g) that are simultaneously achievable for groups $\{G_1, \dots, G_g\}$.

► **Algorithm 2.** *Run dynamic program based on function T , defined below, that takes $\forall \ell \cup_{i \in G_\ell} \{p_i\}$ and k as input and outputs $T(\tau_{\min}, k)$, as the Pareto-frontier improvement tuples, and $S(\tau_{\min}, k)$, as the Pareto-frontier sets of targets ; where $\tau_{\min} = \min\{\tau \in \mathcal{T}_p\}$ and $\tau_{\max} = \max\{\tau \in \mathcal{T}_p\}$. $T(\tau, \kappa)$ constructs the Pareto frontier for groups' social welfare for agents on or to the right of $\tau \in \mathcal{T}_p$ when at most κ target levels can be selected. Function T is defined as follows.*

- 1) For any $\tau \in \mathcal{T}_p$, $T(\tau, 0) = \mathbf{0}_g$.
- 2) For any $1 \leq \kappa \leq k$, $T(\tau_{\max}, \kappa) = \mathbf{0}_g$.
- 3) For any $\tau \in \mathcal{T}_p, \tau < \tau_{\max}$ and $1 \leq \kappa \leq k$:

$$T(\tau, \kappa) = \left\{ \left(I_\ell + \left(\sum_{\substack{\tau \leq p_i < \tau' \\ \text{s.t. } \tau' - p_i \leq \Delta_i}} \mathbb{1}\{i \in G_\ell\}(\tau' - p_i) \right) \right)_{\ell=1}^g, \text{ s.t. } (I_\ell)_{\ell=1}^g \in T(\tau', \kappa - 1), \tau' \in \mathcal{T}_p, \tau' > \tau \right\}$$

All the dominated solutions are removed from $T(\tau, \kappa)$. $S(\tau, \kappa)$ stores the sets of targets corresponding to the improvement tuples in $T(\tau, \kappa)$.

► **Theorem 5.** *Algorithm 2 constructs the Pareto frontier for groups' social welfare using at most k targets given n agents in g groups. When all p_i, Δ_i values are integral, it has a time-complexity of $\mathcal{O}(n^{g+2}kg\Delta_{\max}^g)$, where Δ_{\max} is the maximum improvement capacity.*

► **Corollary 6.** *There is an efficient algorithm that finds a set of at most k targets that maximizes minimum improvement across all groups, i.e., maximizing $\min_{1 \leq \ell \leq g} SW_\ell$.*

The algorithm mentioned in Corollary 6 is pseudo-polytime since its time-complexity depends on the numeric value of Δ_{\max} . Appendix C provides a Fully Polynomial Time Approximation Scheme (FPTAS) to maximize the minimum improvement across all groups for the setting where each group G_ℓ has its own improvement capacity Δ_ℓ . In the FPTAS, we assume that p_i, Δ_i values are real numbers.

5 Simultaneous Approximate Optimality

In this section, we establish a structural result about the Pareto optimal solutions in the common improvement capacity model, and show there exists a simultaneously approximately optimal solution on the Pareto frontier, where the approximation factor depends on the number of groups. More specifically, given g groups, and limit $k \geq g$ on the number of target levels, we provide Algorithm 3 whose improvement per group is simultaneously an $\Omega(1/g^3)$ approximation of the optimal k -target solution for each group; implying a constant

approximation when the number of groups is constant. This result is of significance because natural outcomes such as the max-min fair solution and the union of group-optimal targets may lead to arbitrarily poor performance in terms of simultaneous approximate optimality – See Examples 10 and 11. This result only holds for the common improvement capacity model, and in Example 12, we show such a solution does not exist for the individualized improvement capacity model.

► **Theorem 7.** *Algorithm 3, given limit $k \geq g$ on the number of target levels, outputs a solution that is simultaneously $\Omega(1/g^3)$ -approximately optimal for each group, in the common improvement capacity model. More specifically, it provides a solution such that for all $1 \leq \ell \leq g$, $SW_\ell \geq 1/(16g^3)OPT_\ell^k$, where OPT_ℓ^k is the optimal social welfare of group ℓ using at most k target levels.*

► **Corollary 8.** *There is an efficient algorithm to find a simultaneously α^* -approximately optimal solution for each group in the common improvement capacity model, where α^* , defined as the best approximation factor possible, is $\Omega(1/g^3)$.*

We are not aware if $\Omega(1/g^3)$ is the best possible ratio; however, the following example shows there are no simultaneously approximately optimal solutions with approximation factor $> 1/g$.

► **Example 9.** Let $\Delta = 1$. Suppose group $\ell \in \{1, 2, \dots, g\}$ has a single agent at position $(\ell - 1)/g$; i.e., the agents are at $0, 1/g, \dots, (g - 1)/g$. For each group, the optimal total improvement is 1 in isolation (independent of the limit on the number of targets). However, using any number of targets in total there are no solutions with $> 1/g$ improvement for all groups.

The following example shows that the max-min fair solution does not satisfy a simultaneous constant approximation per group even when there are only two groups.

► **Example 10.** Let $\Delta = 1$. Group A has n agents; one agent at each position $1, 2, \dots, n$. Group B has n agents in k bundles of size n/k . The bundles of agents are at positions $n + 1 - k^2/n, \dots, n + k - k^2/n$. The unique max-min solution has targets at $n - k + 1, n - k + 2, \dots, n + 1$, and leads to k total improvement for each group which is k/n of the optimal total improvement for group B .

The following example shows solving the optimization problem separately per group and outputting the union of the targets can lead to arbitrarily low group improvement compared to the optimum.

► **Example 11.** Suppose there are two groups A and B and no limit on the number of targets. Group A has n agents at positions $1, 3, 5, \dots, 2n - 1$. Group B has n agents at positions $2 - \varepsilon, 4 - \varepsilon, \dots, 2n - \varepsilon$. First, consider the common capacity model, where $\Delta = 1$. In this case, the optimal solution for group A in isolation consists of targets at positions $\{2, 4, \dots, 2n\}$ and the optimal solution for group B in isolation is $\{3 - \varepsilon, 5 - \varepsilon, \dots, 2n + 1 - \varepsilon\}$. Now, consider a solution that is the union of the targets in the two separate solutions. Since each agent in group B is in ε proximity of a target from group A , the total improvement in group B is $n\varepsilon$. Therefore, the total improvement in group B can be arbitrarily close to 0. Next, consider the individualized capacity model, where agents in group A have $\Delta_A = 1$, and agents in group B have $\Delta_B = 1 + 2\varepsilon$. The optimal set of targets in isolation for group A is $\{2, 4, \dots, 2n\}$, and for group B is $\{3 + \varepsilon, 5 + \varepsilon, \dots, 2n + 1 + \varepsilon\}$. The union of these solutions result in $1 + (n - 1)\varepsilon$ for group A , and $n\varepsilon$ for group B which are arbitrarily low compared to the optimum, which is simultaneously $\geq n(1 - \varepsilon)$ for group A and $\geq n$ for group B .

The following example shows that if agents can improve by different amounts (the individualized improvement capacity model), then no approximation factor only as a function of g of optimal improvement per group is possible.

► **Example 12.** Suppose groups A and B each have a single agent at position 0. The agent in group A has improvement capacity $\Delta_A = \varepsilon$ and the agent in group B has improvement capacity $\Delta_B = 1$. The optimal total improvement in isolation for group A is ε , and for group B is 1. However, when considering both groups, no placement of targets with positive improvement for group A leads to $> \varepsilon$ improvement for group B .

First, we describe a high-level overview of Algorithm 3 that proceeds in four main steps.

1. **Optimal targets in isolation.** Run Algorithm 1 separately for each group to find an optimal allocation of at most $\lceil k/g \rceil$ targets³. Let \mathcal{T}_ℓ be the output for group ℓ .
2. **Distant targets in isolation.** Delete 3/4 fraction of each set of target levels, \mathcal{T}_ℓ , such that (1) the distance between every two consecutive targets in each set is at least 2Δ and (2) the new \mathcal{T}_ℓ (after deletion) guarantees an $\Omega(1)$ approximation of the previous step when the targets for each group are considered in isolation. Section 5.2 below shows this is possible.
3. **Locally optimized distant targets in isolation.** For each ℓ and $\tau \in \mathcal{T}_\ell$, consider the agents in group ℓ that afford to reach τ (agents in $G_\ell \cap [\tau - \Delta, \tau)$). Optimize τ to maximize the total improvement for this set of agents.
4. **Resolve interference of targets.** Consider sets of interfering targets. Relocate these targets locally to guarantee $\Omega(1/g^2)$ approximation per group compared to the previous step where each group was considered in isolation. Section 5.4 below shows this is possible.

5.1 Step 1: Optimal targets in isolation

At the end of step 1, \mathcal{T}_ℓ is the optimal set of targets for G_ℓ in isolation. The following observation shows that without loss of optimality, we may assume the distance between every other target level is at least Δ .⁴

► **Observation 13.** Consider a set of target levels $\mathcal{T} : \tau_1 < \tau_2 < \dots$. Suppose $\tau_{j+2} < \tau_j + \Delta$. By removing τ_{j+1} , any agent with $\tau_j \leq p_i < \tau_{j+1}$ improves strictly more, and other agents improve the same amount. This weakly increases social welfare.

5.2 Step 2: Distant targets in isolation

Step 2 of the algorithm runs the following procedure for \mathcal{T}_ℓ .

- **Definition 14** (Distant targets procedure). Consider solution $\mathcal{T} : \tau_1 < \tau_2 < \dots$, where for all j , $\tau_{j+2} - \tau_j \geq \Delta$ as input to the following procedure.
- Partition \mathcal{T} into 4 parts, P_1, P_2, P_3, P_4 , where $P_i =: \tau_i, \tau_{4+i}, \tau_{8+i}, \dots$. Consider the part P_i that introduces the highest improvement. Update \mathcal{T} to P_i (and delete the rest).

Lemma 15 shows that at the end of this step, target levels in \mathcal{T}_ℓ are 2Δ apart, this step provides a 4-approximation compared to the previous step, and the number of targets designated to each group is at most $\lceil k/g \rceil$.

³ Although the total number of targets used in this step can be more than k , after the algorithm ends at most k targets are being used in total.

⁴ Example 30, however, shows the distance between two consecutive targets may be arbitrarily smaller than Δ .

■ **Algorithm 3** Simultaneous approximate optimality per group.

```

1 for  $\ell = 1$  to  $g$  do
  /* Step 1 */
2   Let  $\mathcal{T}_\ell : \tau_1 < \tau_2 < \dots$  be the output of Algorithm 1 for agents in  $G_\ell$  and limit
    $\lceil k/g \rceil$  on the number of targets.
  /* Step 2 */
3   Partition  $\mathcal{T}_\ell$  to 4 parts  $P_1, P_2, P_3, P_4$ , where  $P_i := \tau_i, \tau_{4+i}, \tau_{8+i}, \dots$ 
4   Update  $\mathcal{T}_\ell$  by keeping the part with the highest improvement and deleting the
   rest.
  /* Step 3 */
5   Update  $G_\ell$  by deleting the agents that do not improve given  $\mathcal{T}_\ell$ .
6   For all  $\tau \in \mathcal{T}_\ell$ , replace  $\tau$  with the output of Algorithm 1 for agents in
    $[\tau - \Delta, \tau) \cap G_\ell$  and limit 1 on the number of targets.
  /* Step 4 */
7  $\mathcal{T} : \tau_1 < \tau_2 < \dots = \cup_\ell \mathcal{T}_\ell$ 
8  $S, \mathcal{T}^* = \emptyset$ 
9 for  $\tau_j \in \mathcal{T}$  do
10   $s_j = \tau_j - \Delta$ 
11   $S = S \cup \{s_j\}$ 
12 Partition  $S : s_1 < s_2 < \dots$  into the least number of parts of consecutive points:
    $S_1, S_2, \dots$ , such that in each part,  $S_i$ , each two consecutive points are at distance less
   than  $\Delta/g$ .
13 for all  $S_i : s_u < s_{u+1} < \dots < s_v$  do
14   $\tau_i^* = \min\{\tau_u, s_{v+1}\}$ .
15   $\mathcal{T}^* = \mathcal{T}^* \cup \tau_i^*$ .
16 return  $\mathcal{T}^*$ 

```

► **Lemma 15.** Consider solution $\mathcal{T} : \tau_1 < \tau_2 < \dots$ with total improvement I such that for all j , $\tau_{j+2} - \tau_j \geq \Delta$. Consider the procedure in Definition 14. This procedure results in a solution $\mathcal{T}' : \tau'_1 < \tau'_2 < \dots$ where $\forall j \tau'_{j+1} - \tau'_j \geq 2\Delta$, has total improvement at least $I/4$, and $|\mathcal{T}'| \leq \lceil |\mathcal{T}|/4 \rceil$. Particularly, for $|\mathcal{T}| \leq \lceil k/g \rceil$ where $k \geq g$, the number of final targets, $|\mathcal{T}'|$, is at most $\lceil k/g \rceil$.

Proof. Since the best out of 4 parts have been selected, the total improvement at the end of the procedure is at least $1/4$ fraction of I . In addition, in the final set, every pair of consecutive targets are indexed τ_j and τ_{j+4} . Therefore, since originally for all j , $\tau_{j+2} - \tau_j \geq \Delta$, we have $\tau_{j+4} - \tau_j \geq 2\Delta$. Finally, since in each set of $\tau_j, \dots, \tau_{j+4}$ exactly one target is selected, the final number of targets is at most $\lceil |\mathcal{T}|/4 \rceil$. ◀

5.3 Step 3: Locally optimized distant targets in isolation

At the end of step 2, every two targets in \mathcal{T}_ℓ , the set of targets for group ℓ , are at distance at least 2Δ . Consider only the targets and agents in group ℓ . For each $\tau \in \mathcal{T}_\ell$, agents in $[\tau - \Delta, \tau)$ improve to τ and the remaining agents do not improve. To continue with the algorithm, we first delete the agents that do not improve (and update G_ℓ accordingly). Then, we optimize \mathcal{T}_ℓ for the set of agents that do improve. This modification is necessary for the next step. To do the optimization, we use Algorithm 1 for agents in $[\tau - \Delta, \tau)$ for any $\tau \in \mathcal{T}_\ell$ and limit 1 on the number of targets, and replace τ with the output of the algorithm.

► **Lemma 16.** *At the end of step 3 in Algorithm 3, (i) the distance between every two targets in \mathcal{T}_ℓ is at least Δ ; (ii) each target $\tau \in \mathcal{T}_\ell$ is optimal, i.e., maximizes total improvement for the remaining agents in $G_\ell \cap [\tau - \Delta, \tau)$; and (iii) the total amount of improvement of G_ℓ using solution \mathcal{T}_ℓ does not decrease compared to the previous step.*

Proof. Let τ be a target at the beginning of step 3 and τ' be its replacement at the end of this step.

We first prove statement (i). First, we argue for agents in $[\tau - \Delta, \tau)$, the optimal target τ' belongs to $[\tau, \tau + \Delta]$. Intuitively, the reason is that all these agents afford to improve to τ ; therefore, a target smaller than τ is suboptimal. Also, none of the agents affords to improve beyond $\tau + \Delta$. More formally, if $\tau' < \tau$, agents in $[\tau - \Delta, \tau')$ improve less compared to a target at τ and agents in $[\tau', \tau)$ do not improve. On the other hand, if $\tau' > \tau + \Delta$, none of the agents can reach τ' and the total improvement for these agents will be 0. Therefore, at the end of this step, every target τ is replaced with $\tau' \in [\tau, \tau + \Delta]$. Now, by Observation 13 and Lemma 15, the distance between consecutive targets at the end of step 2 is at least 2Δ . Therefore, after the modification explained (shifting each target to the right by less than Δ) this distance decreases by at most Δ and becomes at least Δ .

Now, we move on to statement (ii). We need to argue if τ' is optimal for agents in $[\tau - \Delta, \tau)$, it is also optimal for agents in $[\tau' - \Delta, \tau')$. By Lemma 15, at the beginning of step 3, there are no targets in $(\tau, \tau + 2\Delta)$; more specifically, there are no targets for agents in $[\tau, \tau + \Delta)$ and these agents get eliminated in this step. Therefore, since τ' belongs to $[\tau, \tau + \Delta]$, as shown in the proof of statement (i), we only need to argue that if τ' is optimal for $[\tau - \Delta, \tau)$, it is also optimal for $[\tau' - \Delta, \tau)$. Suppose this was not the case, and there was another target τ'' which was optimal for this set. Since the agents in $[\tau' - \Delta, \tau)$ are the only agents with positive amount of improvement for target τ' , replacing τ' with τ'' would result in higher improvement for the whole set of agents in $[\tau - \Delta, \tau)$ which is in contradiction with definition of τ' .

Finally, we argue statement (iii). In step 3, we consider two sets of agents: those who do not improve in step 2, and those who do. The new targets in this step have been optimized for the second set and (weakly) increase their total improvement. Since the first set did not have any improvement in the first place, the total amount of improvement (for the first and second set) does not decrease in this step. ◀

Next, we extract properties about optimal solutions. Since at the end of step 3, \mathcal{T}_ℓ is optimal for G_ℓ we take advantage of these properties in the remaining steps of the algorithm. Lemma 17 shows that if τ is optimal for agents in $[\tau - \Delta, \tau)$, a considerable fraction of these agents reside in the left-most part of the interval. Lemma 18 shows that if τ is optimal for agents in $[\tau - \Delta, \tau)$, substituting τ with another target in this interval, far enough from the left endpoint, $\tau - \Delta$, guarantees a considerable fraction of the optimal improvement.

► **Lemma 17.** *Consider optimal target τ for the set of agents A in $[\tau - \Delta, \tau)$ in absence of other targets. For each $0 \leq x \leq 1$, at least x fraction of A belong to $[\tau - \Delta, \tau - \Delta + x\Delta)$. In particular, at least $1/(2g)$ fraction of the agents are in $[\tau - \Delta, \tau - (2g - 1)/(2g)\Delta)$.*

Proof. Let p_x be the fraction of agents in A in $[\tau - \Delta, \tau - \Delta + x\Delta)$. Each of these agents is improving by at least $(1 - x)\Delta$. Therefore, the contribution of these agents to total improvement of A is at least $p_x|A|(1 - x)\Delta$. Since τ is the optimal target, it introduces at least as much improvement as any other target, and in particular a target at $\tau' = \tau + x$. Consider the total improvement introduced by τ' compared to τ (in absence of target τ). The contribution of the agents in $[\tau - \Delta, \tau - \Delta + x\Delta)$ to total improvement reduces to 0, but

the contribution of the agents in $[\tau - \Delta + x\Delta, \tau)$ increases by $(1 - p_x)|A|x\Delta$. Since τ is the optimal target, the loss of substituting it with τ' is at least as much as the gain. Therefore, $p_x(1 - x)\Delta \geq (1 - p_x)x\Delta$; which implies $p_x \geq x$. \blacktriangleleft

► **Lemma 18.** *Consider optimal target τ for agents A in $[\tau - \Delta, \tau)$ in absence of other targets. By relocating τ to any point in $[\tau - \Delta + x\Delta, \tau]$, for $0 \leq x \leq 1$, the total improvement of A is at least $x^2/4$ of the optimum. In particular, by relocating τ to any point in $[\tau - \Delta + \Delta/g, \tau]$, the total improvement is at least $1/(4g^2)$ of the optimum.*

Proof. Similar to the previous lemma, let $p_{x/2}$ be the fraction of agents in $[\tau - \Delta, \tau - \Delta + (x/2)\Delta)$. After the relocation, each such agent improves by at least $(x/2)\Delta$; therefore, the contribution of these agents to total improvement is at least $p_{x/2}|A|(x/2)\Delta$. The optimal total improvement is bounded by $|A|\Delta$. Therefore, using $p_{x/2} \geq x/2$, by Lemma 17, the total improvement after relocation is at least $x^2/4$ of the optimum. \blacktriangleleft

5.4 Step 4: Resolve interference of targets

In this step, we consider the solutions for all groups together and resolve the interference of targets designed for different groups. As illustrated in Example 11, this interference can lead to arbitrarily low social welfare. To resolve this issue, we take advantage of sparsity of the targets designed for the same group (step 2) and optimality of \mathcal{T}_ℓ for G_ℓ (step 3).

The main purpose of this step is to recover an approximation guarantee of the total improvement of *each target in isolation* at the end of step 3 by removing the interference among the targets. Particularly, for each target $\tau \in \mathcal{T}_\ell$ in isolation, we consider agents in G_ℓ reaching to that, i.e., agents in interval $[\tau - \Delta, \tau)$. By Lemma 17, a considerable fraction of these agents are on the left-most side of the interval. And as shown in Lemma 18, as long as there exists a target far enough from the left endpoint we are in good shape. More precisely, if for all τ at the beginning of this step, there is a target in the final solution in $[\tau - \Delta + \Delta/g, \tau]$ (property 1), and no targets in $(\tau - \Delta, \tau - \Delta + \Delta/g)$ (property 2), a $1/(4g^2)$ fraction is achievable. The set of targets at the end of step 3 may fail to satisfy these properties, because there may be targets $\tau' < \tau$ such that τ' is not far enough from the left endpoint of the interval corresponding to τ ; i.e., for $s = \tau - \Delta$, $s < \tau' < s + \Delta/g$.

To resolve the interference among the targets, in step 4, we work as follows. First, we consider the left endpoints of improvement intervals corresponding to the targets; i.e., $\forall \tau_j$, at the end of step 3, consider $s_j = \tau_j - \Delta$. Then, we partition these left endpoints into maximal parts S_1, S_2, \dots , such that in each part, the distance between every two consecutive points is small, particularly, less than Δ/g . Using the sparsity of the targets (step 2) the number of points in each part is bounded. Finally, we design a new target τ_i^* (defined formally below) corresponding to part S_i , such that τ_i^* is to the left of any S_j with $j > i$, and at distance between Δ/g and Δ to the right of the points in S_i (satisfying properties 1 and 2). Using optimality of \mathcal{T}_ℓ for G_ℓ (step 3) this results in the desired approximation factor. More formally, this step proceeds as follows.

1. Let $\mathcal{T} : \tau_1 < \tau_2 < \dots$ be the union of the set of targets found at the end of step 3.
2. Construct $S : s_1 < s_2 < \dots$ from \mathcal{T} , such that $\forall \tau_j \in \mathcal{T}$, include $s_j = \tau_j - \Delta$ in S .
3. Partition S into the least number of parts of consecutive points: S_1, S_2, \dots , such that in each part $S_i : s_u < s_{u+1} < \dots < s_v$, each two consecutive points are at distance less than Δ/g ; i.e., $\forall s_r, s_{r+1} \in S_i, s_{r+1} - s_r < \Delta/g$. By construction of the first three steps (and as shown in the proof of Lemma 19), the number of points in each part is at most g .
4. For each $S_i : s_u < s_{u+1} < \dots < s_v$, consider new target $\tau_i^* = \min\{\tau_u, s_{v+1}\}$.
5. Output the set of new targets.

► **Lemma 19.** Consider \mathcal{T} as the union of all solutions at the end of step 3. For all $\tau \in \mathcal{T}$, consider the interval $[\tau - \Delta, \tau)$ which consists of agents that improve to target τ if it were the only target available. At the end of step 4, (i) there will be a target in $[\tau - \Delta + \Delta/g, \tau)$, and (ii) there will be no targets in $(\tau - \Delta, \tau - \Delta + \Delta/g)$.

Proof. Statement (i) is equivalent to (i') for any $s \in S$, there will be a target in $[s + \Delta/g, s + \Delta)$; and statement (ii) is equivalent to (ii') for any $s \in S$, there will be no targets in $(s, s + \Delta/g)$. We prove (i') and (ii').

We first show the size of each part is at most g ; i.e. $\forall i, |S_i| \geq g$. The proof is by contradiction. Suppose there exists $|S_i| \geq g + 1$. Therefore, there exist $s_j < s_{j'} \in S_i$ and group index ℓ , such that $s_j + \Delta, s_{j'} + \Delta \in \mathcal{T}_\ell$, and all s satisfying $s_j < s < s_{j'} \in S_i$ corresponding to targets in distinct groups other than ℓ . Therefore, there are at most $g - 1$ such s . Hence, $s_{j'} - s_j < g \times \Delta/g = \Delta$, implying there are two targets in \mathcal{T}_ℓ at distance strictly less than Δ which is in contradiction with Lemma 16.

Now, we prove statement (i''). In step 4, the final target corresponding to part $S_i : s_u \leq s_{u+1} \leq \dots \leq s_v$ is defined as $\tau_i^* = \min\{\tau_v, s_{v+1}\}$. By definition, $\tau_i^* \leq s_{v+1}$; therefore, it is (weakly) to the left of any s_j for $j \geq v + 1$. Also, using $|S_i| \leq g$, $s_v < s_u + (g - 1)\Delta/g$, which implies $\tau_u - s_v > \Delta/g$, and since by definition, $s_{v+1} - s_v \geq \Delta/g$, both s_{v+1} and τ_u are at least at distance Δ/g to the right of s_v and any s_j such that $j \leq v$. This proves statement (i'').

Finally, we prove (i'). In the proof of (ii'), we showed that $\tau_i^* \geq s_v + \Delta/g$ which implies $\tau_i^* \geq s + \Delta/g, \forall s \in S_i$. Therefore, it suffices to show $\tau_i^* \leq s_u + \Delta$, which then implies $\tau_i^* \leq s + \Delta, \forall s \in S_i$. The definition of τ_i^* directly implies $\tau_i^* \leq s_u + \Delta$. ◀

5.5 Putting everything together

► **Theorem 20.** Algorithm 3, given $k \geq g$, provides a solution with at most k number of targets, such that for all $1 \leq \ell \leq g$, $SW_\ell \geq 1/(16g^2)OPT_\ell^{\lceil k/g \rceil}$, where OPT_ℓ^k is the optimal social welfare of group ℓ using at most k target levels. This statement holds in the common improvement capacity model.

Proof. By Observation 13 and Lemma 15, when the targets designed for each group are considered separately and in isolation, at the end of step 2, there are at most $\lfloor k/g \rfloor$ targets designed for group ℓ and the total improvement in this group is $1/4$ -approximation of $OPT_\ell^{\lceil k/g \rceil}$. By Lemma 16, Lemma 18, and Lemma 19, we lose another $4g^2$ factor compared to step 2. In total, Algorithm 3 results in $SW_\ell \geq 1/(16g^2)OPT_\ell^{\lceil k/g \rceil}$, for all groups $1 \leq \ell \leq g$. Also, when $k \geq g$, the total number of targets is at most $g \lfloor k/g \rfloor \leq k$. ◀

Proof of Theorem 7. Given Theorem 20, it suffices to argue $OPT_\ell^{\lceil k/g \rceil} \geq OPT_\ell^k/g$; i.e., when the number of targets increases by a factor, here g , the optimal total improvement increases by at most that factor. The argument follows using the subadditivity of total improvement as a function of the set of targets. Specifically, consider the optimal k -target solution and an arbitrary partition with g parts of size $\lceil k/g \rceil$ or $\lfloor k/g \rfloor$; by subadditivity, one of the parts provides at least $1/g$ of the total improvement. ◀

Proof of Corollary 8. Algorithm 2 in Section 4 outputs the Pareto frontier for groups' social welfare. By definition, the solution provided in Algorithm 3 is dominated by a solution on the Pareto frontier. By computing the factor of simultaneous approximate optimality of each solution on the Pareto frontier, we find the solution that achieves the best simultaneous approximation factor α^* , and by Theorem 7, this solution is simultaneously $\Omega(1/g^3)$ -approximately optimal. ◀

► **Remark 21** (a weaker benchmark and a tighter gap). In contrast with Theorem 7 that measures the performance of Algorithm 3 with respect to the optimal k -target solution for each group (the notion of simultaneous approximate optimality), Theorem 20 measures the performance with respect to the optimal $\lceil k/g \rceil$ -target solution for each group. Since the lower bound provided in Example 9 shows achieving better than $1/g$ of either of these benchmarks is not possible, there is only a factor g gap in the performance of the algorithm and the lower bound with respect to the optimal $\lceil k/g \rceil$ -target solution.

6 Generalization Guarantees

In this section, we generalize our results to a setting where we only have sample access to agents and provide sample complexity results. Section 6.1 provides a guarantee for the maximization objective in absence of fairness, and Section 6.1 provides a guarantee for the fairness objectives.

6.1 Generalization Guarantees For the Maximization Objective

Suppose there is a distribution \mathcal{D} over agents' positions. Our goal is to find a set of k targets \mathcal{T} that maximizes expected improvement of an agent when we only have access to n agents sampled from \mathcal{D} . For any distribution \mathcal{D} over agents' positions, we define $I_{\mathcal{D}}(\mathcal{T}) = \mathbb{E}_{p \sim \mathcal{D}}[I_p(\mathcal{T})]$, where $I_p(\mathcal{T})$ captures the improvement of agent p given the targets in \mathcal{T} . In Theorem 22, we provide a generalization guarantee that shows if we sample a set S of size $n \geq \varepsilon^{-2}(\Delta_{\max}^2(k \ln(k) + \ln(1/\delta)))$ drawn *i.i.d* from \mathcal{D} , then with probability at least $1 - \delta$, for all sets \mathcal{T} of k targets, we can bound the difference between average performance over S and actual expected performance, such that $|I_S(\mathcal{T}) - I_{\mathcal{D}}(\mathcal{T})| \leq \mathcal{O}(\varepsilon)$. Formally, we show the following theorem holds:

► **Theorem 22** (Generalization of the maximization objective). *Let \mathcal{D} be a distribution over agents' positions. For any $\varepsilon > 0$, $\delta > 0$, and number of targets k , if $S = \{p_i\}_{i=1}^n$ is drawn *i.i.d* from \mathcal{D} where $n \geq \varepsilon^{-2}\Delta_{\max}^2(k \ln(k) + \ln(1/\delta))$, then with probability at least $1 - \delta$, for all sets \mathcal{T} of k targets, $|I_S(\mathcal{T}) - I_{\mathcal{D}}(\mathcal{T})| \leq \mathcal{O}(\varepsilon)$.*

In particular, the solution \mathcal{T}^* that maximizes improvement on \mathcal{S} , also maximizes improvement on \mathcal{D} within an additive factor of $\mathcal{O}(\varepsilon)$.

In order to prove Theorem 22, we use two main ideas. First, using a framework developed by Balcan et al. [6], we bound the *pseudo-dimension* complexity of our improvement function. Then, using classic results from learning theory [24], we show how to translate *pseudo-dimension* bounds into generalization guarantees. The framework proposed by Balcan et al. [6] depends on the relationship between primal and dual functions. When the dual function is piece-wise constant, piece-wise linear or generally piece-wise structured, they show a general theorem that bounds the *pseudo-dimension* of the primal function. Formally *pseudo-dimension* is defined as following:

► **Definition 23** (Pollard's Pseudo-Dimension). *A class \mathcal{F} of real-valued functions P -shatters a set of points $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$ if there exists a set of thresholds $\gamma_1, \gamma_2, \dots, \gamma_n$ such that for every subset $T \subseteq \mathcal{X}$, there exists a function $f_T \in \mathcal{F}$ such that $f_T(x_i) \geq \gamma_i$ if and only if $x_i \in T$. In other words, all 2^n possible above/below patterns are achievable for targets $\gamma_1, \dots, \gamma_n$. The pseudo-dimension of \mathcal{F} , denoted by $\text{PDim}(\mathcal{F})$, is the size of the largest set of points that it P -shatters.*

Balcan et al. [6] show when the dual function is piece-wise structured, the *pseudo-dimension* of the primal function gets bounded as following:

► **Theorem 24** (Bounding Pseudo-Dimension [6]). *Let $\mathcal{U} = \{u_\rho \mid \rho \in \mathcal{P} \subseteq \mathbb{R}^d\}$ be a class of utility functions defined over a d -dimensional parameter space. Suppose the dual class \mathcal{U}^* is $(\mathcal{F}, \mathcal{G}, m)$ -piecewise decomposable, where the boundary functions $\mathcal{G} = \{f_{\mathbf{a}, \theta} : \mathcal{U} \rightarrow \{0, 1\} \mid \mathbf{a} \in \mathbb{R}^d, \theta \in \mathbb{R}\}$ are halfspace indicator functions $g_{\mathbf{a}, \theta} : u_\rho \rightarrow \mathbb{I}_{\mathbf{a} \cdot \rho \leq \theta}$ and the piece functions $\mathcal{F} = \{f_{\mathbf{a}, \theta} : \mathcal{U} \rightarrow \mathbb{R} \mid \mathbf{a} \in \mathbb{R}^d, \theta \in \mathbb{R}\}$ are linear functions $f_{\mathbf{a}, \theta} : u_\rho \rightarrow \mathbf{a} \cdot \rho + \theta$, and m shows the number of boundary functions. Then, $\text{PDim}(\mathcal{U}) = \mathcal{O}(d \ln(dm))$.*

We use Theorem 24 to bound the *pseudo-dimension* of the improvement function.

► **Lemma 25.** *Let $\mathcal{U} = \{u_{\mathcal{T}} : p \rightarrow u_{\mathcal{T}}(p) \mid \mathcal{T} \in \mathbb{R}^k, p \in \mathbb{R}\}$ be a set of functions, where each function defined by a set of k targets, takes as input a point $p \in \mathbb{R}$ that captures an agent's position, and outputs a number showing the improvement that the agent can make. Then, $\text{PDim}(\mathcal{U}) = \mathcal{O}(k \ln(k))$.*

Proof. We use Theorem 24 to bound $\text{PDim}(\mathcal{U})$. First, we define the dual class of \mathcal{U} denoted by \mathcal{U}^* . The function class $\mathcal{U}^* = \{u_p^* : \mathcal{T} \rightarrow u_p(\mathcal{T}) \mid \mathcal{T} \in \mathbb{R}^k, p \in \mathbb{R}\}$ is a set of functions, where each function defined by an agent p , takes as input a set $\mathcal{T} \in \mathbb{R}^k$ of k targets⁵, and outputs the improvement that p can make given \mathcal{T} . Geometrically, in the dual space, there are k dimensions τ_1, \dots, τ_k , and each dimension is corresponding to one target. In order to use Theorem 24, we show that $\mathcal{U}^* = (\mathcal{F}, \mathcal{G}, k)$ is piecewise-structured. The boundary functions in \mathcal{G} are defined as follows. If agent p improves to a target τ_i , then $0 < \tau_i - p \leq \Delta$, where Δ is the improvement capacity of p . Additionally, between all the targets within a distance of at most Δ , p improves to the closest one. For each pair of integers (i, j) , where $1 \leq i, j \leq k$, we add the hyperplane $\tau_i - \tau_j = 0$ to \mathcal{G} . Above this hyperplane is the region where $\tau_i > \tau_j$, implying that τ_i comes after τ_j . Below the hyperplane is the region where the ordering is reversed. In addition, for each target τ_i , we add the boundary functions $\tau_i = p$ and $\tau_i = p + \Delta$ to \mathcal{G} . In the region between $\tau_i = p$ and $\tau_i = p + \Delta$, τ_i is effective and the agent can improve to it. Now, the dual space is partitioned into a set of regions. In each region, either there exists a unique closest effective target (τ_r), or all the targets are ineffective. In the former case, the improvement that the agent makes is a linear function of its distance from the closest effective target ($f = \tau_r - p$). In the later case, the agent makes no improvement ($f = 0$). Therefore, the piece functions in \mathcal{F} are either constant or linear. Now, since the total number of boundary functions is $m = \mathcal{O}(k^2)$ and the space is k -dimensional, using Theorem 24, $\text{PDim}(\mathcal{U})$ is $\mathcal{O}(k \ln(k^3)) = \mathcal{O}(k \ln(k))$. ◀

Now, we are ready to prove Theorem 22.

Proof. Classic results from learning theory [24] show the following generalization guarantees: Suppose $[0, H]$ is the range of functions in hypothesis class \mathcal{H} . For any $\delta \in (0, 1)$, and any distribution \mathcal{D} over \mathcal{X} , with probability $1 - \delta$ over the draw of $\mathcal{S} \sim \mathcal{D}^n$, for all functions $h \in \mathcal{H}$, the difference between the average value of h over \mathcal{S} and its expected value gets bounded as follows:

$$\left| \frac{1}{n} \sum_{x \in \mathcal{S}} h(x) - \mathbb{E}_{y \sim \mathcal{D}}[h(y)] \right| = \mathcal{O}\left(H \sqrt{\frac{1}{n} \left(\text{PDim}(\mathcal{H}) + \ln\left(\frac{1}{\delta}\right) \right)}\right)$$

⁵ If the input consists of k' targets where $k' < k$, it resembles the case where k targets are used and $k - k'$ of them are ineffective, i.e., are put at position τ_{\min} .

In the case of maximizing improvement, $H = \Delta_{max}$ and $\text{PDim}(\mathcal{H}) = \mathcal{O}(k \ln(k))$. By setting $n \geq \varepsilon^{-2} \Delta_{max}^2 (k \ln(k) + \ln(1/\delta))$, with probability at least $1 - \delta$, the difference between the average performance over \mathcal{S} and the expected performance on \mathcal{D} gets upper-bounded by $\mathcal{O}(\varepsilon)$. \blacktriangleleft

6.2 Generalization Guarantees For Fairness Objectives

Suppose there is a distribution \mathcal{D}_ℓ of agents' positions for each group ℓ . Let $\mathcal{D} = \sum_{\ell=1}^g \alpha_\ell \mathcal{D}_\ell$ be a weighted mixture of distributions $\mathcal{D}_1, \dots, \mathcal{D}_g$. Let $\alpha_{\min} = \min_{1 \leq \ell \leq g} \alpha_\ell$. Suppose we have sampling access to \mathcal{D} and cannot directly sample from $\mathcal{D}_1, \dots, \mathcal{D}_g$. Our goal is to derive generalization guarantees for different objective functions across multiple groups when we only have access to a set S of n agents sampled from distribution \mathcal{D} . Let $I_{G_\ell}(\mathcal{T})$ denote the average improvement of agents in group $G_\ell \subseteq S$ given a set \mathcal{T} of k targets. Let $I_{\mathcal{D}_\ell}(\mathcal{T}) = \mathbb{E}_{p \sim \mathcal{D}_\ell}[I_p(\mathcal{T})]$, where $I_p(\mathcal{T})$ captures the improvement of agent p given \mathcal{T} . In Theorem 26, we show if we sample a set S of $\mathcal{O}\left(\alpha_{\min}^{-1} \left(\varepsilon^{-2} \Delta_{max}^2 (k \ln(k) + \ln(g/\delta)) + \ln(g/\delta)\right)\right)$ examples drawn *i.i.d.* from \mathcal{D} , then for all sets \mathcal{T} of k targets and for all groups ℓ , $|I_{G_\ell}(\mathcal{T}) - I_{\mathcal{D}_\ell}(\mathcal{T})| \leq \mathcal{O}(\varepsilon)$.

► Theorem 26 (Generalization across multiple groups). *Let \mathcal{D} be a distribution over agents' positions. For any $\varepsilon > 0$, $\delta > 0$, and number of targets k , if $S = \{p_i\}_{i=1}^n$ consisting of g groups $\{G_\ell\}_{\ell=1}^g$ is drawn *i.i.d.* from \mathcal{D} , where $n \geq (2/\alpha_{\min}) \left(\varepsilon^{-2} \Delta_{max}^2 (k \ln(k) + \ln(2g/\delta)) + 4 \ln(2g/\delta)\right)$, then with probability at least $1 - \delta$, for all sets \mathcal{T} of k targets, for all groups ℓ , $|I_{G_\ell}(\mathcal{T}) - I_{\mathcal{D}_\ell}(\mathcal{T})| \leq \mathcal{O}(\varepsilon)$.*

Proof. Let S be partitioned into g groups where each group G_ℓ has size n_ℓ . First, for each group ℓ , let A_ℓ denote the event that $n_\ell \geq (n\alpha_\ell)/2$. Using Chernoff-Hoeffding bounds we have $\Pr[n_\ell < (n\alpha_\ell)/2] \leq e^{(-n\alpha_\ell)/8} \leq \delta/(2g)$. The last inequality holds since $n \geq 8 \ln(2g/\delta)/\alpha_\ell$. Next, for each group ℓ , let B_ℓ denote the event that $|I_{G_\ell}(\mathcal{T}) - I_{\mathcal{D}_\ell}(\mathcal{T})| \leq \mathcal{O}(\varepsilon)$, then:

$$\Pr[B_\ell] \geq \Pr[B_\ell \cap A_\ell] = \Pr[B_\ell | A_\ell] \cdot \Pr[A_\ell] \geq (1 - \delta/(2g))(1 - \delta/(2g)) \geq (1 - \delta/g) \quad (1)$$

In the above statement, inequality $\Pr[B_\ell | A_\ell] \geq (1 - \delta/(2g))$ holds since given A_ℓ happens, then $n_\ell \geq \varepsilon^{-2} \Delta_{max}^2 (k \ln(k) + \ln(2g/\delta))$, and by Theorem 22, event B_ℓ happens with probability at least $1 - \delta/(2g)$. Now, by Equation (1), $\Pr[B_\ell] \geq 1 - \delta/g$. By applying a union bound, event B_ℓ happens with probability at least $1 - \delta$ for any group ℓ . \blacktriangleleft

In particular, solution \mathcal{T}^* satisfying one of the fairness notions considered in this paper, e.g., simultaneous approximate optimality or maximizing minimum improvement across groups, on input S , achieves a performance guarantee within an additive factor of $\mathcal{O}(\varepsilon)$ on inputs drawn from \mathcal{D} .

References

- 1 Saba Ahmadi, Hedyeh Beyhaghi, Avrim Blum, and Keziah Naggita. The strategic perceptron. In Péter Biró, Shuchi Chawla, and Federico Echenique, editors, *EC '21: The 22nd ACM Conference on Economics and Computation, Budapest, Hungary, July 18-23, 2021*, pages 6–25. ACM, 2021. doi:10.1145/3465456.3467629.
- 2 Saba Ahmadi, Hedyeh Beyhaghi, Avrim Blum, and Keziah Naggita. On classification of strategic agents who can both game and improve. In L. Elisa Celis, editor, *3rd Symposium on Foundations of Responsible Computing, FORC 2022, June 6-8, 2022, Cambridge, MA, USA*, volume 218 of *LIPIcs*, pages 3:1–3:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.FORC.2022.3.

- 3 Tal Alon, Magdalen Dobson, Ariel Procaccia, Inbal Talgam-Cohen, and Jamie Tucker-Foltz. Multiagent evaluation mechanisms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(02):1774–1781, April 2020. doi:10.1609/aaai.v34i02.5543.
- 4 Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec. Steering user behavior with badges. In *Proceedings of the 22nd International Conference on World Wide Web, WWW '13*, pages 95–106, New York, NY, USA, 2013. Association for Computing Machinery. doi:10.1145/2488388.2488398.
- 5 Moshe Babaioff, Shahar Dobzinski, Sigal Oren, and Aviv Zohar. On bitcoin and red balloons. In Boi Faltings, Kevin Leyton-Brown, and Panos Ipeirotis, editors, *Proceedings of the 13th ACM Conference on Electronic Commerce, EC 2012, Valencia, Spain, June 4-8, 2012*, pages 56–73. ACM, 2012. doi:10.1145/2229012.2229022.
- 6 Maria-Florina Balcan, Dan DeBlasio, Travis Dick, Carl Kingsford, Tuomas Sandholm, and Ellen Vitercik. How much data is sufficient to learn high-performing algorithms? generalization guarantees for data-driven algorithm design. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2021*, pages 919–932, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3406325.3451036.
- 7 Yahav Bechavod, Katrina Ligett, Zhiwei Steven Wu, and Juba Ziani. Gaming helps! learning from strategic interactions in natural dynamics. In Arindam Banerjee and Kenji Fukumizu, editors, *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pages 1234–1242. PMLR, 2021. URL: <http://proceedings.mlr.press/v130/bechavod21a.html>.
- 8 Yahav Bechavod, Chara Podimata, Zhiwei Steven Wu, and Juba Ziani. Information discrepancy in strategic learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 1691–1715. PMLR, 2022. URL: <https://proceedings.mlr.press/v162/bechavod22a.html>.
- 9 Mark Braverman and Sumegha Garg. The role of randomness and noise in strategic classification. In Aaron Roth, editor, *1st Symposium on Foundations of Responsible Computing, FORC 2020, June 1-3, 2020, Harvard University, Cambridge, MA, USA (virtual conference)*, volume 156 of *LIPICs*, pages 9:1–9:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.FORC.2020.9.
- 10 Michael Brückner and Tobias Scheffer. Stackelberg games for adversarial prediction problems. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '11*, pages 547–555, New York, NY, USA, 2011. Association for Computing Machinery. doi:10.1145/2020408.2020495.
- 11 Moira Burke, Cameron Marlow, and Thomas M. Lento. Feed me: motivating newcomer contribution in social network sites. In Dan R. Olsen Jr., Richard B. Arthur, Ken Hinckley, Meredith Ringel Morris, Scott E. Hudson, and Saul Greenberg, editors, *Proceedings of the 27th International Conference on Human Factors in Computing Systems, CHI 2009, Boston, MA, USA, April 4-9, 2009*, pages 945–954. ACM, 2009. doi:10.1145/1518701.1518847.
- 12 Moira Burke and Burr Settles. Plugged in to the community: social motivators in online goal-setting groups. In Marcus Foth, Jesper Kjeldskov, and Jeni Paay, editors, *Proceedings of the Fifth International Conference on Communities and Technologies, C&T 2011, Brisbane, QLD, Australia, June 29 - July 2, 2011*, pages 1–10. ACM, 2011. doi:10.1145/2103354.2103356.
- 13 Emily Diana, Travis Dick, Hadi Elzayn, Michael Kearns, Aaron Roth, Zachary Schutzman, Saeed Sharifi-Malvajerdi, and Juba Ziani. Algorithms and learning for fair portfolio design. In *Proceedings of the 22nd ACM Conference on Economics and Computation, EC '21*, pages 371–389, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3465456.3467646.

- 14 Jinshuo Dong, Aaron Roth, Zachary Schutzman, Bo Waggoner, and Zhiwei Steven Wu. Strategic classification from revealed preferences. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, EC '18, pages 55–70, New York, NY, USA, 2018. Association for Computing Machinery. doi:10.1145/3219166.3219193.
- 15 David Easley and Arpita Ghosh. Incentives, gamification, and game theory: An economic approach to badge design. In *Proceedings of the Fourteenth ACM Conference on Electronic Commerce*, EC '13, pages 359–376, New York, NY, USA, 2013. Association for Computing Machinery. doi:10.1145/2492002.2482571.
- 16 Alex Frankel and Navin Kartik. Improving Information from Manipulable Data. *Journal of the European Economic Association*, 20(1):79–115, June 2021. doi:10.1093/jeea/jvab017.
- 17 Nika Haghtalab, Nicole Immorlica, Brendan Lucier, and Jack Z. Wang. Maximizing welfare with incentive-aware evaluation mechanisms. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 160–166. International Joint Conferences on Artificial Intelligence Organization, July 2020. Main track. doi:10.24963/ijcai.2020/23.
- 18 Moritz Hardt, Nimrod Megiddo, Christos Papadimitriou, and Mary Wootters. Strategic classification. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, ITCS '16, pages 111–122, New York, NY, USA, 2016. Association for Computing Machinery. doi:10.1145/2840728.2840730.
- 19 Keegan Harris, Hoda Heidari, and Zhiwei Steven Wu. Stateful strategic regression. *CoRR*, abs/2106.03827, 2021. arXiv:2106.03827.
- 20 Lily Hu, Nicole Immorlica, and Jennifer Wortman Vaughan. The disparate effects of strategic manipulation. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, pages 259–268, New York, NY, USA, 2019. ACM. doi:10.1145/3287560.3287597.
- 21 Jon Kleinberg and Manish Raghavan. How do classifiers induce agents to invest effort strategically? In *Proceedings of the 2019 ACM Conference on Economics and Computation*, EC '19, pages 825–844, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3328526.3329584.
- 22 John Miller, Smitha Milli, and Moritz Hardt. Strategic classification is causal modeling in disguise. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 6917–6926. PMLR, 2020. URL: <http://proceedings.mlr.press/v119/miller20b.html>.
- 23 Smitha Milli, John Miller, Anca D. Dragan, and Moritz Hardt. The social cost of strategic classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, pages 230–239, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3287560.3287576.
- 24 D. Pollard. *Convergence of Stochastic Processes*. Springer New York, 1984. URL: <https://books.google.com/books?id=B2vgGMA9vd4C>.
- 25 Yonadav Shavit, Benjamin Edelman, and Brian Axelrod. Learning from strategic agents: Accuracy, improvement, and causality. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume abs/2002.10066 of *Proceedings of Machine Learning Research*, pages 8676–8686. PMLR, 13–18 July 2020. URL: <http://proceedings.mlr.press/v119/shavit20a.html>.
- 26 Ravi Sundaram, Anil Vullikanti, Haifeng Xu, and Fan Yao. Pac-learning for strategic classification. In *International Conference on Machine Learning*, pages 9978–9988. PMLR, 2021.
- 27 David P Williamson and David B Shmoys. *The design of approximation algorithms*. Cambridge university press, 2011.
- 28 Shenke Xiao, Zihe Wang, Mengjing Chen, Pingzhong Tang, and Xiwang Yang. Optimal common contract with heterogeneous agents. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7309–7316, April 2020. doi:10.1609/aaai.v34i05.6224.

A

 Missing Proofs of Theorem 3

A.1 Proof of Theorem 4

► **Theorem 4.** *Algorithm 1 finds a set of targets that achieves the optimal social welfare (maximum total improvement) that is feasible using at most k targets given n agents. The algorithm runs in $\mathcal{O}(n^3)$.*

Proof. Proof of correctness follows by induction. Suppose that the value computed for all $T(\tau', \kappa')$ where $(\tau', \kappa') < (\tau, \kappa)$ is correct. Here “ $<$ ” means (τ', κ') is computed before (τ, κ) which is when $\kappa' < \kappa$ and $\tau' \geq \tau$. First, if either $\tau = \tau_{\max}$ or $\kappa = 0$, the induction hypothesis holds since $T(\tau_{\max}, \kappa) = 0$ for all $1 \leq \kappa \leq k$, and $T(\tau, 0) = 0$, for all $\tau \in \mathcal{T}_p$. To show the inductive step holds note that the algorithm considers the optimal value for $T(\tau, \kappa)$ as the maximum of the $T(\tau', \kappa - 1) + \sum_{\tau \leq p_i < \tau' \text{ s.t. } \tau' - p_i \leq \Delta_i} (\tau' - p_i)$ over all the possible placement of the leftmost target τ' . Since $T(\tau', \kappa - 1)$ is computed correctly by the induction hypothesis and all the possible placements of the leftmost target are considered, the value obtained at $T(\tau, \kappa)$ is optimal and correct.

Now we proceed to bounding the time-complexity. There are $\mathcal{O}(nk)$ subproblems to be computed. Consider a pre-computation stage where $\sum_{\tau \leq p_i < \tau' \text{ s.t. } \tau' - p_i \leq \Delta_i} (\tau' - p_i)$ is computed for all pairs of $\tau, \tau' \in \mathcal{T}_p$. This stage takes $\mathcal{O}(n^3)$ time. Computation of each subproblem $T(\tau, \kappa)$ for all $\tau \in \mathcal{T}_p$ and $1 \leq \kappa \leq k$ requires $\mathcal{O}(n)$ operations. This is because to compute max in property 3), we compute $T(\tau', \kappa - 1) + \sum_{\tau \leq p_i < \tau' \text{ s.t. } \tau' - p_i \leq \Delta_i} (\tau' - p_i)$ for $\mathcal{O}(n)$ potential target levels greater than τ , for which each takes $\mathcal{O}(1)$ time. Since there are $\mathcal{O}(nk)$ subproblems, the running time of the algorithm is $\mathcal{O}(n^2k + n^3) = \mathcal{O}(n^3)$. ◀

B

 Missing Proofs of Section 4

B.1 Proof of Theorem 5

► **Theorem 5.** *Algorithm 2 constructs the Pareto frontier for groups' social welfare using at most k targets given n agents in g groups, and has a running time of $\mathcal{O}(n^{g+2}kg\Delta_{\max}^g)$, where Δ_{\max} is the maximum improvement capacity.*

Proof. Proof of correctness follows by induction and it is along the same lines as proof of Algorithm 1. Suppose that Pareto-frontiers constructed for all $T(\tau', \kappa')$ where $(\tau', \kappa') < (\tau, \kappa)$ is correct. Here “ $<$ ” means (τ', κ') is computed before (τ, κ) which is when $\kappa' < \kappa$ and $\tau' \geq \tau$. First, if either $\tau = \tau_{\max}$ or $\kappa = 0$, the induction hypothesis holds since $T(\tau_{\max}, \kappa) = \emptyset$ for all $1 \leq \kappa \leq k$, and $T(\tau, 0) = \emptyset$, for all $\tau \in \mathcal{T}_p$. The inductive step holds since the algorithm considers all the possible placement of the leftmost target τ' . Since $T(\tau', \kappa - 1)$ is computed correctly by the induction hypothesis and all the possible placements of the leftmost target are considered, the Pareto-frontier constructed at $T(\tau, \kappa)$ is correct.

Now we proceed to bounding the time complexity. Initially, in a pre-computation stage, for each pair of targets $\tau, \tau' \in \mathcal{T}_p$, $\sum_{\tau \leq p_i < \tau' \text{ s.t. } \tau' - p_i \leq \Delta_i} \mathbb{1}\{i \in G_\ell\} (\tau' - p_i)$ is pre-computed for all groups and is stored in a tuple of size g . This stage can be done in $\mathcal{O}(n^3)$. Each set $T(\tau, \kappa)$ has size at most $(n\Delta_{\max} + 1)^g$, since each individual can move for one of the values $\{0, \dots, \Delta_{\max}\}$ and therefore, the total improvement in each group is one of the values $\{0, \dots, n\Delta_{\max}\}$. At each step of the recurrence, given the information stored in the pre-computation stage, the summation can be computed in $\mathcal{O}(g)$. When computing a subproblem $T(\tau, \kappa)$, the recurrence searches over $\mathcal{O}(n)$ targets $\tau' \in \mathcal{T}_p$, and at most $(n\Delta_{\max} + 1)^g$ tuples of group improvement in $T(\tau', \kappa - 1)$. As a result, solving each subproblem takes $\mathcal{O}(ng(n\Delta_{\max})^g)$. The total number of subproblems that need to get solved is $\mathcal{O}(nk)$. Therefore, the total running time of the algorithm is $\mathcal{O}(n^{g+2}kg\Delta_{\max}^g + n^3) = \mathcal{O}(n^{g+2}kg\Delta_{\max}^g)$. ◀

B.2 Proof of Corollary 6

► **Corollary 6.** *There is an efficient algorithm that finds a set of at most k targets that maximizes minimum improvement across all groups, i.e., maximizing $\min_{1 \leq \ell \leq g} SW_\ell$.*

Proof. Algorithm 2 constructs the Pareto frontier for groups' social welfare. By iterating through all Pareto-optimal solutions, we can find the solution that maximizes the minimum improvement across all groups. There are at most $(n\Delta_{\max} + 1)^g$ Pareto-optimal solutions. Finding the minimum improvement in each solution takes $\mathcal{O}(g)$. Therefore, in total, finding the solution that maximizes the minimum improvement across all groups takes $\mathcal{O}(g(n\Delta_{\max})^g)$. ◀

C An FPTAS for Maximizing Minimum Group Improvement

In this section, we present a Fully Polynomial Time Approximation Scheme (FPTAS) to maximize minimum improvement across all groups. Here, we assume that each group ℓ has its own improvement capacity Δ_ℓ . The algorithm finds a set of at most k targets that approximates the max-min objective within a factor of $1 - \varepsilon$ for any arbitrary value of $\varepsilon > 0$. Here, we relax the assumption on the integrality of p_i, Δ_i values needed for the running time guarantee in Theorem 5, and suppose all p_i, Δ_i values are real numbers. Similar to the dynamic program based on Algorithm 2, for each subproblem, a set containing all g -tuples of improvements (I_1, I_2, \dots, I_g) that are simultaneously achievable for all groups is stored. However, computing all such tuples takes exponential time since $\sum_{i=1}^k \binom{2^n}{i}$ possible cases of targets' placements need to be considered. Therefore, we discretize the set of all possible improvements for this problem by rounding all the improvement tuples, and develop an FPTAS algorithm. The recurrence for the dynamic program is given in Algorithm 4. The algorithm runs efficiently when the number of groups is a constant.

► **Algorithm 4.** *The algorithm considers two separate cases of $k < g$, and $k \geq g$. For the $k \geq g$ case, the algorithm finds a set of k targets that approximates the max-min objective within a factor of $1 - \varepsilon$ for any arbitrary value of $\varepsilon > 0$. For the $k < g$ case, it finds an optimal solution for the max-min objective.*

For the $k \geq g$ case, there exists an FPTAS for the max-min objective as follows. First, run a dynamic program using the following recursive function to get a set of Pareto-optimal solutions. In this Pareto-frontier, we show the solution that maximizes minimum improvement across all groups, gives a $(1 - \varepsilon)$ -approximation for the max-min objective. In the recurrence, $\mu_\ell = \varepsilon \Delta_\ell / (16kg^3)$ for $1 \leq \ell \leq g$, and Δ_ℓ is the improvement capacity of agents in group ℓ .

$$\mathcal{F}(\tau', k') = \left\{ \left(\mu_\ell \left[\frac{I'_\ell + \left(\sum_{\substack{\tau' \leq p_i < \tau \\ \text{s.t. } \tau - p_i \leq \Delta_\ell}} \mathbb{1}\{i \in G_\ell\} (\tau - p_i) \right)}{\mu_\ell} \right] \right)_{\ell=1}^g, \right. \\ \left. \text{s.t. } (I'_\ell)_{\ell=1}^g \in \mathcal{F}(\tau, k' - 1), \tau \in \mathcal{T}_p, \tau \geq \tau' \right\}$$

Intuitively, $\mathcal{F}(\tau', k')$ stores the rounded down values of the feasible tuples of group improvements when all agents on or to the right of τ' are available and k' targets are used. The corresponding set of targets used to construct the improvement tuples in $\mathcal{F}(\tau', k')$ is kept in a hash table $\mathcal{S}(\tau', k')$, whose keys are the improvement tuples in $\mathcal{F}(\tau', k')$. The dynamic program ends after computing $\mathcal{F}(\tau_{\min}, k)$ and $\mathcal{S}(\tau_{\min}, k)$. At the end, we output the set of targets in $\mathcal{S}(\tau_{\min}, k)$ corresponding to the improvement tuple that maximizes the improvement of the worst-off group. Lemma 27 shows that this algorithm gives a $(1 - \varepsilon)$ -approximation for the max-min objective when $k \geq g$.

When $k < g$, for each subset of \mathcal{T}_p of size at most k that is corresponding to a placement of targets, we store its corresponding improvement tuple. Next, we iterate through all improvement tuples and output the one that maximizes minimum improvement.

► **Lemma 27.** *Algorithm 4 gives a $(1 - \varepsilon)$ -approximation for the max-min objective when $k \geq g$.*

Proof. The proof is by induction. Consider an improvement tuple (I_1, \dots, I_g) corresponding to an arbitrary set of $k - 1$ targets, and let (I'_1, \dots, I'_g) denote the rounded down values where $I'_\ell = \mu_\ell \lfloor \frac{I_\ell}{\mu_\ell} \rfloor$ for all $1 \leq \ell \leq g$. Suppose that for all $1 \leq \ell \leq g$, $I'_\ell \geq I_\ell - (k - 1)\mu_\ell$.

Now consider an improvement tuple (J_1, \dots, J_g) corresponding to an arbitrary set of k targets. For each $1 \leq \ell \leq g$, let $J'_\ell = \mu_\ell \lfloor \frac{J_\ell}{\mu_\ell} \rfloor$. We show that for each $1 \leq \ell \leq g$, $J'_\ell \geq J_\ell - k\mu_\ell$. For all $1 \leq \ell \leq g$, let $J_\ell = L_\ell + I_\ell$, where L_ℓ is the improvement of group ℓ that the leftmost target provides, and I_ℓ captures the true improvement of group ℓ that the remaining $k - 1$ targets provide. Let $I'_\ell = \mu_\ell \lfloor \frac{I_\ell}{\mu_\ell} \rfloor$. Then $J'_\ell = \mu_\ell \lfloor \frac{L_\ell + I'_\ell}{\mu_\ell} \rfloor$ implying that $J'_\ell \geq L_\ell + I'_\ell - \mu_\ell$. By the induction hypothesis, $I'_\ell \geq I_\ell - (k - 1)\mu_\ell$. Therefore,

$$J'_\ell \geq L_\ell + I'_\ell - \mu_\ell \geq L_\ell + I_\ell - (k - 1)\mu_\ell - \mu_\ell = L_\ell + I_\ell - k\mu_\ell$$

Therefore, for each set of k targets, the rounded improvement of each group ℓ stored in the table is within an additive factor of $k\mu_\ell = \varepsilon\Delta_\ell/(16g^3)$ of its true improvement. We argue that in the solution returned by the algorithm, improvement of each group is at least $(1 - \varepsilon)OPT$. First, when $k \geq 1$, each group can improve for at least Δ_ℓ by setting a target within a distance of Δ_ℓ from its rightmost agent. Now, using Theorem 7 when $k \geq g$, there exists a solution that is simultaneously $1/(16g^3)$ -optimal for all groups. Therefore, the optimum value of the max-min objective is at least $OPT \geq \Delta_\ell/(16g^3)$ for all $1 \leq \ell \leq g$. Therefore, for each solution consisting of k targets, the rounded improvement of each group is within an additive factor of εOPT of its true improvement. As a result, the minimum group improvement in the returned solution is at least $(1 - \varepsilon)OPT$. ◀

In the following, we bound the approximation factor of our algorithm in both cases of $k \geq g$ and $k < g$.

► **Corollary 28.** *Algorithm 4 described above gives a $(1 - \varepsilon)$ -approximation for the max-min objective.*

Proof. For the case of $k \geq g$, by Lemma 27 the algorithm outputs a $(1 - \varepsilon)$ -approximation. For $k < g$, it outputs an optimum solution. Therefore, in total, it gives a $(1 - \varepsilon)$ -approximation for the max-min objective. ◀

In the following, we bound the time-complexity of the algorithm.

► **Theorem 29.** *Algorithm 4 has a running time of $\mathcal{O}(n^{g+2}k^{g+1}g^{3g+1}/\varepsilon^g)$.*

Proof. Initially, in a pre-computation stage, for each pair of targets $\tau, \tau' \in \mathcal{T}_p$, $\sum_{\tau' \leq p_i < \tau \text{ s.t. } \tau - p_i \leq \Delta_\ell} \mathbb{1}\{i \in G_\ell\}(\tau - p_i)$ is pre-computed for all groups and is stored in a tuple of size g . This stage can be done in $\mathcal{O}(n^3)$. Now, first consider the case where $k \geq g$. We show the dynamic programming algorithm using recurrence $\mathcal{F}(\tau', k')$ has a running time of $\mathcal{O}(n^{g+2}k^{g+1}g^{3g+1}/\varepsilon^g)$. Each set $\mathcal{F}(\tau', k')$ and $\mathcal{S}(\tau', k')$ has size at most $\prod_{\ell=1}^g (n\Delta_\ell/\mu_\ell)^g = (16nkg^3/\varepsilon)^g$. At each step of the recurrence, given the information stored in the pre-computation stage, the summation can be computed in $\mathcal{O}(g)$. When computing $\mathcal{F}(\tau', k')$, the recurrence searches over $\mathcal{O}(n)$ targets $\tau \in \mathcal{T}_p$, and at most

$\prod_{\ell=1}^g (n\Delta_\ell/\mu_\ell)^g = (16nkg^3/\varepsilon)^g$ tuples of group improvement in $\mathcal{F}(\tau, k' - 1)$. As a result, solving each subproblem takes $\mathcal{O}(ng(nkg^3/\varepsilon)^g)$. The total number of subproblems that need to get solved is $\mathcal{O}(nk)$. Therefore, the total running time of computing $\mathcal{F}(\tau_{\min}, k)$ is $\mathcal{O}(n^{g+2}k^{g+1}g^{3g+1}/\varepsilon^g)$.

Next, consider the case where $k < g$. The algorithm considers $\mathcal{O}(n^g)$ placements of targets. Given the pre-computation stage, computing the improvement tuple corresponding to each placement of targets takes $\mathcal{O}(kg)$. As a result, this case takes $\mathcal{O}(kgn^g)$.

Therefore, the total running time of algorithm is $\mathcal{O}(n^3 + n^{g+2}k^{g+1}g^{3g+1}/\varepsilon^g + kgn^g) = \mathcal{O}(n^{g+2}k^{g+1}g^{3g+1}/\varepsilon^g)$. ◀

D Distance between consecutive target levels

Observation 13 shows it is without loss of optimality to assume the distance between every other targets is at least Δ in the common improvement capacity model. The following example investigates this property for *consecutive* targets, and shows an instance where the distance between two consecutive targets is arbitrarily small compared to Δ in the optimal solution.

► **Example 30.** Suppose $\Delta = 1$ and there is no limit on the number of targets. Suppose there is an agent at position 0, an agent at position 1, and m agents at position $1 + 1/m$. The optimal solution is $\mathcal{T} = \{\tau_1 = 1, \tau_2 = 1 + 1/m, \tau_3 = 2 + 1/m\}$. As $m \rightarrow \infty$, the distance between τ_1 and τ_2 approaches 0.

E Extensions and Open Problems

E.1 Extension 1: A lower bound on the number of agents that improve

Consider Algorithm 1 whose goal is to find a set of at most k target levels that maximizes the total improvement for a collection of n agents. It is possible that the solution of this algorithm focuses on a small fraction of the agents and does not help many agents to improve. In Algorithm 5, we show how to modify Algorithm 1 to ensure at least n_{lb} agents improve. The main idea for the recursive step (item 4 in Algorithm 5) is to first consider the potential leftmost targets $\tau' > \tau$, let x denote the number of agents that are within reach to τ' , and use the smaller subproblem of finding the optimal targets for agents on or to the right of τ' with one less available target level and an updated lower bound of $\eta - x$, i.e., $S(\tau', \kappa - 1, \eta - x)$. We add the performance of each potential leftmost target to the optimal improvement of the remaining subproblem and pick the leftmost target that maximizes this summation.

► **Algorithm 5.** Run dynamic program based on function S , defined below, that takes $\cup_i \{p_i\}$ and k as input and outputs $S(\tau_{\min}, k, n_{lb})$, as the optimal improvement, and $S'(\tau_{\min}, k, n_{lb})$, as the optimal set of targets; where $\tau_{\min} = \min\{\tau \in \mathcal{T}_p\}$ and $\tau_{\max} = \max\{\tau \in \mathcal{T}_p\}$. $S(\tau, \kappa, \eta)$ captures the maximum improvement possible for agents on or to the right of $\tau \in \mathcal{T}_p$ when κ target levels can be selected and at least η agents need to improve. If $S(\tau_{\min}, k, n_{lb}) = -\infty$ then incentivizing at least n_{lb} agents to improve is impossible. Function S is defined as follows.

- 1) For any $\tau \in \mathcal{T}_p, \eta \geq 1$, we have $S(\tau, 0, \eta) = -\infty$.
- 2) For any $1 \leq \kappa \leq k, \eta \geq 1$, $S(\tau_{\max}, \kappa, \eta) = -\infty$, where $\tau_{\max} = \max\{\tau \in \mathcal{T}_p\}$. This holds since no agents can improve to τ_{\max} , however at least η agents to the right of τ_{\max} need to improve which is a contradiction.

- 3) For any $\tau \in \mathcal{T}_p, 0 \leq \kappa \leq k, \eta \leq 0, S(\tau, \kappa, \eta) = T(\tau, \kappa)$ where function T is defined in Algorithm 1.
- 4) For any $\tau \in \mathcal{T}_p, \tau < \tau_{max}, 1 \leq \kappa \leq k,$ and $1 \leq \eta \leq n$:

$$S(\tau, \kappa, \eta) = \max_{\substack{\tau' \in \mathcal{T}_p \\ s.t. \tau' > \tau}} \left(S(\tau', \kappa - 1, \eta - 1 [i \mid \tau \leq p_i < \tau' \text{ s.t. } \tau' - p_i \leq \Delta_i]) + \sum_{\substack{\tau \leq p_i < \tau' \\ s.t. \tau' - p_i \leq \Delta_i}} (\tau' - p_i) \right)$$

$S'(\tau, \kappa, \eta)$ keeps track of the optimal set of targets corresponding to $S(\tau, \kappa, \eta)$.

E.2 Extension 2: Optimizing the number of target levels

The nonmonotonicity property may make adding a new target level to the current placement reduce the maximum improvement (see Figure 1b), or wasteful if we place the new target level somewhere no agent can reach or on top of an existing target. Therefore, when considering $k = 1, 2, \dots, n$, it is possible that the maximum total improvement is achieved at $k < n$. Using the dynamic program based on Algorithm 1 we can find the minimum value of k that satisfies this property and minimizes the number of targets subject to achieving maximum total improvement. Furthermore, by finding the total amount of improvement for different values of k , the principal can decide how many targets are sufficient to achieve a desirable total improvement (bi-criteria objective).

E.3 Open Problem: Tightening the approximation gap

Algorithm 3, as stated in Theorem 7, provides an $\Omega(1/g^3)$ -approximation simultaneous guarantee compared to the optimal solution for each group using at most k targets; and as stated in Theorem 20, provides an $\Omega(1/g^2)$ -approximation simultaneous guarantee compared to the optimal solution for each group using at most $\lceil k/g \rceil$ targets. Example 9, on the other hand, shows an instance where no solutions with $> 1/g$ simultaneous approximation for the groups is possible for either of the benchmarks. Therefore, there is a gap of $\mathcal{O}(g^2)$ for the first, and a gap of $\mathcal{O}(g)$ for the second benchmark. Finding the optimal order of approximation guarantees for these benchmarks and tight lower bounds are the main problems left open by our work.

Screening with Disadvantaged Agents

Hedyeh Beyhaghi¹ ✉ 

Carnegie Mellon University, Pittsburgh, PA, USA

Modibo K. Camara ✉ 

University of Chicago, IL, USA

Jason Hartline ✉ 

Northwestern University, Evanston, IL, USA

Aleck Johnsen¹ ✉ 

Geminus Research, Cambridge, MA, USA

Sheng Long ✉ 

Northwestern University, Evanston, IL, USA

Abstract

Motivated by school admissions, this paper studies screening in a population with both advantaged and disadvantaged agents. A school is interested in admitting the most skilled students, but relies on imperfect test scores that reflect both skill and effort. Students are limited by a budget on effort, with disadvantaged students having tighter budgets. This raises a challenge for the principal: among agents with similar test scores, it is difficult to distinguish between students with high skills and students with large budgets.

Our main result is an optimal stochastic mechanism that maximizes the gains achieved from admitting “high-skill” students minus the costs incurred from admitting “low-skill” students when considering two skill types and n budget types. Our mechanism makes it possible to give higher probability of admission to a high-skill student than to a low-skill, even when the low-skill student can potentially get higher test-score due to a higher budget. Further, we extend our admission problem to a setting in which students uniformly receive an exogenous subsidy to increase their budget for effort. This extension can only help the school’s admission objective and we show that the optimal mechanism with exogenous subsidies has the same characterization as optimal mechanisms for the original problem.

2012 ACM Subject Classification Applied computing → Economics; Theory of computation → Algorithmic mechanism design

Keywords and phrases screening, strategic classification, budgeted mechanism design, fairness, effort-incentives, subsidies, school admission

Digital Object Identifier 10.4230/LIPIcs.FORC.2023.6

Related Version *Full Version:* <https://arxiv.org/abs/2305.18068>

Funding *Jason Hartline:* Supported in part by NSF CCF 1934931.

Sheng Long: Supported in part by NSF CCF 1934931.

1 Introduction

Screening is a problem in which a *principal* desires to select only a qualified sub-population of *agents* who exceed an appropriate threshold applied to the agents’ *private types*.

Many real-world problems may be interpreted as special cases of screening problems, including some well-studied problems in standard frameworks (for example within auction design, how to give away an item to an agent who values it the most). As further examples: school admissions, hiring employees, selecting romantic partners, identifying winners of

¹ Corresponding authors



prestigious awards, qualifying applicants for government-issued licenses, assigning school grades at any level of evaluation (from homework grades to testing grades to overall-course grades), drug-testing, tournament-qualifying, . . . , all of these and many more scenarios may be modeled as problems of screening.

The challenge of screening is that the principal has only indirect access to the agents' private types, and critically, the agents are either unwilling to reveal their types or are incentivized to take actions that make it difficult for the principal to infer their types. Since these agents are *strategic*, their private information is only fully or partially elicited by offering appropriate incentives.

This paper considers a screening model of *school admissions* where some students may be disadvantaged relative to others. The school seeks the most skilled students but only has access to an imperfect measure of skill, via test scores. Relative to their inherent skill, disadvantaged students may perform worse on tests because they have less time to prepare (e.g., due to work obligations or childcare). Advantaged students may perform better on tests relative to their skill because they have access to additional resources (e.g., a private tutor or test prep). We model this heterogeneity by assuming that applicants are distinguished both in their skill as well as their *budget* (i.e., how much time and resources they are able to put towards the test). Students with high skill and high budget are able to excel, provided that they are willing to put in the effort. However, students with similarly high skill may test poorly if their budgets are too low.

More precisely, we study a mechanism design problem for screening of budgeted agents. A principal is interested in admitting only an agent with high skill-type above a given threshold. The agent can only reveal private skill-level to the principal indirectly, by combining it with an amount of effort into a publicly displayable signal of quality. However, the agent is limited by a budget on effort, which induces a key difficulty for the principal: amongst agent types exhibiting similar-quality signals, how to distinguish between talented agents with high-skill-low-budget types and endowed agents with low-skill-high-budget types, while contending with agents' incentive-compatibility constraints.

A key observation from the model is that it may be beneficial to admit students with average test scores with nonzero probability, while at the same time always admitting students with the highest test scores. By not guaranteeing admission for students with average test scores, we limit the incentive for those students to put in effort. High-skill agents (regardless of their budget) find effort less costly than low-skill students; therefore, as we decrease the probability of admission, the low-skill students will reduce their effort more sharply than high-skill students. Loosely speaking, if we lower the probability of admission enough for students with average test scores, the equilibrium level of effort will drop until the high-skill disadvantaged students' budget constraint is no longer binding. This allows the school to screen efficiently, at the cost of admitting high-skill students at a lower rate.

As a result, these randomized admission policies make it possible to implement a counter-intuitive outcome. A student with high skill but low maximum test score (due to limited budget) can receive strictly larger allocation than a student with low skill but high maximum test score. The latter student is able to achieve scores that are strictly higher than the former student can achieve, but the benefit of obtaining those scores (some probability of admission) is not worth the effort for a low-skill student.

Our main result formalizes this intuition. It gives (1) a characterization of the structure of the optimal mechanism for a (one-agent) setting with 2 skill types and n budget types, and (2) a polynomial-time algorithm to find it. An interpretation of our main result is that high-skill agent types may be shown *preference* over high-budget types despite the difference in the types' *exogenous* resources. Thus, our setting effectively studies the possibilities and limits of improved-welfare of allocation to effort-budgeted agents.

The paper ends with an introductory study of an extended setting which introduces *uniform, exogenous, unconditional subsidies* to relax the agents' budget constraints.² Intuitively, the goal is to modify the environment of the admissions problem (as screening) to further increase the balance of allocation in favor of high-skilled types. Subsidies are a potent intervention because high-skill, budget-constrained agents are best able to use additional effort to increase their highly-valued allocations. We show that the setting with subsidies has optimal mechanisms with the same characterization as the original screening problem.

Related Works

Previous literature has varied its modeling of this central challenge of screening. [27] models agents as having private abilities (types) that the market doesn't observe, and agents with higher abilities have economic incentives to be identified. [26] studied the role of interest rates as a screening device, and showed that returns are not necessarily monotone with respect to interest rates – a result that holds in equilibrium whenever borrowers *strategically* react to the interest-rate mechanism.

In addition to the economics literature on screening, this work contributes to ongoing research on strategic classification, mechanism design with budgeted agents, and fairness.

There is a well-developed literature on mechanism design where agents face budget constraints. Earlier work focused on the case where budgets were public knowledge (e.g., [20, 21]). More recent work, like ours, focuses on the case where the agents' budgets are their private knowledge (e.g., [24, 11, 9]). Typically, budgets are monetary: they represent upper bounds on how much each agent can transfer to the principal. In contrast, we consider budgets on effort: upper bounds on how much effort the agent can put into its task.

In recent years, there has been a lot of interest in strategic classification problems, where a principal is trying to classify agents on the basis of observed scores and agents are able to manipulate (or “game”) the scores to influence the principal's actions [13, 8, 15, 23, 1, 6, 10, 5, 18, 14, 3, 28, 22, 12, 4, 2]. Our model can be considered a strategic classification problem where the school attempts to classify students into “admit” or “not admit”, but students are strategic in how much effort they put in. The closest to our work are [15] and [5]. [5] show the power of randomization when agents are able to manipulate their scores, while [15] study a similar problem where disadvantaged students find it more difficult to manipulate their scores.

In most models of strategic classification, agents obscure their true type at a cost. As a result, costly effort makes scores less informative. In contrast, in our model of screening, even high skill students need to put in effort in order to achieve a high score (albeit less effort than low-skill students). If no students put in effort, they will all achieve a score of zero, and the school will not be able to distinguish high-skill from low-skill students. As a result, costly effort is necessary for scores to be informative in our model. We must balance the benefits of costly effort in screening with the challenges of costly effort in strategic classification.

Finally, our work relates to a growing literature on fairness in mechanism design and algorithms. Much of this literature is concerned with fair treatment of different subgroups (e.g., based on demographic variables like race or gender), and various different definitions of fairness have been proposed and criticized (see e.g., [7, 19]). Some of this work, like ours, has been explicitly applied to school admissions (e.g., [17]). In line with the fairness literature,

² Subsidies, measured in units of effort, can for example be monetary transfers from third-parties that increase an agent's effort-budget by freeing up time by reducing other paid work or by buying services.

we consider the implications of a biased test (where high budget students may perform better, regardless of their skill) for admissions. Unlike race and gender, the subgroups we are interested in (students with a particular budget) are not publicly observable. Like [25] and [16], we explicitly consider how economic incentives interact with policies designed to correct for sources of unfairness.

2 Setting and Fundamental Structures

A principal P considers admitting an agent $A = (s, b)$ with private types as skill s and budget b (budget on *effort*, see below). The agent's skill and budget are treated as independent, positive Bayesian variables drawn respectively from known distributions S with support $\mathbf{S} \in \mathbb{R}_+$ and B with support $\mathbf{B} \in [0, 1]$, i.e., $s \sim S$ and $b \sim B$. The principal only wants to admit the agent in the case that the agent's skill is above a threshold $\tau \in \mathbb{R}_+$ (which we implicitly treat as the principal's fixed type). In summary, the principal's problem is an admission game $\mathcal{G} = (S, B, \tau)$.

An agent of any skill will want to be admitted and thus, the principal must design a test which uses incentives to elicit information from the agent. The principal will ask the agent to commit to a *private* level of effort e which (a) is constrained by individual budget b , and (b) induces a *public*, deterministic signal of quality $q = s \cdot e$. Note that quality is a multiplicative function of effort, rather than additive. This captures two features of our motivating example of school admissions: (i) even high-skill agents that put in no effort will obtain a low score, but (ii) high-skill agents require less effort to achieve a given score than low-skill agents.³

The principal's problem will be to design an admission allocation rule $y : \mathbb{R}_+ \rightarrow [0, 1]$ which maps quality q to a stochastic allocation x of admitting the agent. Practically, the principal's challenge is to optimally discriminate against resource-rich agent types with quality resulting from large effort-budgets, in favor of agents with quality resulting from high skill. (Note, any "reasonable" rule will inherently admit all high-skill-high-budget agents.)

The agent's utility is defined to be $-\infty$ if effort exceeds budget, and otherwise is defined to be the probability of allocation minus effort:

$$u_A(e, x) = x - e \tag{1}$$

which implicitly sets the agent's value of being admitted to 1. Utility can be equivalently written as a function of the allocation rule y and either effort or quality:

$$u_A(y, e) = y(s \cdot e) - e \quad \text{or} \quad u_A(y, q) = y(q) - q/s \tag{2}$$

(Further, we may drop the input y where its assignment is clear from context.) The agent perceives the allocation rule y as a menu (for which the domain is quality space), albeit top-truncated at the agent's maximum quality set by $q^\dagger = s \cdot b$. This perspective induces for the agent an optimal utility function u_A^* and an allocation rule x in skill space (which overloads notation):

$$u_A^*(y, s) := \max_{e \in [0, b]} y(s \cdot e) - e = \max_{q \in [0, q^\dagger]} y(q) - q/s \tag{3}$$

$$x = x(y, s) := y(s \cdot \left[\operatorname{argmax}_{e \in [0, b]} y(s \cdot e) - e \right]) \tag{4}$$

³In contrast, suppose quality were an additive function $q = s + e$ of skill and effort. Then property (ii) would hold, but not property (i).

For a given agent A , the principal's utility from admitting A is $u_P(A \mid \text{admitted}) = s - \tau$. Thus, our principal's mechanism design problem is to maximize $u_P(\mathcal{G}, y)$ which is the expected utility from an admitted agent's skill versus the threshold, weighted by allocation probability:

$$\max_y u_P(\mathcal{G}, y) := \max_y \mathbf{E}_{A \sim (S \times B)} [x(y, s) \cdot (s - \tau)] \quad (5)$$

Threshold Mechanisms

A natural mechanism to consider is a *threshold mechanism* with the threshold set in quality space.

► **Definition 1.** A (deterministic) threshold mechanism $y^{q'}$ sets a quality threshold $q' \in \mathbb{R}_+$ and admits an agent if and only if the agent exhibits public quality $q \geq q'$.

The intuition for a threshold mechanism is that an agent who is able to exhibit the threshold quality with effort less than budget will put in the (minimal amount of) effort necessary to be admitted with probability 1; versus, an agent with maximum-quality q^\dagger less than the threshold will put in zero effort and get passed. Recall that the agent's skill and budget are independent in our setting. The role of thresholds generally is to conditionally allocate agents in decreasing order of skill:

► **Fact 2.** Given a population of agents as $S \times B$, consider the subset $\mathbf{B}_{\bar{b}}$ of agent types which conditionally have a specific budget \bar{b} . For a threshold mechanism with any $q' > 0$, the subset of $\mathbf{B}_{\bar{b}}$ of agent skill-types which are admitted is upward-closed.

Fact 2 implies that threshold mechanisms are sufficient for the special case in which there is only one budget type (with the proof of Proposition 3 in Appendix A.1):

► **Proposition 3.** Assume that an agent has constant budget \bar{b} on effort, i.e., the distribution B is a singular point mass. The threshold mechanism $y^{q'}$ with $q' = \tau \cdot \bar{b}$ is optimal.

Intuitively, Proposition 3 holds because single-budget is a simple setting in which quality-thresholds directly implement skill-thresholds, in particular for the principal's threshold τ .

To outline this section: Section 2.1 shows that threshold mechanisms are not optimal for arbitrary distributions B and thus, we will need more-complicated mechanism forms. Section 2.2 quantifies agent feasibility to achieve a given quality-allocation pair and, given an allocation rule y , discusses implications of feasibility for optimal design. Section 2.3 gives geometric interpretation of agent types (s, b) and their demand under an allocation rule y (for input as quality q).

2.1 Generalization of Threshold Mechanisms to Lottery Menus

This section states that deterministic threshold mechanisms are *not optimal* in general (when the distribution over budgets has multiple support). Consequently, we need to generalize the class of mechanisms being considered. This section gives the sufficient extension to *lottery menus* (Definition 6 below).

Insufficiency of deterministic thresholds is stated simply:

► **Proposition 4.** For admission games \mathcal{G} in which the set of budgets is multiple, i.e. $|B| > 1$, (deterministic) threshold mechanisms are not optimal generally.

The proof is by counter-example – we give the details and analysis of Example 28 in Appendix A.2 where we conclude that all deterministic threshold mechanisms are dominated by stochastic allocation $x = (1 - \epsilon_L/1 + \epsilon_L - \epsilon_x)$ with “small” ϵ_x for agents exhibiting at least a minimum quality.

Although we must now consider allocation rules y more generally than threshold mechanisms, without loss of generality, we may assume monotonicity of y :

► **Lemma 5 (Monotonicity).** *For every admission game \mathcal{G} , there exists an optimal allocation rule that is weakly monotone increasing.*

Proof. For every allocation rule \tilde{y} that is strictly decreasing somewhere on its domain, the principal gets the same utility from the “ratcheted” allocation rule $\bar{y}(\tilde{y})$ which increases the allocation in every decreasing region of \tilde{y} to be equal to the left end point of the region, i.e., flat on the region. (The resulting $\bar{y}(\tilde{y})$ is weakly monotone increasing.)

Principal utility is the same for \bar{y} and \tilde{y} because every agent $A = (s, b)$ gets the same allocation: all qualities q where $\bar{y}(q) \neq \tilde{y}(q)$ are ignored because they are dominated for both functions by the “ratchet point”-quality (a weakly larger allocation requiring strictly less effort is preferred). ◀

Thus, in order to identify the optimal mechanism, we propose lottery menus:⁴

► **Definition 6 (Menu).** *A lottery menu mechanism is a (weakly) monotone allocation rule y with menu options $(q, x = y(q))$, where x is the allocation probability for an agent exhibiting quality q .*

2.2 Leveraging Agent Feasibility to Improve Screening

This section formalizes the feasibility for an agent to choose a given menu option. Subsequently, this section explains how lottery menus effectively leverage feasibility to promote the principal’s objective: decreasing allocation necessarily discriminates in favor of higher-skill agents (summarized below as Proposition 9; note, we can already observe this effect working in Example 28).

Feasibility is due to (a) the budget constraint, and (b) a non-negative utility requirement:

► **Definition 7.** *Menu option (q, x) is feasible for agent $A = (s, b)$ if:*

1. (affordability) *minimal effort $e^* = q/s$ (to achieve quality q) is at most b , i.e., $e^* \leq b$; and*
2. (rationality) *(q, x) induces non-negative utility for A , i.e., $u_A(e^*, x) = x - e^* \geq 0$.*

► **Fact 8.** *Menu option (q, x) is feasible for agent $A = (s, b)$ if and only if $q/s \leq \min\{b, x\}$. Upon choosing this option, A achieves utility $u_A(q) = x - q/s$.*

Fact 8 implies that we can use stochastic (partial) allocation to improve the principal’s expected utility by discouraging a low-skill agent from applying. Consider two agents described qualitatively as: high-skill-low-budget (A_H) and low-skill-high-budget (A_L), where we naturally prefer to admit A_H . Intuitively, we decrease x for a fixed \bar{q} , we get the following effects: (a) for larger b , the upper bound on \bar{q}/s is set by x “sooner” (as it decreases, rather than by budget); and (b) rationality is violated for A_L before it is violated for A_H . Both effects (a) and (b) threaten A_L ’s utility. We state this formally as a *ceteris paribus* result, where dependence on feasibility is clear in the proof:

⁴ If we consider admitting multiple agents drawn independently from $S \times B$ and our utility is (independently) additive across decisions, it may be possible to negatively correlate admission decisions to target the total number of admits. For example, if our setting is discrete and we choose an allocation rule y , if k_q agents apply with the same quality q , we may decide to run a *lottery* which admits exactly $1/y(q)$ of the agents uniformly at random.

► **Proposition 9** (The Lotteries-in-Screening Proposition). *For a fixed quality \bar{q} , decreasing the allocation $y(\bar{q})$ when an agent exhibits quality \bar{q} increases the lower bound on the skill of agents who feasibly choose $(\bar{q}, y(\bar{q}))$.*

Proof. The agent's utility is the difference between allocation and effort: $y(\bar{q}) - e$. Utility is 0 for a marginally-skilled agent with skill s^* who must put in effort $e^* = y(\bar{q})$ to achieve quality \bar{q} . We also have the abstract definition: $q = s \cdot e$. Substituting from the definition, we have $y(\bar{q}) = \bar{q}/s^*$. The quality \bar{q} is fixed, thus decreasing the left-hand side requires increasing the skill threshold s^* . ◀

2.3 Geometric Interpretation

This section introduces geometric interpretations of the problem (that will be useful for our analysis of optimal mechanisms). The first of these visualizations is graphical representation of an agent's feasible allocations. Regions of feasibility map directly onto a graph of an allocation rule y which has quality space as its domain and allocation as its output. As exhibited in Figure 1(Top) which gives two graphic examples of these regions, we have the following geometric observations:

- **Fact 10.** *An agent A with skill s (ignoring budget and affordability):*
 - *is partially identified by a ray out of the origin with slope $1/s$; this ray necessarily lower-bounds A 's feasible region because this is the zero-utility line, i.e., points (q, x) on this line result in A achieving utility of 0;*
 - *who chooses a menu option (q, x) – independent of being rational or not – will get utility equal to the vertical difference between the chosen allocation x and the height $q \cdot (1/s)$ of the zero-utility line at q (which directly interprets from definitions: $u_A(q) = x - q/s$).*

From the points of Fact 10, agent types partitioned by skill $s_i \in \mathbf{S}$ are identified with their respective zero-utility lines. We illustrate this in Figure 1(Bottom) by expanding its (Top)graphics to show a setting with two skill types: low skill s_L and high skill s_H . Within this context, we give formal definitions:

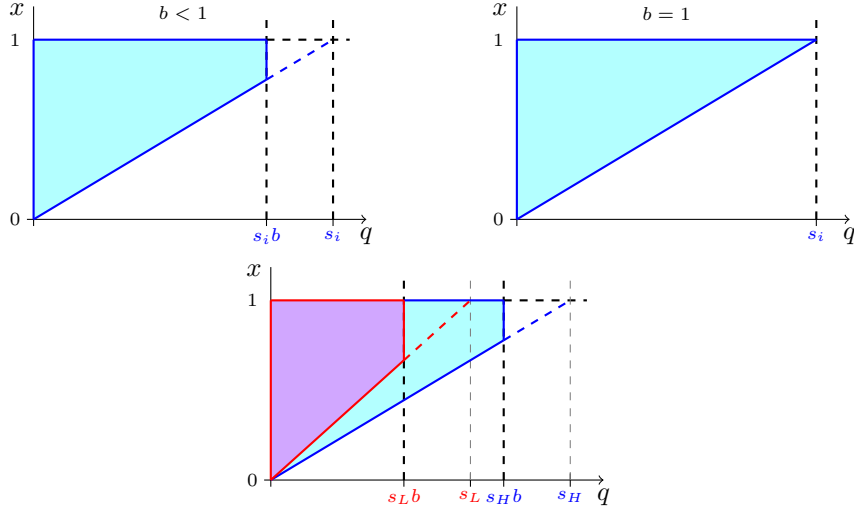
- **Definition 11.** *The low-skill agents' line is their zero-utility line with slope $1/s_L$ on the (quality, allocation) graph for (budget-unconstrained) low-skill agents. Similarly, the high-skill agents' line is their zero-utility line with slope $1/s_H$. Generally, we refer to zero-utility lines as skill lines.*

3 The Optimal Mechanism for 2-skill, Discrete-budget Types

This section solves the discrete-type setting for a principal with skill threshold τ and a stochastic agent $A = (s, b)$ with type-space defined by two skill-types with $s_L < \tau < s_H$ and n budget-types with $0 < b_1 < b_2 < \dots < b_n$. Due to the discrete type-space, the optimal mechanism may not be unique. Theorem 13 is sufficient to identify an optimal mechanism, which is a *slanted-stair function*:

- **Definition 12.** *A slanted-stair function $f : \mathbb{R}_+ \rightarrow [0, 1]$ (as an allocation rule) has $f(0) = 0$; and is a weakly increasing function that begins as a sequence of line segments that all have the same (constant), positive derivative. Each line segment has open lower bound and closed upper bound. (The function's output must reach 1 and is identically 1 for larger inputs.)*

We refer to the line segments as slanted-steps. We refer to the (necessarily positive) vertical gaps between slanted-steps as jumps.



■ **Figure 1** (Top) A menu option is a point (q, x) with coordinates respectively from quality space \mathbb{R}_+ and allocation space $[0, 1]$. The blue regions are feasible for agent $A = (s_i, b)$, i.e., A can select these menu options (when they exist) and achieve non-negative utility. The regions' lower-bound line has slope $1/s_i$. (Bottom) The red region is feasible for an agent $A_L = (s_L, b)$. The blue region (which entirely encompasses red) is feasible for agent $A_H = (s_H, b)$. Regarding discussion of Proposition 9 in Section 2.2, observe how for fixed quality set by $q = s_L \cdot b$, it is possible to use decreased allocation awarded to a fixed quality (at/below the vertical boundary between red and blue regions), in order to exclusively admit a high-skill agent.

For a set of types T , let $\Delta(T)$ be the probability simplex over the elements of T . Before giving our main result, we state an interesting observation: there will be nothing in the proof of Theorem 13 that requires the independence of S and B . Thus to state a stronger main result, we define a *correlated* admission game by $\mathcal{H} = (S, \mathcal{B}^S, \tau)$ where \mathcal{B}^S is a set of conditional budget-distributions: one budget-distribution corresponding to each skill-type with positive support in S .⁵

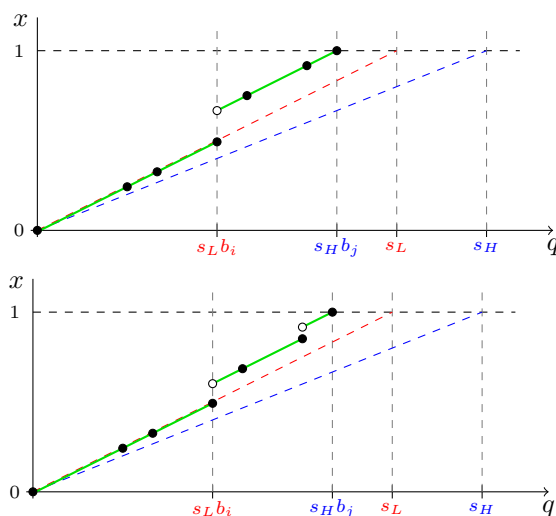
Agents in a correlated game have the same description as in the original, independent game. By contrast, the principal's objective must be updated to reflect the correlation:

$$\max_y u_P(\mathcal{H}, y) := \max_y \mathbf{E}_{A \sim (S, \mathcal{B}^S)} [x(y, s) \cdot (s - \tau)] \quad (6)$$

► **Theorem 13 (Main Result).** *Given a correlated admission game $\mathcal{H} = (S = \Delta(\{s_L, s_H\}), \mathcal{B}^S = \{B_{s_L} = \Delta_{s_L}(\{b_1, \dots, b_n\}), B_{s_H} = \Delta_{s_H}(\{b_1, \dots, b_n\})\}, \tau)$ with $0 < s_L < \tau < s_H$ and $0 < b_1 < b_2 < \dots < b_n$. There exists an optimal mechanism y^* for the correlated admission game \mathcal{H} that is a slanted-stair function f with constant slope equal to $1/s_L$ and at most one jump, and with:*

1. the region of the first slanted-step characterized by: equality to the low-skill agents' zero-utility line;
2. the quality-index at which f jumps q^{jump} (if it exists) characterized by: occurring either at quality $q_0 = 0$, or occurring at some maximum-possible quality exhibited by some low-skilled agent $A_{L,i} = (s_L, b_i)$, i.e., at some $q^{\text{jump}} = q_{L,i}^\dagger = s_L \cdot b_i$;

⁵ Assuming discrete budget-distributions, while elements of \mathcal{B}^S may have distinct support, it is without loss of generality to assume that they all have common, enumerated support b_1, \dots, b_n because any locally-unused budget type b_i can be locally assigned probability 0.



■ **Figure 2** (Top) A one-jump, slanted-stair allocation curve y (solid green) with $q^{\text{jump}} = s_L \cdot b_i$ and $q^{x=1} = s_H \cdot b_j$. The black dots are an example of discrete menu options. Recall that agent utility is interpretable as the vertical difference between allocation and (zero-utility) skill line. Any low-skilled agent $A_L = (s_L, b_k)$ with $q_{L,k}^\dagger = s_L \cdot b_k \leq s_L \cdot b_i$ will choose menu option $(0, 0)$ (per the tie-breaking rule, see Definition 18). Any low-skilled agent with $q_{L,k}^\dagger = s_L \cdot b_k > s_L \cdot b_i$ will choose $(s_L b_i + \epsilon, y(s_L b_i + \epsilon))$ with $\epsilon \rightarrow 0$. Each high-skilled agent $A_H = (s_H, b_k)$ with $k < j$ will achieve maximum quality $q_{H,k}^\dagger = s_H \cdot b_k < s_H b_j$; and those with $k \geq j$ will achieve quality $s_H b_j$ (and are allocated with probability 1). (Bottom) A two-jump, slanted-stair allocation curve y (solid green).

3. the region of the second slanted-step (if it exists) characterized by: the quality at which f intersects the allocation-of-1 horizontal line is the maximum-possible quality exhibited by some high-skilled agent $A_{H,j} = (s_H, b_j)$, i.e., at some $q^{x=1} = q_{H,j}^\dagger = s_H \cdot b_j$.

(Note, optimal assignment of mechanism parameters and the given characterizations of Theorem 13 are sufficient to identify the height of the vertical jump, starting from the low-skill agents' line.)

The proof of Theorem 13 depends on a sequence of lemmas which we state at the end of this section. The proofs of Theorem 13 and its supporting lemmas appear in the main version of the paper. Graphically, the optimal menu (which may be discrete, corresponding to our discrete setting) will qualitatively have the single-jump structure of Figure 2(Top) with menu options on only two line segments (as two slanted-steps). Multi-jump structures are precluded, such as the three-slanted-steps in Figure 2(Bottom).

The statement of Theorem 13 induces the following corollary regarding the polynomial running time of a brute-force algorithm that searches over the possible combinations of jump-points and jump-heights, which is sufficient to find the optimal algorithm of the statement's setting.

► **Corollary 14** (Running Time). *Given a correlated admission game $\mathcal{H} = (S = \Delta(\{s_L, s_H\}), \mathcal{B}^S = \{B_{s_L} = \Delta_{s_L}(\{b_1, \dots, b_n\}), B_{s_H} = \Delta_{s_H}(\{b_1, \dots, b_n\})\}, \tau)$ with $0 < s_L < \tau < s_H$ and $0 < b_1 < b_2 < \dots < b_n$ and – per Theorem 13 – the sufficient, discrete search space for an optimal algorithm.*

The optimal mechanism may be identified by a brute-force search over the $O(n^2)$ unknown combinations of parameters of the optimal characterization (Theorem 13). The time to evaluate each allocation rule (resulting from a combination of parameters) also runs in polynomial time.

3.1 Discussion of Optimal Characterization in Theorem 13

Having a characterization of optimal mechanisms, we would like to understand qualitatively their performance. We will discuss two dimensions of efficacy: (1) mechanism performance, of course, as the originally-defined objective; and (2) *fairness*, which informally is a measurement of how well outcomes-per-agent-type conform to some definition of what outcomes the agents *arguably deserve*, specifically compared to other agents' type-outcome pairs.

Regarding mechanism performance, we know that the single-jump, slanted-stair characterization of Theorem 13 improves on the (deterministic) threshold mechanisms of Definition 1 which are not generally optimal (by Proposition 4), except for games with convenient distributions S and B (e.g., Proposition 3). On the other hand, optimal mechanism performance still falls short of the *offline optimal* benchmark which has full information and which is generally unachievable; rather, we may use it as a reference mechanism to which we compare performance:

► **Definition 15.** *Given a stochastic agent $A = (s, b)$, the offline optimal mechanism for a principal requiring skill-threshold τ – which is assumed to know the realized skill type of the agent as $\hat{s} \sim S$ – admits the agent if $\hat{s} > \tau$ and only if $\hat{s} \geq \tau$; and this admission decision is independent of the agent's realized budget type $\hat{b} \sim B$.*

The offline optimal mechanism is unconditionally optimal, as it fully allocates every agent with skill above the threshold and fully rejects every agent with skill below it. In order to increase the performance of mechanisms beyond what is possible from Theorem 13 – i.e., from standard mechanism design subject to agents' incentive compatibility constraints – in Section 4 we consider a modified admission problem in which the agent may have exogenous access to a *subsidy*.

Regarding fairness, we first must consider the philosophical concept of what comparisons between distinct agents' type-outcome pairs may arise as fair or as unfair within the parameters of our model (Section 2). Loosely summarizing: our agents independently have higher or lower skills and higher or lower budgets; and by best-responding to a given allocation rule based on skill and budget, agents are consequently admitted with larger or smaller probability. Reasonably, agent “skill” is positively correlated with student value and agent budget is independent, so we posit that higher skill types are *more-deserving* of being admitted than lower types, independent of budget. Moreover, the degree of worthiness should increase with increasing *cardinal* difference in skill types.

Thus, we consider the following concept of fairness: regardless of budget, larger (admission) allocations given to lower-skilled agents are comparatively judged to be unfair outcomes as the higher-skilled type is more-deserving; and the larger the skill-difference, the larger the unfairness. Furthermore the strict contrapositive also holds: comparatively larger allocations given to higher-skilled agents are more fair. However, the choice of function used to measure technically the unfairness of an allocation rule remains debatable.

From the following intuition, the mechanism design problem of our admission-game model should be positively aligned with objectives resulting from our concept of fairness. First, recall the principal's utility from admitting an agent A , which is $u_P(A \mid \text{admitted}) = s - \tau$. Given this utility function, the principal has a precise, cardinal utility measure over admitting agent skill-types, which has both order and cardinality aligned with fairness as desired, regardless of the technical fairness measure. I.e., the principal is incentivized to choose an allocation rule that increases fairness. In at least one sense, this is strictly true, which moreover motivates the principal's objective function itself (see equation (6)) as a formal example of fairness measure:

► **Fact 16.** *Where incentive compatibility permits, the principal is incentivized to inherently prefer that between two agent types with different skill levels, the agent type with higher skill will receive the larger allocation probability.*

► **Corollary 17.** *Given a correlated admission game $\mathcal{H} = (S = \Delta(\{s_L, s_H\}), \mathcal{B}^S = \{B_{s_L}, B_{s_H}\}, \tau)$ with $0 < s_L < \tau < s_H$ and $0 < b_1 < b_2 < \dots < b_n$. For the fair mechanism design problem which maximizes the fairness measure set equal to the principal's utility function, the optimal mechanism and characterization of optimal mechanisms are determined identically to Theorem 13.*

Second, the offline optimal mechanism can illustrate the alignment between the principal's mechanism design incentives and fairness. On one hand, offline optimal represents perfect – albeit generally unachievable – performance for the mechanism. On the other hand, by giving allocation 1 to an upward-closed set of skill-types above τ , allocation 0 to a downward-closed set of skill-types below τ , and any constant allocation to skill-type exactly τ , the allocation is arguably fair because no rejected skill-type can protest for increased allocation on the basis that it is strictly more-deserving than any admitted skill-type. Thus, the offline optimal mechanism as ideal-objective further aligns the principal and fairness.

For purposes of space, we defer discussion of a third intuitive perspective supporting the alignment of optimal mechanisms and fairness to Appendix A.3.

3.2 The Proof of Theorem 13

We need one more critical detail to set up the proof of Theorem 13. Depending on allocation rule y , an agent A may be indifferent between a set of quality-allocation menu options that are optimal for A . To address this, we define our tie-breaking rule:

► **Definition 18.** *When an agent's set of optimal menu options is multiple, the tie-breaking rule is: all agents choose the smallest menu option of the set. (Note, “smallest” is the same in either dimension of quality or allocation.)*

This tie-breaking rule is material for our results: it is sufficient to break ties optimally in favor of the principal's objective.⁶ Recalling that utility is equal to the vertical difference between the allocation and the height of the zero-utility line (Fact 10), the key effect of tie-breaking is observed in Figure 2: within a region of a single slanted-step, *low-skill agents are indifferent everywhere and choose the minimal allocation at the left endpoint of the region.* This tie-breaking rule applies for all result statements and proofs in this paper.

As an overview, the proof of Theorem 13 proceeds as a search for the optimal mechanism. This search is organized as a sequence of reductions of the search space: it starts with an allocation rule that is monotone (Lemma 5 on page 6) but is otherwise arbitrary; and then with each successive lemma, we prove that it is sufficient to restrict attention to a smaller set of allocation rules. Lemma 23 is the last reduction in the sequence and states that the optimal mechanism must be a slanted-stair function (Definition 12) with at most one jump. The final proof of Theorem 13 starts from the statement of Lemma 23 and proves the additional details in its own statement.

⁶ This tie-breaking rule is justified similarly to tie-breaking in other areas of mechanism design, e.g., in auctions with a revenue objective in which agents with value equal to price are assumed to buy, in favor of the designer's objective. Intuitively, the justification is that small perturbations to the design can achieve the same outcome within arbitrary (lossy) required precision; so instead, we simplify the analysis by allowing ties and breaking them favorably, rather than accounting for a notation-heavy perturbation.

6:12 Screening with Disadvantaged Agents

All of the following lemmas assume the same setting as the statement of Theorem 13, which is: given a correlated admission game $\mathcal{H} = (S = \Delta(\{s_L, s_H\}), \mathcal{B}^S = \{B_{s_L} = \Delta_{s_L}(\{b_1, \dots, b_n\}), B_{s_H} = \Delta_{s_H}(\{b_1, \dots, b_n\})\}, \tau)$ with $0 < s_L < \tau < s_H$ and $0 < b_1 < b_2 < \dots < b_n$.

With this overview in place, the sequence of reductions of the search space is:

► **Lemma 19** (Lower bound). *An optimal allocation rule is never under the low-skill agents' line.*

► **Lemma 20** (Strong monotonicity). *There exists an optimal allocation rule y^* that everywhere has a derivative lower bound set by $1/s_L$ (the slope of the low-skill agents' line).*

► **Lemma 21** (Constant allocation slope). *There exists an optimal allocation rule y^* that is a slanted-stair function, i.e., it everywhere has constant derivative equal to $1/s_L$ (the slope of a low-skill agents' line), allowing for arbitrary, discretely-indexed, positive, vertical jumps.*

► **Lemma 22** (A corner-case exclusion). *There exists an optimal allocation rule y^* for which the optimal menu option of the agent type with smallest maximum-quality gives 0-allocation. (This agent is $A = (s_L, b_1)$ with $q_{L,1}^\dagger = s_L \cdot b_1$.)*

► **Lemma 23** (Sufficiency of at-most one jump). *There exists an optimal allocation rule y^* that is a slanted-stair function with at most one jump; furthermore, if there is a jump in a given y^* , then its allocation in the region of the first slanted-step must be equal to the low-skilled agents' line.*

The proofs for each lemma in this sequence appear in the full version of the paper.

4 Mechanisms for Agents with Subsidized Effort

This section considers agent subsidies directly in *effort-space*. A budget on effort implies a time-constraint. Effort-subsidies are an intervention that increases the agent's effort-budget by freeing up an agent's time spent on other obligatory activities. Technically, we consider subsidies as uniform, additive increases to agents' budget constraints. These subsidies are offered *unconditionally*: agents may spend the time on an outside-option (leisure) activity; or they may invest the time in effort, which they experience as *costly* (i.e., as the opportunity cost of the forfeited leisure time). E.g., subsidies may be provided by performing time-costly tasks for agents' benefit (like uniformly offering free postal pickup/delivery) – freed from the burden of the task, agents enjoy leisure or spend their time exerting effort in our model.

The main goal of this section is to solve for the characterization of the optimal mechanism of the (modified) admission game which has expanded setting parameters that make it possible to consider a combined-question of screening and *design of unconditional subsidies*. Corollary 24 states that its characterization is the same as Theorem 13. We also show that this subsidies setting can only help the principal's objective (in Proposition 26).

4.1 The Setting with Subsidies

We add the following elements to the correlated setting of Theorem 13 (based on Section 2).

The mechanism designer may a priori offer to the agent $A = (s, b)$ an *effort-budget subsidy* d from a non-negative range, i.e., the subsidy is $d \in [D_-, D_+]$. The agent accepts the whole subsidy unconditionally and the agent's new budget is $b + d$.

It is not possible to restrict access to the subsidy to sub-classes of agent-types: not to high-skill agents and not to disadvantaged agents. The constant subsidy amount is necessarily available to each type indiscriminately because the realizations of an agent's skill/budget types are unknown at the time of the offer, i.e., at the time of subsidized-mechanism design. While we can not use uniform subsidies to discriminate directly, we will be able to improve the *principal's objective* using the following observation: given an optimal single-jump, slanted-stair allocation (as characterized by Theorem 13), note that the budget constraint binds for *all* high-skill agents receiving allocation less than 1 and they would benefit from relaxing the budget constraint; but for almost all low-skill agents, the budget constraint is not binding because their utility is constant on each slanted-step. This first-order-condition analysis suggests that high-skill agents will voluntarily convert unconditional subsidies to effort and increased allocation, whereas low-skill agents will not.

The subsidy (to increase effort-budget) is exogenous as if enacted and paid by an unrelated third party at no cost to the mechanism. E.g., in an admission problem, the school may be a city's unique, public, magnet high school. The subsidy may be paid uniformly to each eligible applicant by a citywide scholarship program which is separate from the school's admissions office but which has the money to provide the subsidy (up to D_+ per student) and *must support* a citywide goal of maximizing utility from specifically the magnet school's admissions policies. In this case, the magnet school admissions office (as our model's principal) optimizes $d \in [D_-, D_+]$ and the scholarship program must approve it.

For this Section 4, the updated correlated admission game with subsidies is given by $\mathcal{D} = (S, \mathcal{B}^S, \tau, D_-, D_+)$. For a given subsidy $d > 0$, agent $A = (s, b)$ has maximum quality $q^\ddagger = s \cdot (b + d)$, which is larger than the maximum quality without the subsidy ($q^\dagger = s \cdot b$). The agent's updated optimal utility function v_A^* and updated optimal allocation rule w in skill space – subject to allocation rule y – are:

$$v_A^*(y, s, d) := \max_{e \in [0, b+d]} y(s \cdot e) - e = \max_{q \in [0, q^\ddagger]} y(q) - q/s \quad (7)$$

$$x = w(y, s, d) := y(s \cdot \left[\operatorname{argmax}_{e \in [0, b+d]} y(s \cdot e) - e \right]) \quad (8)$$

In equation (8), note that because the subsidy is unconditional, the agent pays the full cost of effort e , including the (opportunity) cost of effort above the original budget b .

For a given agent A , the principal's utility from admitting A remains the function $u_P(A \mid \text{admitted}) = s - \tau$. Thus, the principal's updated mechanism design problem is to maximize $v_P(\mathcal{D}, y, d)$ which is the expected utility from an admitted agent's skill versus the principal's threshold τ , weighted by allocation probability according to w (which accounts for the subsidy):

$$\max_{y, d \in [D_-, D_+]} v_P(\mathcal{D}, y, d) := \max_y \mathbf{E}_{A \sim (S, \mathcal{B}^S)} [w(y, s, d) \cdot (s - \tau)] \quad (9)$$

4.2 Results with Subsidies

The main result of this section is: the optimal mechanism when agents have access to unconditional subsidies has the same characterization as the original game, as described in Theorem 13. Moreover, we are immediately ready to state and prove it as a corollary:

► **Corollary 24.** *Given a correlated admission game with subsidies $\mathcal{D} = (S = \Delta(\{s_L, s_H\}), \mathcal{B}^S = \{B_{s_L} = \Delta_{s_L}(\{b_1, \dots, b_n\}), B_{s_H} = \Delta_{s_H}(\{b_1, \dots, b_n\})\}, \tau, D_-, D_+)$ with $0 < s_L < \tau < s_H$ and $0 < b_1 < b_2 < \dots < b_n$. The structure of the optimal mechanism has the same characterization as the standard game, as given in Theorem 13.*

Proof. As part of identifying the optimal mechanism – according to equation (9) – the designer selects an optimal assignment to the subsidy variable $d \in [D_-, D_+]$.

Consider an optimal assignment d^* (any element of the argmax is fine). The optimal mechanism associated with d^* must be the same as the optimal mechanism for an alternative game \mathcal{D}' which sets parameters S, \mathcal{B}^S, τ to be the same as \mathcal{D} , but which assigns the endpoints of allowable subsidies to both be d^* , i.e., \mathcal{D}' has $D_- = D_+ = d^*$.

This corollary then follows directly from Lemma 25(2) below. \blacktriangleleft

While Corollary 24 is sufficient to give us characterization, it does not give us an algorithm to find the optimal mechanism because it uses theoretical existence of the optimal subsidy d^* without identifying it.

The following observations regarding correlated admission games with subsidies are straightforward. Omitted proofs in this section appear in the full version of the paper.

► **Lemma 25.** *A correlated admission game with subsidies is $\mathcal{D} = (S = \Delta(\{s_L, s_H\}), \mathcal{B}^S = \{B_{s_L} = \Delta_{s_L}(\{b_1, \dots, b_n\}), B_{s_H} = \Delta_{s_H}(\{b_1, \dots, b_n\})\}, \tau, D_-, D_+)$. Consider arbitrary \mathcal{D} , i.e., consider its inputs as variables.*

1. *Without loss of generality, we may reduce \mathcal{D} to a correlated game \mathcal{D}' which has $D'_- = 0$.*
2. *A game \mathcal{D} fixing an exact subsidy by setting $D_- = D_+$ is equivalently described by a game $\mathcal{H}_{\mathcal{D}}$ and thus is characterized by the statement of Theorem 13.*
3. *If $D_- = 0$, then expanding the original correlated admission game $\mathcal{H} = (S, \mathcal{B}^S, \tau)$ to consider admissions with subsidies – formulated as the updated game \mathcal{D} – can only increase the utility of the principal.*
4. *Given S, \mathcal{B}^S, τ , there exists a minimal subsidy upper bound D_+^m such that for all $D_+ \geq D_+^m$, the optimal mechanism achieves the offline optimal performance (see Definition 15), i.e., it is able to perfectly discriminate between high-skill and low-skill agent types regardless of their budgets.*

Lemma 25(3) is fairly obvious: if $D_- = 0$, then the principal has the option of “free disposal” of the subsidy-variable and can do no worse than the game without subsidies. The more interesting statement is that the principal’s objective can only improve for $D_- > 0$ generally:

► **Proposition 26.** *For arbitrary $D_- \geq 0$, expanding the original correlated admission game $\mathcal{H} = (S, \mathcal{B}^S, \tau)$ to consider admissions with subsidies can only increase the utility of the principal.*

In the proof of Proposition 26, we consider specifically the subsidy $d = D_- > 0$ and (deterministically) transform the optimal allocation rule without subsidies into a new allocation rule with weakly larger performance given the uniform, unconditional agent’s budget-subsidy D_- . In particular in comparison to the optimal allocation without subsidies, the new allocation gives all low-skill agent-types weakly smaller allocation, and gives all high-skill agent-types weakly larger allocation.

This new allocation rule is not necessarily optimal for its (subsidized) setting, but by dominating the original setting, its existence proves that the principal’s objective can only improve. On the other hand, the new allocation rule may harm the agents’ utilities (for any agent type, except low-skill-low-budget agents who already get 0-allocation before subsidies and who still get 0). While this assessment is not a final judgment (because the new allocation is not necessarily optimal), it is consistent with observations in [15] which showed that subsidies for disadvantaged agents might harm their utilities.

References

- 1 Saba Ahmadi, Hedyeh Beyhaghi, Avrim Blum, and Keziah Naggita. The strategic perceptron. In Péter Biró, Shuchi Chawla, and Federico Echenique, editors, *EC '21: The 22nd ACM Conference on Economics and Computation, Budapest, Hungary, July 18-23, 2021*, pages 6–25. ACM, 2021. doi:10.1145/3465456.3467629.
- 2 Saba Ahmadi, Hedyeh Beyhaghi, Avrim Blum, and Keziah Naggita. On classification of strategic agents who can both game and improve. In L. Elisa Celis, editor, *3rd Symposium on Foundations of Responsible Computing, FORC 2022, June 6-8, 2022, Cambridge, MA, USA*, volume 218 of *LIPICs*, pages 3:1–3:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICs.FORC.2022.3.
- 3 Tal Alon, Magdalen Dobson, Ariel Procaccia, Inbal Talgam-Cohen, and Jamie Tucker-Foltz. Multiagent evaluation mechanisms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(02):1774–1781, April 2020. doi:10.1609/aaai.v34i02.5543.
- 4 Yahav Bechavod, Katrina Ligett, Zhiwei Steven Wu, and Juba Ziani. Causal feature discovery through strategic modification. *ArXiv*, abs/2002.07024, 2020. URL: <https://arxiv.org/abs/2002.07024>.
- 5 Mark Braverman and Sumegha Garg. The role of randomness and noise in strategic classification. In Aaron Roth, editor, *1st Symposium on Foundations of Responsible Computing, FORC 2020, June 1-3, 2020, Harvard University, Cambridge, MA, USA (virtual conference)*, volume 156 of *LIPICs*, pages 9:1–9:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.FORC.2020.9.
- 6 Michael Brückner and Tobias Scheffer. Stackelberg games for adversarial prediction problems. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '11*, pages 547–555, New York, NY, USA, 2011. Association for Computing Machinery. doi:10.1145/2020408.2020495.
- 7 Sam Corbett-Davies and Sharad Goel. The measure and mismeasure of fairness: A critical review of fair machine learning. *CoRR*, abs/1808.00023, 2018. arXiv:1808.00023.
- 8 Jinshuo Dong, Aaron Roth, Zachary Schutzman, Bo Waggoner, and Zhiwei Steven Wu. Strategic classification from revealed preferences. In *Proceedings of the 2018 ACM Conference on Economics and Computation, EC '18*, pages 55–70, New York, NY, USA, 2018. Association for Computing Machinery. doi:10.1145/3219166.3219193.
- 9 Yiding Feng, Jason D. Hartline, and Yingkai Li. Simple mechanisms for non-linear agents. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 3802–3816. SIAM, 2023. doi:10.1137/1.9781611977554.ch148.
- 10 Alex M. Frankel and Navin Kartik. Improving information from manipulable data. *arXiv: Theoretical Economics*, June 2019. doi:10.1093/jeea/jvab017.
- 11 Jason Gaitonde, Yingkai Li, Bar Light, Brendan Lucier, and Aleksandrs Slivkins. Budget pacing in repeated auctions: Regret and efficiency without convergence. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA*, volume 251 of *LIPICs*, pages 52:1–52:1. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICs.ITCS.2023.52.
- 12 Nika Haghtalab, Nicole Immorlica, Brendan Lucier, and Jack Z. Wang. Maximizing welfare with incentive-aware evaluation mechanisms. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 160–166. International Joint Conferences on Artificial Intelligence Organization, July 2020. Main track. doi:10.24963/ijcai.2020/23.
- 13 Moritz Hardt, Nimrod Megiddo, Christos Papadimitriou, and Mary Wootters. Strategic classification. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, ITCS '16*, pages 111–122, New York, NY, USA, 2016. Association for Computing Machinery. doi:10.1145/2840728.2840730.

- 14 Keegan Harris, Hoda Heidari, and Zhiwei Steven Wu. Stateful strategic regression. In Marc’Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 28728–28741, 2021. URL: <https://proceedings.neurips.cc/paper/2021/hash/f1404c2624fa7f2507ba04fd9dfc5fb1-Abstract.html>.
- 15 Lily Hu, Nicole Immorlica, and Jennifer Wortman Vaughan. The disparate effects of strategic manipulation. In danah boyd and Jamie H. Morgenstern, editors, *Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* 2019, Atlanta, GA, USA, January 29-31, 2019*, pages 259–268. ACM, 2019. doi:10.1145/3287560.3287597.
- 16 Christopher Jung, Sampath Kannan, Changhwa Lee, Mallesh M. Pai, Aaron Roth, and Rakesh Vohra. Fair prediction with endogenous behavior. In Péter Biró, Jason D. Hartline, Michael Ostrovsky, and Ariel D. Procaccia, editors, *EC ’20: The 21st ACM Conference on Economics and Computation, Virtual Event, Hungary, July 13-17, 2020*, pages 677–678. ACM, 2020. doi:10.1145/3391403.3399473.
- 17 Jon Kleinberg, Jens Ludwig, Sendhil Mullainathan, and Ashesh Rambachan. Algorithmic fairness. *AEA Papers and Proceedings*, 108:22–27, May 2018. doi:10.1257/pandp.20181018.
- 18 Jon Kleinberg and Manish Raghavan. How do classifiers induce agents to invest effort strategically? In *Proceedings of the 2019 ACM Conference on Economics and Computation, EC ’19*, pages 825–844, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3328526.3329584.
- 19 Jon M. Kleinberg. Inherent trade-offs in algorithmic fairness. In Konstantinos Psounis, Aditya Akella, and Adam Wierman, editors, *Abstracts of the 2018 ACM International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS 2018, Irvine, CA, USA, June 18-22, 2018*, page 40. ACM, 2018. doi:10.1145/3219617.3219634.
- 20 Jean-Jacques Laffont and Jacques Robert. Optimal auction with financially constrained buyers. *Economics Letters*, 52(2):181–186, 1996. doi:10.1016/S0165-1765(96)00849-X.
- 21 Eric S. Maskin. Auctions, development, and privatization: Efficient auctions with liquidity-constrained buyers. *European Economic Review*, 44(4):667–681, 2000. doi:10.1016/S0014-2921(00)00057-X.
- 22 John Miller, Smitha Milli, and Moritz Hardt. Strategic classification is causal modeling in disguise. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 6917–6926. PMLR, 2020. URL: <http://proceedings.mlr.press/v119/miller20b.html>.
- 23 Smitha Milli, John Miller, Anca D. Dragan, and Moritz Hardt. The social cost of strategic classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* ’19*, pages 230–239, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3287560.3287576.
- 24 Mallesh M. Pai and Rakesh Vohra. Optimal auctions with financially constrained buyers. *J. Econ. Theory*, 150:383–425, 2014. doi:10.1016/j.jet.2013.09.015.
- 25 Ashesh Rambachan, Jon Kleinberg, Jens Ludwig, and Sendhil Mullainathan. An economic perspective on algorithmic fairness. *AEA Papers and Proceedings*, 110:91–95, May 2020. doi:10.1257/pandp.20201036.
- 26 Joseph Stiglitz and Andrew Weiss. Credit rationing in markets with imperfect information. *American Economic Review*, 71(3):393–410, 1981. URL: <https://EconPapers.repec.org/RePEc:aea:aecrev:v:71:y:1981:i:3:p:393-410>.
- 27 Joseph E Stiglitz. The Theory of “Screening,” Education, and the Distribution of Income. *American Economic Review*, 65(3):283–300, June 1975. URL: <https://ideas.repec.org/a/aea/aecrev/v65y1975i3p283-300.html>.
- 28 Shenke Xiao, Ziheng Wang, Mengjing Chen, Pingzhong Tang, and Xiwang Yang. Optimal common contract with heterogeneous agents. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7309–7316, April 2020. doi:10.1609/aaai.v34i05.6224.

A Deferred Proofs of Propositions and Lemmas

A.1 Proof that a Threshold Mechanism is Optimal for Single-budget

► **Proposition 3.** *Assume that an agent has constant budget \bar{b} on effort, i.e., the distribution B is a singular point mass. The threshold mechanism $y^{q'}$ with $q' = \tau \cdot \bar{b}$ is optimal.*

Proof. The optimality of $y^{q'}$ in fact follows from the stronger statement in Lemma 27 (below). ◀

The offline optimal mechanism (Definition 15) is generally unachievable. Despite that caveat, it is possible to achieve the offline optimal mechanism for the special case of singular budgets, as subsequently stated in Lemma 27. Recall the intuition given in the main body of the paper: “Proposition 3 holds because single-budget is a simple setting in which quality-thresholds directly implement skill-thresholds, in particular for the principal’s threshold τ .”

► **Lemma 27.** *Assume that an agent has constant budget \bar{b} on effort, i.e., the distribution B is a singleton point mass. Without directly observing the realization of the agent’s skill $\hat{s} \sim S$, the threshold mechanism $y^{q'}$ with $q' = \tau \cdot \bar{b}$ is offline optimal.*

Proof. We will show that $y^{q'}$ is offline optimal by showing that it gives allocation 1 to every (randomized) agent skill-type which gives positive utility to the principal, and gives allocation 0 to every agent skill-type which gives negative utility to the principal, thus pointwise-maximizing the principal’s utility function.

For agent $A = (s, \bar{b})$, the minimum effort required to reach threshold q' is $e' = q'/s = (\tau/s)\bar{b}$. Then $e' \leq \bar{b}$ is affordable (and rational) for A if and only if $\tau/s \leq 1$, an inequality which itself is true if and only if the principal’s utility $s - \tau \geq 0$ (from admitting A ; see page 5). By setting the quality threshold to be the maximum achievable by the skill level τ (which corresponds to 0-utility for skill-type τ), the mechanism allocates to exactly the upward closed set of all agent types from which it receives positive utility (Fact 2), and no others. ◀

A.2 Example of Insufficiency of Deterministic Mechanisms

The following Example 28 provides the proof-by-counterexample for Proposition 4.

► **Example 28.** Admission game admission game $\mathcal{G} = (S, B, \tau)$ is defined as follows.

Agent A has discrete skill space with two types (i.e., $|S| = 2$) with low skill $s_L = 1 + \epsilon_L$ (for $\epsilon_L \rightarrow 0$) and high skill $s_H = 2$. Agent A has discrete budget space with two types ($|B| = 2$) with low budget $b_L = 1/2$ and high budget $b_H = 1$. The distributions S and B have positive mass on each element of their respective supports but otherwise we leave them indeterminate. The principal P ’s skill threshold to measure utility is $\tau = 3/2$.

The following analysis will show that for the setting of Example 28, all deterministic threshold mechanisms are dominated by stochastic allocation $x = (1 - \epsilon_L/1 + \epsilon_L - \epsilon_x)$ for agents exhibiting quality at least 1.

As a starting point, consider the deterministic threshold mechanism $y^{q'}$ which picks $q' = 1$. The following gives initial analysis of an agent with high skill type s_H :

- minimum effort to achieve q' is: $e_H = 1/2$;
- utility from achieving threshold q' is: $1 - 1/2 = 1/2$;
- an agent of type (s_H, b_H) will put in effort to be admitted (given $q' = 1$ and furthermore, whenever $q' < 2$);

6:18 Screening with Disadvantaged Agents

- an agent of type (s_H, b_L) will also put in effort to be admitted, but critically, can not put in effort to be admitted if q' is increased above 1 by any $\epsilon_q > 0$ because this agent-type (s_H, b_L) is bounded by maximum quality $q^\dagger = s_H \cdot b_L = 2 \cdot 1/2 = 1$.

Alternatively, the following gives initial analysis of an agent with low skill type s_L :

- minimum effort to achieve q' is: $e_L = 1/1 + \epsilon_L$;
- utility from achieving threshold q' is: $1 - 1/1 + \epsilon_L = \epsilon_L/1 + \epsilon_L$;
- an agent of type (s_L, b_H) will put in effort to be admitted (given $q' = 1$);
- an agent of type (s_L, b_L) will put in 0 effort (because maximum quality is less than q').

Offline-optimal (Definition 15) allocates all agents with skill (s_H, \cdot) and rejects all agents (s_L, \cdot) . The current quality threshold under consideration $q' = 1$ is the largest threshold that will admit types (s_H, b_L) . Let $\pi_{a,T}$ be the probability corresponding to arbitrary agent type-attribute $a \in \{s, b\}$ and tier $T \in \{L, H\}$. The performance of every threshold mechanism fails to approach the performance of offline optimal (we write “ \gg ” to indicate that the gap is bounded away from 0):

- thresholds $q'_+ > q' = 1$ will not admit types (s_H, b_L) and thus will additively underperform offline optimal by at least:

$$\pi_{s,H} \cdot \pi_{b,L} \cdot (s_H - \tau) = \pi_{s,H} \cdot \pi_{b,L} \cdot (1/2) \gg 0$$

- thresholds $q'_- \leq q' = 1$ will admit types (s_L, b_H) and thus will additively underperform offline optimal by at least:

$$\pi_{s,L} \cdot \pi_{b,H} \cdot (\tau - s_L) = \pi_{s,L} \cdot \pi_{b,H} \cdot (1/2 - \epsilon_L) \gg 0$$

However, if we maintain $q' = 1$ and rather *decrease the probability of allocation* from 1 to $(1 - \epsilon_L/1 + \epsilon_L - \epsilon_x)$ for $\epsilon_x \rightarrow 0$, then all high types still strictly put in effort and will be admitted (with near-certainty), but the low types now strictly prefer to put in 0 effort.

Formally, for (single-menu-option) allocation $x = y(1) = (1 - \epsilon_L/1 + \epsilon_L - \epsilon_x)$, the utility calculations are (assuming minimum effort to be admitted, ignoring affordability due to budget):

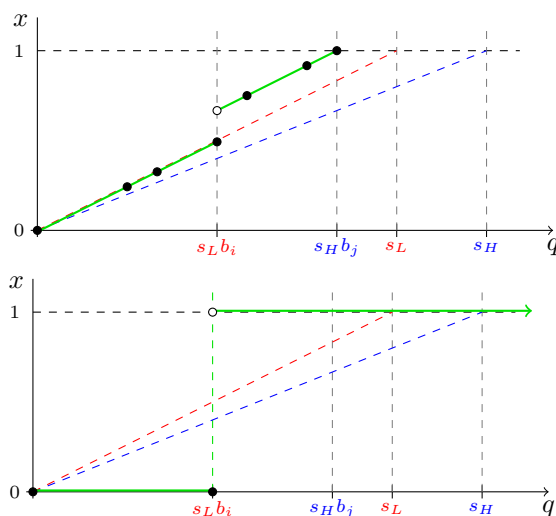
- agents with high skill type s_H have utility: $u_{(H,\cdot)} = y(1) - e_H = 1/2 - \epsilon_L/1 + \epsilon_L - \epsilon_x > 0$;
- agents with low skill type s_L have utility: $u_{(L,\cdot)} = y(1) - e_L = -\epsilon_x < 0$.

Considering, $\epsilon_L \rightarrow 0$ and $\epsilon_x \rightarrow 0$, the admission-rate of high-skill agents approaches 1 and thus the expected performance of this mechanism becomes arbitrarily close to the performance of offline optimal. Therefore, it strictly improves on the best of any deterministic threshold mechanism (which can't approach performance of offline optimal by the analysis above).

This completes the counterexample to illustrate that deterministic mechanisms are not sufficient.

A.3 A Comparison of Slanted-Stair Allocation to Deterministic Threshold

Section 3.1 gives discussion of the optimal characterization of mechanisms in Theorem 13. For purposes of space, we complete here the discussion of alignment between optimal mechanisms and *fairness*. To summarize the initial discussion in the main body, this alignment first is observed intuitively from the structure of the principal's utility from admitting an agent A , which is $u_P(A \mid \text{admitted}) = s - \tau$. Second, the offline optimal allocation is the “perfect” mechanism performance and is also arguably an ideal allocation in terms of fairness. We now give an additional intuitive perspective supporting this alignment.



■ **Figure 3** (Top) Illustration of a one-jump, slanted-stair allocation curve y^* (solid green), which is assumed to be optimal for its game parameters (for analysis purposes). The black dots are an example of discrete menu options. The single jump occurs at $q^{\text{jump}} = s_L b_i$. Regarding discussion in Appendix A.3: the first, left-most region is “below the jump;” the second, middle region is “above the jump but not fully allocated;” and the third, right-most region is “full allocation.” (Bottom) The (strictly-greater-than) threshold mechanism with threshold set equal to the jump-point in (Top), i.e., with $q' = q^{\text{jump}} = s_L b_i$.

Third – analyzing qualitatively for both mechanism performance and fairness – we can make a comparison between (a) an optimal single-jump-at- q^{jump} , slanted-stair allocation rule y^* of Theorem 13; and (b) the specific – albeit modified – threshold mechanism that jumps from allocation 0 to 1 at the same quality q^{jump} . For convenience, we copy Figure 2(Top) into Figure 3.

The modification is that the threshold mechanism in this section will require for admission that an agent’s exhibited quality be *strictly greater* than the threshold. This organizes the closed-versus-open endpoints of the threshold-step in a way that allows for a more-direct comparison to slanted-stair functions. This is illustrated in Figure 3(Bottom).

Graphically, the optimal mechanism y^* (which may be a discrete menu, corresponding to our discrete setting) will qualitatively have the single-jump structure of Figure 3(Top). Using agent skill/budget-indexing of Figure 3 (i.e., notation), the general structure of y^* has three regions:

1. the left-most region is “below the jump” defined by qualities $q \in [0, q^{\text{jump}} = s_L b_j]$;
2. the middle region is “above the jump but not full allocation” defined by qualities $q \in (q^{\text{jump}} = s_L b_j, s_H b_L)$;
3. the right-most region is “full allocation” defined by the quality $q = s_H b_L$ (and all larger qualities, though rational agents never choose these larger levels, which require exerting superfluous effort to achieve, without an increase in allocation).

The allocation rule y^* is optimal for the standard principal-objective, so it obviously dominates the threshold mechanism with its quality-space threshold set to be $q' = q^{\text{jump}} = s_L b_j$. In the following discussion, agents are considered to be “in” the region which contains their optimally-chosen quality for the given mechanism (subject to tie-breaking). We qualitatively analyze the same comparison for fairness:

1. in the left-most region, low-skill agents receive 0-allocation according to both y^* and the threshold mechanism; by contrast, high-skill agents receive 0-allocation according to the threshold mechanism, but positive allocation according to y^* (the solid green line in Figure 3(Top)); we suggest in this first region – regardless of the choice of fairness measure – that the fairness of y^* dominates the fairness of the threshold mechanism;
2. in the middle region, *all* low-skill agents receive allocation $y^*(q)$ for $q \rightarrow (q^{\text{jump}})^+$ (from above) according to y^* (by tie-breaking), which for *all* low-skill agents increases to full-allocation of 1 according to the threshold mechanism; whereas each high-skill agent (s_H, b_j) is exhibiting its respective maximum quality $q_{H,j}^\dagger = s_H \cdot b_j$ and receives allocation $y^*(q_{H,j}^\dagger)$ according to y^* which increases to full-allocation of 1 according to the threshold mechanism;
in this second region, we can not make a dominance argument because it partially depends on the unknown densities of agent-types represented in this region and it also depends on the technical measure of fairness; however, ignoring expectation and proportional density and instead simply comparing agents one-to-one, we do observe that low-skill types receive the larger benefit (increase in allocation) if we start with y^* as our default mechanism and consider changing to the threshold mechanism; furthermore, the threshold mechanism abolishes the (properly oriented) cardinal difference between low-skill and high-skill agents by instead awarding them an “arguably unfair” constant allocation (of 1);
3. in the right-most region, all skill-types in all mechanisms receive the same allocation of 1; thus in this third region, the mechanism y^* and the threshold mechanism are equally fair (or equally unfair).

Intuitively, the preceding comparison between the optimal mechanism y^* and the threshold mechanism – which specifically have jumps at the same quality-index q^{jump} – suggests that (single-jump) slanted-stair mechanisms are indeed more fair. In fact, we have already stated a strict dominance relationship for an obvious, special-case choice of the technical fairness measure.

► **Corollary 17.** *Given a correlated admission game $\mathcal{H} = (S = \Delta(\{s_L, s_H\}), \mathcal{B}^S = \{B_{s_L}, B_{s_H}\}, \tau)$ with $0 < s_L < \tau < s_H$ and $0 < b_1 < b_2 < \dots < b_n$. For the fair mechanism design problem which maximizes the fairness measure set equal to the principal’s utility function, the optimal mechanism and characterization of optimal mechanisms are determined identically to Theorem 13.*

Recall, the principal is naturally aligned with fairness. Then if we assign the fairness measure to be equal to the utility function of the principal, the analysis of the optimal mechanism for fairness gives the identical result as Theorem 13.

Fair Grading Algorithms for Randomized Exams

Jiale Chen ✉

Department of Management Science and Engineering, Stanford University, CA, USA

Jason Hartline ✉

Department of Computer Science, Northwestern University, Evanston, IL, USA

Onno Zoeter ✉

Booking.com, Amsterdam, The Netherlands

Abstract

This paper studies grading algorithms for randomized exams. In a randomized exam, each student is asked a small number of random questions from a large question bank. The predominant grading rule is simple averaging, i.e., calculating grades by averaging scores on the questions each student is asked, which is fair ex-ante, over the randomized questions, but not fair ex-post, on the realized questions. The fair grading problem is to estimate the average grade of each student on the full question bank. The maximum-likelihood estimator for the Bradley-Terry-Luce model on the bipartite student-question graph is shown to be consistent with high probability when the number of questions asked to each student is at least the cubed-logarithm of the number of students. In an empirical study on exam data and in simulations, our algorithm based on the maximum-likelihood estimator significantly outperforms simple averaging in prediction accuracy and ex-post fairness even with a small class and exam size.

2012 ACM Subject Classification Social and professional topics → Student assessment

Keywords and phrases Ex-ante and Ex-post Fairness, Item Response Theory, Algorithmic Fairness in Education

Digital Object Identifier 10.4230/LIPIcs.FORC.2023.7

Related Version *Full Version*: <https://arxiv.org/abs/2304.06254>

Funding *Jason Hartline*: The author was supported in part by NSF CCF 1733860.

1 Introduction

A common approach for deterring cheating in online examinations is to assign students random questions from a large question bank. This random assignment of questions with heterogeneous difficulties leads to different overall difficulties of the exam that each student faces. Unfortunately, the predominant grading rule – simple averaging – averages all question scores equally and results in an unfair grading of the students. This paper develops a grading algorithm that utilizes structural information of the exam results to infer student abilities and question difficulties. From these abilities and difficulties, fairer and more accurate grades can be estimated. This grading algorithm can also be used in the design of short exams that maintain a desired level of accuracy.

During the COVID-19 pandemic, learning management systems (LMS) like Blackboard, Moodle, Canvas by Instructure, and D2L have benefited worldwide students and teachers in remote learning [20]. The current exam module in these systems includes four steps. In the first step, the instructor provides a large question bank. In the second step, the system assigns each student an independent random subset of the questions. (Assigning each student an independent random subset of the questions helps mitigate cheating.) In the third step, students answer the questions. In the last step, the system grades each student proportionally to her accuracy on assigned questions, i.e., by *simple averaging*.



© Jiale Chen, Jason Hartline, and Onno Zoeter;

licensed under Creative Commons License CC-BY 4.0

4th Symposium on Foundations of Responsible Computing (FORC 2023).

Editor: Kunal Talwar; Article No. 7; pp. 7:1–7:22

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

While randomizing questions and grading with simple averaging is ex-ante fair, it is not generally ex-post fair. When questions in the question bank have varying difficulties, then by random chance a student could be assigned more easy questions than average or more hard questions than average. Ex-post in the random assignment of questions to students, the simple averaging of scores on each question allows variation in question difficulties to manifest as ex-post unfairness in the final grades.

The aim of this paper is to understand grading algorithms that are fair and accurate. Given a bank of possible questions, a benchmark for both fairness and accuracy is the counterfactual grade that a student would get if the student was asked all of the questions in the question bank. Exams that ask fewer questions to the students may be inaccurate with respect to this benchmark and the inaccuracy may vary across students and this variation is unfair. This benchmark allows for both the comparison of grading algorithms and the design of randomized exams, i.e., the method for deciding which questions are asked to which students.

The grading algorithms developed in this paper are based on the Bradley-Terry-Luce model [6] on bipartite student-question graphs. This model is also studied in the psychology literature where it is known as the Rasch model [19]. This model views the student answering process as a noisy comparison between a parameter of the student and a parameter of the question. Specifically, there is a merit value vector u which describes the student abilities and question difficulties and is unknown to the instructor. The probability that student i answers question j correctly is defined to be

$$f(u_i - u_j) = \frac{\exp(u_i)}{\exp(u_i) + \exp(u_j)},$$

where $f(x) = \frac{1}{1 + \exp(-x)}$, and u_i, u_j represents the merit value of student i and question j respectively.

The paper develops a grading algorithm that is based on the maximum likelihood estimator \mathbf{u}^* of the merit vector. Compared to simple averaging which only focuses on student in-degrees and out-degrees, our grading algorithm incorporates more structural information about the exam result and, as we show, reduces ex-post unfairness.

Results

Our theoretical analysis considers a sequence of distributions over random question assignment graphs indexed by n and m by setting the number of students to n and number of questions in the question bank to $m \geq n$ and assigning $d_{n,m}$ random questions uniformly and independently to each student. The exam result can be represented by a directed graph, where an edge from a student to a question represents a correct answer and the opposite direction represents an incorrect answer. We prove that the maximum likelihood estimator exists and is unique within a strongly connected component (Theorem 10). Let $\alpha_{n,m} = \max_{1 \leq i, j \leq n+m} u_i - u_j$ be the largest difference between any pair of merits. We prove that if

$$\frac{\exp(\alpha_{n,m})(n+m) \log(n+m)}{nd_{n,m}} \rightarrow 0 \quad (n, m \rightarrow \infty),$$

then the probability that the exam result graph is strongly connected goes to 1 (Theorem 11). Thus, the existence and uniqueness of the MLE are guaranteed under the model. We also prove that if $\exp(2(\alpha_{n,m} + 1)) \Delta_{n,m} \rightarrow 0$ ($n, m \rightarrow \infty$), where $\Delta_{n,m} = \sqrt{\frac{m \log^3(n+m)}{nd_{n,m} \log^2(\frac{n}{m} d_{n,m})}}$, then the MLEs are uniformly consistent, i.e., $\|\mathbf{u}^* - \mathbf{u}\|_\infty \xrightarrow{\mathbb{P}} 0$ (Theorem 13). These

theoretical results complement the empirical and simulation results from the literature on the Rasch model with random missing data. Our analysis is similar to Han et al.'s [15] which studies Erdős-Rényi random graphs.

Our empirical analysis considers a study of grading algorithms on both anonymous exam data and numerical simulations. The exam data set consists of 22 questions and 35 students with all students answering all questions. From this data set, randomized exams with fewer than 22 questions can be empirically studied and grading algorithms can be compared. Our algorithm outperforms simple averaging when students are asked at least seven questions. We fit the model parameters to this real-world dataset and run numerical simulations with the resulting generative model. With these simulations, we compare our algorithm and simple averaging on ex-post bias and ex-post error, two notions of ex-post unfairness. For example, when each of the 35 students answers a random 10 of the 22 questions, we find that the expected maximum ex-post bias of simple averaging is about 100 times higher than that of our algorithm. The expected output of simple averaging has about 13% expected deviation from the benchmark for the most unlucky student, which would probably lead to a different letter grade for the students, while the deviation is only about 1.6% for our algorithm. In the same setting, we found that our algorithm achieves a factor of 8 percent smaller ex-post error, which is a noisier concept of ex-post unfairness. After the decomposition of ex-post error into ex-post bias and variance, we found that our algorithm achieves a significantly smaller ex-post bias with the cost of a slightly larger variance of the output, and in combination it reduces the ex-post error.

Related Work

The literature on peer grading also compares estimation from structural models and simple averaging. When peers are assigned to grade submissions, the quality of peer reviews can vary. Structural models can be used to estimate peer quality and calculate grades on the submissions that put higher weight on peers who give higher-quality reviews. Alternatively, submission grades can be calculated by simply averaging the reviews of each peer. The literature has mixed results. De Alfaro and Shavlovsky [7] propose an algorithm based on the reputation that largely outperforms simple averaging on synthetic data, and is better on real-world data when student grading error is not random. Reily et al. [21] and Hamer et al. [14] also point out that sophisticated aggregation improves the accuracy compared to simple averaging and also helps to avoid rogue strategies including laziness and aggressive grading. On the other hand, Sajjadi et al. [23] show that statistical and machine learning methods do not perform better than simple averaging on their dataset. In contrast, our result that structural models outperform simple averaging is replicated on several data sets. We believe this difference with the peer grading literature is due to differences in the degrees of the bipartite graphs considered. The exam grading graphs are of a higher degree than the peer-grading graphs.

In psychometrics, item response theory (IRT) considers mathematical models that build relationships between unobserved characteristics of respondents and items and observed outcomes of the responses. The Rasch model is a commonly used model of IRT that can be applied to psychometrics, educational research [19], health sciences [5], agriculture [18], and market research [4]. Previous simulation studies showed that among different item parameter estimation methods for the Rasch model, the joint maximum likelihood (JML) method, and its variants provides one of the most efficient estimates [22], especially with missing data [25, 8]. In our setting, random assignment of questions to students can be seen as a special case of missing data. With complete data, the condition for the consistency of the

maximum likelihood estimators is analyzed [12, 13]. With missing data, though plenty of work on simulation exists, there is a lack of theoretical work that proves mathematically the consistency of the maximum likelihood estimators.

The Rasch model can be regarded as a special case of the Bradley-Terry-Luce (BTL) model [6] for the pairwise comparison of respondents with items by restricting the comparison graph to a bipartite graph. For the BTL model with Erdős-Rényi graph $G(n, p_n)$, the maximum likelihood estimator (MLE) can be solved by an efficient algorithm [27, 9, 16], and is proved to be a consistent method in l_∞ norm when $\liminf_{n \rightarrow \infty} p_n > 0$ [24, 26], and recently when $p_n \geq \frac{\log n^3}{n}$ [15] which is close to the theoretical lower bound of $\frac{\log n}{n}$, below which the comparison graph would be disconnected with positive probability and there is no unique MLE.

In this paper, we follow the method of Han et al.'s [15] to prove the consistency of the Rasch model with missing data, or BTL model with a sparse bipartite graph, when each vertex in the left part is assigned small number of random edges to the vertices in the right part. We also propose an extension of the algorithm that reasonably deals with the cases where the MLE does not exist.

Fowler et al. [10] recently studied unfairness detection of the simple averaging under the same randomized exam setting and argue that “the exams are reasonably fair”. They use certain IRT model to fit exams based on their real-world data, and find that the simple averaging gives grades that are strongly correlated with the students’ inferred abilities. They also simulate under the IRT model, over random assignment and the student answering process. The simulation shows that, if given any fixed assignment we consider the absolute error of the students’ expected performance over their answering process, the average absolute error over different assignments reaches a 5-percentage bias. We find similar results in our simulation, and design a method to reduce the corresponding error by a factor of ten. Our method solves one of their future directions by adjusting grades of the students based on their exam variant.

All large-scale standardized tests including the Scholastic Aptitude Test (SAT) and Graduate Record Examination (GRE) are using item response theory (IRT) to generate score scales for alternative forms [1]. This test equating process can be divided into two steps, linking and equating. Linking refers to how to estimate the IRT parameters of students and questions under the model; and equating refers to how to adjust the raw grade of the students to adapt to different overall difficulty levels in different version of the exam (e.g. [17]). One of the most popular test equating processes is IRT true-score equating with nonequivalent-groups anchor test (NEAT) design. In the NEAT design, there are two test forms given to two population of students, where a set of common questions is contained in both forms. Linking performs by putting the estimated parameter of the common items onto the same scale through a linear transformation, since any linear transformation gives the same probability under the IRT model. Equating performs by taking the estimated ability of the student from the second form and compute the expected number of accurate answers in the first form as the adjusted grade. Since these large-scale standardized tests have a large population of students for each variant of the exam, the above test equating process works well. Our methods can be viewed as adapting the statistical framework of linking and equating to the administration of a single exam for a small population of students. In our randomized exam setting with small scale, however, every student receives a different form of the exam, thus it is almost impossible to estimate the parameters for every form separately or to decide an anchor set of question and do the same linking. Our algorithm uses the concurrent linking that estimates all parameters at the same time based on the information in all forms. As for equating, we use a similar method of true-score equating, but compute on the whole question bank instead of one specific form.

In the problem of fair allocation of indivisible items, Best-of-Both-Worlds (BoBW) fairness mechanisms (e.g., [2, 11, 3]) try to provide both ex-ante fairness and ex-post fairness to agents. An ex-ante fair mechanism is easy to be found. For example, giving all items to one random agent guarantees that every agent receives a $\frac{1}{n}$ fraction of the total value in expectation (ex-ante proportionality). However, such a mechanism is clearly not ex-post fair. Likewise, simple averaging gives every student an unbiased grade ex-ante, but neglects the different overall difficulty among students ex-post. We propose another grading rule that evaluates the difficulties of the questions and adjusts the grades according to them, which achieves better ex-post fairness of the students.

2 Model

Consider a set of students S and a bank of questions Q . A merit vector \mathbf{u} is used to describe the key property of the students and questions. Specifically, for any student $i \in S$, u_i represents the ability of the student; for any question $j \in Q$, u_j represents the difficulty of the question. We put them in the same vector for convenience. The merit vector is unknown when the exam is designed. Denote w_{ij} as the outcome of the answering process. Then w_{ij} s are independent Bernoulli random variables, where $w_{ij} = 1$ represents a correct answer, $w_{ij} = 0$ represents an incorrect answer, and

$$\Pr[w_{ij} = 1] = 1 - \Pr[w_{ij} = 0] = \frac{\exp(u_i)}{\exp(u_i) + \exp(u_j)} = f(u_i - u_j),$$

where $f(x) = \frac{1}{1 + \exp(-x)}$. The goal of the exam design is to assign a small number of questions to each student (task assignment graph), and based on the exam result (exam result graph), give each student a grade (grading rule) that accurately estimates her performance over the whole question bank (benchmark). We give a formal description of the task assignment graph, exam result graph, benchmark, and grading rule below.

► **Definition 1 (Task Assignment Graph).** *The task assignment graph $G = (S \cup Q, E)$ is an undirected bipartite graph, where the left part of the vertices represents the students and the right part represents the questions, and an edge between $i \in S$ and $j \in Q$ exists if and only if the instructor decides to assign question j to student i .*

► **Definition 2 (Exam Result Graph).** *The exam result graph $G' = (S \cup Q, E')$ is a directed bipartite graph constructed from the task assignment graph G . All directed edges are between students and questions. For any edge $(i, j) \in G$ in the task assignment graph, where $i \in S$ and $j \in Q$, if student i answers question j correctly in the exam, i.e., we observe that $w_{ij} = 1$, there is an edge $i \rightarrow j$ in G' ; if the answer is incorrect, i.e., we observe that $w_{ij} = 0$, there is an edge $j \rightarrow i$ in G' . For other student-question pairs that do not occur in the task assignment graph G , there is also no edge between them in the exam result graph G' .*

To evaluate different exam designs and grading rules, we propose the following benchmark.

► **Definition 3 (Benchmark).** *In an ideal case where we know the distribution over the outcome of the answering processes w_{ij} s, the instructor would measure the students' performance by their expected accuracy on a uniformly random question in the bank. Formally, the benchmark for any student i 's grade is*

$$\text{opt}_i = \mathbb{E}_{j \sim \mathcal{U}(Q)}[w_{ij}] = \frac{1}{|Q|} \sum_{j \in Q} f(u_i - u_j). \quad (1)$$

7:6 Fair Grading Algorithms for Randomized Exams

The benchmark is an ideal way to grade the student if the instructor has complete information on all answering processes. On the other hand, when the instructor only observes one sample of each $w_{i,j}$ involved in the exam, we will use a grading rule to grade the students.

► **Definition 4 (Grading Rule).** *In an exam, the instructor gives a grade for each student based on the exam result graph. A grading rule is a mapping $g: G' \rightarrow \mathbb{R}^S$ from the exam result graph to the grades for each student.*

One interpretation of the grade is as an estimation of the benchmark, i.e., students' expected accuracy on a uniformly random question in the bank, which combines the two important criteria of fairness and accuracy. To evaluate the exam design, we compare the performance of the grading rule to the benchmark and aggregate the error among all students. Specifically, there are three stages of the exam design, before the randomization of the task assignment graph, after the randomization of the task assignment graph and before the student answering process, and after the student answering process. In each stage, we might care about the maximum or average unfairness among students.

► **Definition 5 (Ex-ante Bias).** *For a given algorithm alg , the ex-ante bias for student i is defined as the mean square error of the algorithm's expected performance compared to the benchmark, over a random family \mathcal{G} of task assignment graphs, i.e., $(\mathbb{E}_{G \sim \mathcal{G}} \mathbb{E}_w[\text{alg}_i] - \text{opt}_i)^2$.*

► **Definition 6 (Ex-post Bias).** *For a given algorithm alg and a fixed task assignment graph G , the ex-post bias for student i is defined as the mean square error of the algorithm's expected performance compared to the benchmark on G , i.e., $(\mathbb{E}_w[\text{alg}_i] - \text{opt}_i)^2$.*

► **Definition 7 (Ex-post Error).** *For a given algorithm alg , a fixed task assignment graph G , and a fixed realization of the student answering process w , the ex-post error for student i is defined as the mean square error of the algorithm's performance compared to the benchmark on G and w , i.e., $(\text{alg}_i - \text{opt}_i)^2$.*

By definition, ex-ante bias takes expectation over both random graphs and the noisy answering process, ex-post bias takes expectation over the noisy answering process, while ex-post error directly measures the error. Thus ex-post error is harder than ex-post bias which is harder than ex-ante bias to achieve.

► **Example 8 (Simple Averaging).** Simple averaging is a commonly used grading rule in exams. It calculates the average accuracy on the questions the student receives. Formally, given a exam result graph G' , the simple averaging grades student i by

$$\text{avg}_i = \frac{\text{deg}_i^+}{\text{deg}_i^- + \text{deg}_i^+} = \frac{\sum_j 1_{(i,j) \in E'}}{\sum_j 1_{(i,j) \in E}}, \quad (2)$$

where deg^+ and deg^- represents the outdegree and indegree of the vertex in G' , respectively.

► **Theorem 9.** *The simple averaging is ex-ante fair over any family of bipartite graphs \mathcal{G} that is symmetric with respect to the questions, i.e., its ex-ante bias is 0.*

Proof.

$$\begin{aligned} \forall i, \mathbb{E}_{G \sim \mathcal{G}} \mathbb{E}_w[\text{avg}_i] &= \mathbb{E}_{G \sim \mathcal{G}} \mathbb{E}_w \left[\frac{\sum_j 1_{(i,j) \in E'}}{\sum_j 1_{(i,j) \in E}} \right] = \mathbb{E}_{G \sim \mathcal{G}} \mathbb{E}_w \left[\frac{\sum_j w_{ij} 1_{(i,j) \in E}}{\sum_j 1_{(i,j) \in E}} \right] \\ &= \mathbb{E}_{G \sim \mathcal{G}} \left[\frac{\sum_j \mathbb{E}[w_{ij}] 1_{(i,j) \in E}}{\sum_j 1_{(i,j) \in E}} \right] = \sum_j \mathbb{E}[w_{ij}] \mathbb{E}_{G \sim \mathcal{G}} \left[\frac{1_{(i,j) \in E}}{\sum_j 1_{(i,j) \in E}} \right] = \text{opt}_i. \quad \blacktriangleleft \end{aligned}$$

In other words, simple averaging can be seen as an ex-ante unbiased estimator of the benchmark. However, ex-post, i.e., on one specific task assignment graph, simple averaging is unfair. Intuitively, some unlucky students might be assigned harder questions and receive a significantly lower average grade than the benchmark, and the opposite happens to some lucky students. We will visualize this phenomenon in Figure 2 in Section 5.3.1.

Based on the above definitions, we now formalize the procedure and goal of the exam grading problem.

- i. The instructor chooses a task assignment graph G .
- ii. The students receive questions according to G and give their answer sheet back, thus the instructor receives the exam result graph G' .
- iii. The instructor uses a grading rule g to grade the students based on G' .
- iv. The grade $g(G')$ should have a small maximum (average) ex-post bias or ex-post error.

3 Method

In this section, we propose our method for the exam grading problem. According to our formalization of the problem, any method contains two parts: generating the task assignment graph G , and choosing the grading rule g . We describe each of them respectively.

3.1 Task Assignment Graph

To generate the task assignment graph, we independently assign each student d different questions u.a.r. from the question bank.

3.2 Grading Rule

Recall that a grading rule maps from an exam result graph G' to a vector of probabilities. In contrast with simple averaging which only considers the local information (the in-degrees and out-degrees of the students), we use structural information of the exam result graph for analysis. Our grading rule is an aggregation of a prediction matrix $h \in [0, 1]^{S \times Q}$, where h_{ij} represents the algorithm's prediction on the probability that student i answers correctly question j . The grade for student i will be the average of h_{ij} s over all $j \in Q$, i.e. $\text{alg}_i = \frac{1}{|Q|} \sum_{j \in Q} h_{ij}$. We use $u \rightsquigarrow v$ to represent the existence of a directed path in G' that starts with u and ends with v , and $u \not\rightsquigarrow v$ for nonexistence. The algorithm classifies the elements h_{ij} s into four cases: existing edge $(i, j) \in E$, same component $i \rightsquigarrow j \wedge j \rightsquigarrow i$, comparable components $i \rightsquigarrow j \oplus j \rightsquigarrow i$, and incomparable components $i \not\rightsquigarrow j \wedge j \not\rightsquigarrow i$.

Existing Edge

For $(i, j) \in E$, we observe w_{ij} from the exam result graph G' , hence $h_{ij} = w_{ij}$.

Same Component

For student $i \in S$ and question $j \in Q$ satisfy $i \rightsquigarrow j \wedge j \rightsquigarrow i$, they are in the same strongly connected component in G' . We make all predictions in the component simultaneously, by inferring the student abilities and question difficulties from the structure of the component. Formally, denote V' as the vertex set of the component. From Theorem 10, the strong connectivity guarantees the existence of the maximum likelihood estimators (MLEs) $\mathbf{u}^* \in \mathbb{R}^{V'}$. We can use a minorization–maximization algorithm from [16] to calculate the MLEs and set $h_{ij} = f(u_i^* - u_j^*)$ for any missing edge (i, j) between students and questions inside this component.

Comparable Components

W.l.o.g., we assume $i \rightsquigarrow j$ and $j \not\rightsquigarrow i$, thus every directed path between those two vertices starts with the student and ends with the question, showing strong evidence of a correct answer. In other words, considering the strongly connected components they belong to, the component that contains the student has a “higher level” in the condensation graph of G' and can reach the component that contains the question, i.e., they belong to comparable components in the condensation graph. In this case, we set $h_{ij} = 1$. Similarly, if $j \rightsquigarrow i$ and $i \not\rightsquigarrow j$, we set $h_{ij} = 0$

Incomparable Components

For a student i and question j that satisfy $i \not\rightsquigarrow j \wedge j \not\rightsquigarrow i$, i.e., in incomparable components, we use the average of the predictions in the above three cases as the prediction for h_{ij} .

4 Theory

In this section, we show several properties of our algorithm. Due to the limited space, we will defer most detailed proofs to Appendix A. Recall that the Bradley-Terry-Luce model describes the outcome of pairwise comparisons as follows. In a comparison between subject i and subject j , subject i beats subject j with probability

$$p_{ij} = \frac{\exp(u_i)}{\exp(u_i) + \exp(u_j)} = f(u_i - u_j),$$

where $\mathbf{u} = (u_1 \dots, u_{n+m})$ represents the merit parameters of $n + m$ subjects and $f(x) = \frac{1}{1 + \exp(-x)}$. We consider the Bradley-Terry-Luce model under a family of random bipartite task assignment graphs $\mathcal{B}(n, m, d_{n,m})$. Specifically, a task assignment graph $G(L \cup R, E)$ with n vertices in L and m vertices in R , where $n \leq m$, is constructed by linking $d_{n,m}$ different random vertices in R to each left vertex in L , i.e., L is regular but R is not.

Given a task assignment graph G , denote A as its adjacency matrix. For any two subjects i and j , the number of comparisons between them follows $A_{ij} \in \{0, 1\}$. We define A'_{ij} as the number of times that subject i beats subject j , thus $A'_{ij} + A'_{ji} = A_{ij} = A_{ji}$. In other words, A' is the adjacency matrix of the exam result graph G' . Based on the observation of G' , the log-likelihood function is

$$\mathcal{L}(\mathbf{u}) = \sum_{1 \leq i \neq j \leq n+m} A'_{ij} \log p_{ij} = \sum_{1 \leq i \neq j \leq n+m} A'_{ij} \log f(u_i - u_j). \quad (3)$$

Denote $\mathbf{u}^* = (u_1^*, u_1^*, \dots, u_{n+m}^*)$ as the maximum likelihood estimators (MLEs) of \mathbf{u} . Since \mathcal{L} is additive invariant, w.l.o.g. we assume $u_1 = 0$ and set $u_1^* = 0$. Since $(\log f(x))' = 1 - f(x)$ the likelihood equation can be simplified to

$$\sum_{j=1}^{n+m} A'_{ij} = \sum_{j=1}^{n+m} A_{ij} f(u_i^* - u_j^*), \forall i. \quad (4)$$

4.1 Existence and Uniqueness of the MLEs

Zermelo [27] and Ford [9] gave a necessary and sufficient condition for the existence and uniqueness of the MLEs in (4).

Condition A

For every two nonempty sets that form a partition of the subjects, a subject in one set has beaten a subject in the other set at least once.

To provide an intuitive understanding of Condition A, we show its equivalence to the strong connectivity of the exam result graph G' . Then we state our theorem on when Condition A holds.

► **Theorem 10.** *Condition A holds if and only if the exam result graph G' is strongly connected.*

Proof. Condition A says that for any partition (V_1, V_2) of the vertices $L \cup R$, there exists an edge from V_1 to V_2 and also an edge from V_2 to V_1 . If G' is strongly connected, Condition A directly holds by the definition of strong connectivity. Otherwise, if G' is not strongly connected, the condensation of G' contains at least two SCCs. We pick one strongly connected component with no indegree as V_1 and the remaining vertices as V_2 , then there is no edge from V_2 to V_1 , i.e., Condition A fails. ◀

► **Theorem 11** (Existence and Uniqueness of MLEs). *If*

$$\frac{\exp(\alpha_{n,m})(n+m)\log(n+m)}{nd_{n,m}} \rightarrow 0 \quad (n, m \rightarrow \infty), \tag{5}$$

where $\alpha_{n,m} = \max_{1 \leq i, j \leq n+m} u_i - u_j$ is the largest difference between all possible pairs of merits, then $\Pr[\text{Condition A is satisfied}] \rightarrow 1 \quad (n, m \rightarrow \infty)$.

To prove Theorem 11, we analyze the edge expansion property (Lemma 12) of the task assignment graph G and take a union bound on all valid subsets to bound the probability that G' fails Condition A.

► **Lemma 12** (Edge Expansion). *Under condition (5),*

$$\Pr \left[\forall S \subset V, \text{ s.t. } |S| \leq \frac{n+m}{2}, \quad \frac{|\partial S|}{|S|} > \frac{nd_{n,m}}{2(n+m)} \right] \rightarrow 1 \quad (n, m \rightarrow \infty),$$

where $\partial S = \{(u, v) \in E : u \in S, v \in V \setminus S\}$ for the task assignment graph $G(V, E)$.

4.2 Uniform Consistency of the MLEs

Based on condition (5), Theorem 11 shows the existence and uniqueness of the MLEs. In this part, we give an outline of the proof for the uniform consistency of the MLEs (Theorem 13).

► **Theorem 13** (Uniform Consistency of MLEs). *If*

$$\exp(2(\alpha_{n,m} + 1)) \Delta_{n,m} \rightarrow 0 \quad (n, m \rightarrow \infty), \tag{6}$$

where $\Delta_{n,m} = \sqrt{\frac{m \log^3(n+m)}{nd_{n,m} \log^2(\frac{n}{m} d_{n,m})}}$, then the MLEs are uniformly consistent, i.e., $\|\mathbf{u}^* - \mathbf{u}\|_\infty \xrightarrow{\mathbb{P}} 0$.

► **Corollary 14** (Rates). *In the case where $\alpha_{n,m} = O(1)$, and $d_{n,m} = \Omega\left(\frac{m \log^3(n+m)}{n}\right)$, with probability $1 - 2(n+m)^{-2}$, we have*

$$\|\mathbf{u}^* - \mathbf{u}\|_\infty = O\left(\frac{\log n}{\log(\frac{n}{m} d_{n,m})} \sqrt{\frac{m \log(n+m)}{nd_{n,m}}}\right).$$

Denote $\varepsilon_i = u_i^* - u_i$ as the estimation error of the maximum likelihood estimators. Since we assume $u_1 = 0$ and set $u_1^* = 0$, we have $\varepsilon_1 = u_1^* - u_1 = 0$. Consider the two subjects with the most negative estimation error and the most positive estimation error $\underline{i} = \arg \min_i \varepsilon_i \leq \varepsilon_1 = 0$, $\bar{i} = \arg \max_i \varepsilon_i \geq \varepsilon_1 = 0$, and their corresponding error $\underline{\varepsilon} = \min_i \varepsilon_i$, $\bar{\varepsilon} = \max_i \varepsilon_i$, then we have $\|\mathbf{u}^* - \mathbf{u}\|_\infty = \max\{-\underline{\varepsilon}, \bar{\varepsilon}\} \leq \bar{\varepsilon} - \underline{\varepsilon}$. The goal is to identify a specific number D , such that more than half ε_i s are at most $\underline{\varepsilon} + D$, and more than half ε_i s are at least $\bar{\varepsilon} - D$. Then at least one subject is on both sides, thus $\bar{\varepsilon} - \underline{\varepsilon}$ is bounded by $2D$.

To identify D , we check a sequence of increasing numbers $\{D_k\}_{k=0}^{K_{n,m}}$, and the two corresponding growing sets $\{\underline{B}_k\}_{k=0}^{K_{n,m}}$ and $\{\bar{B}_k\}_{k=0}^{K_{n,m}}$ that contains the subjects with estimation errors D_k -close to $\underline{\varepsilon}$ and $\bar{\varepsilon}$ respectively. Under careful choice of $K_{n,m}$ and $\{D_k\}_{k=0}^{K_{n,m}}$, we will show that $\underline{B}_{K_{n,m}}$ and $\bar{B}_{K_{n,m}}$ both contain more than half subjects.

The main difficulty is showing the growth of $\{\underline{B}_k\}_{k=0}^{K_{n,m}}$ and $\{\bar{B}_k\}_{k=0}^{K_{n,m}}$. We prove this by considering the local growth of the sets, i.e., $N(\underline{B}_k) \cap \underline{B}_{k+1}$ and $N(\bar{B}_k) \cap \bar{B}_{k+1}$. By symmetry, we only consider \underline{B}_k . Lemma 15 analyzes the generation of the random task assignment graphs and shows a vertex expansion property that describes the growth of the neighborhoods $N(\underline{B}_k)$. Lemma 16 starts with any vertex i in \underline{B}_k , analyzes the first order equations of the MLE to exclude the vertices that are in the neighborhoods $N(\{i\})$ and but are not in \underline{B}_{k+1} , and gives a lower bound on the size of $N(\{i\}) \cap \bar{B}_{k+1}$. Finally, we jointly consider all vertices in \underline{B}_k and provide a lower bound on the size of $N(\underline{B}_k) \cap \underline{B}_{k+1}$, which shows the growth rate of \underline{B}_k and finishes the proof.

Definition of Notations

- $K_{n,m} = 2 \left\lceil \frac{\log n}{\log(\frac{n}{m} d_{n,m})} - 1 \right\rceil$ is the number of steps of the growth.
- $c_{n,m} = \frac{\exp(-(\alpha_{n,m}+1))}{4}$ is a lower bound on $f'(x)$ for $|x| \leq \alpha_{n,m} + 1$.
- $q_{n,m} = \frac{c_{n,m} \log(\frac{n}{m} d_{n,m})}{5 \log n}$ is a lower bound on the local growth rate $\frac{|N(\{i\}) \cap \underline{B}_{k+1}|}{|N(\{i\})|}$ of vertex $i \in \underline{B}_k$.
- $z_{n,m} = \sqrt{\frac{32m \log(n+m)}{nd_{n,m}}}$ is the deviation used in the Chernoff bound.
- The sequence of numbers $\{D_k\}_{k=0}^{K_{n,m}}$ is set to be

$$D_k = \frac{4k}{c_{n,m}} \sqrt{\frac{2m \log(n+m)}{(1-z_{n,m})nd_{n,m}}} \quad \text{for } k = 0, 1, \dots, K_{n,m} - 1,$$

$$D_{K_{n,m}} = \frac{80K_{n,m}}{c_{n,m}^2} \sqrt{\frac{2m \log(n+m)}{(1-z_{n,m})nd_{n,m}}}.$$

- The two growing sets $\{\underline{B}_k\}_{k=0}^{K_{n,m}}$ and $\{\bar{B}_k\}_{k=0}^{K_{n,m}}$ which contains the subjects with estimation error D_k -close to $\underline{\varepsilon}$ and $\bar{\varepsilon}$ respectively are defined as

$$\underline{B}_k = \{j : \varepsilon_j - \underline{\varepsilon} \leq D_k\},$$

$$\bar{B}_k = \{j : \bar{\varepsilon} - \varepsilon_j \leq D_k\}.$$

► **Lemma 15 (Vertex Expansion).** *Regarding the task assignment graph $G(L \cup R, E) \sim \mathcal{B}(n, m, d_{n,m})$, for a fixed subset of left vertices $X \subset L$ with $|X| \leq \frac{n}{2}$, w.p. $1 - (n+m)^{-4|X|}$ it holds that*

- If $1 \leq |X| < m/d_{n,m}$, $\frac{|N(X)|}{|X|} > (1 - z_{n,m}) \left(1 - \frac{d_{n,m}|X|}{m}\right) d_{n,m}$;
- If $|X| \geq m/d_{n,m}$, $\frac{|N(X)|}{m} > 1 - z_{n,m} - e^{-1}$.

For a fixed subset of right vertices $Y \subset R$ with $|Y| \leq \frac{m}{2}$, w.p. $1 - (n+m)^{-4|Y|}$ it holds that

- If $1 \leq |Y| < m/d_{n,m}$, $\frac{|N(Y)|}{|Y|} > (1 - z_{n,m}) \left(1 - \frac{d_{n,m}|Y|}{m}\right) \frac{nd_{n,m}}{m}$;
- If $|Y| \geq m/d_{n,m}$, $\frac{|N(Y)|}{n} > 1 - z_{n,m} - e^{-1}$.

In above inequalities, $z_{n,m} = \sqrt{\frac{32m \log(n+m)}{nd_{n,m}}}$ as previously defined.

► **Lemma 16** (Local Growth of B_k). For n and m large enough, $k < K_{n,m}$ and a fixed subject $i \in B_k$, it holds w.p. $1 - 2(n+m)^{-4}$ that

- if $k < K_{n,m} - 1$, $|N(\{i\}) \cap B_{k+1}| \geq q_{n,m} |N(\{i\})|$,
where $q_{n,m} = \frac{c_{n,m} \log(\frac{m}{d_{n,m}})}{5 \log n}$ and $c_{n,m} = \frac{\exp(-(\alpha_{n,m}+1))}{4}$ as previously defined;
- if $k = K_{n,m} - 1$, $|N(\{i\}) \cap B_{k+1}| \geq \frac{75}{81} |N(\{i\})|$.

4.3 Analysis of Our Algorithm

Our algorithm uses the MLEs to predict the student's performance within the component. Based on the consistency of the MLEs, we show the ex-post error of our algorithm.

► **Theorem 17.** When Condition A is satisfied, the exam result graph is strongly connected. In this case, the MLE is unique and we have $(\text{alg}_i - \text{opt}_i)^2 \leq \frac{1}{4} \|\mathbf{u} - \mathbf{u}^*\|_\infty^2$.

Next we discuss the performance of our algorithm on several extreme cases of the task assignment graph. For example, the extremely sparse cases when $N(\{i\})$ is mutually disjoint for each student i or each student receives only $d = 1$ question. Another example is that the task assignment graph is a complete bipartite graph. In all of the above cases, our algorithm gives the same grade as simple averaging.

► **Theorem 18.** When the task assignment graph satisfies that $N(i)$ is mutually disjoint for each student i or each student receives only $d = 1$ question, our algorithm gives the same grade as simple averaging.

Proof. In both cases, the exam result graph satisfies that every SCC is a single point, thus the algorithm's output totally relies on cross-component predictions. For each student, the comparable components for each student are exactly the questions that student receives. Thus the algorithm gives the same prediction as the student's correctness on those questions. The prediction for remaining questions is the average accuracy on the assigned questions by the algorithm's rule for incomparable components. Therefore, the algorithm's grade for the student is exactly the same as simple averaging. ◀

► **Theorem 19.** When the task assignment graph is a complete bipartite graph, our algorithm gives the same grade as simple averaging.

Proof. In this case, the output of the algorithm only relies on existing edges. It directly follows that the algorithm gives the same grade as simple averaging. ◀

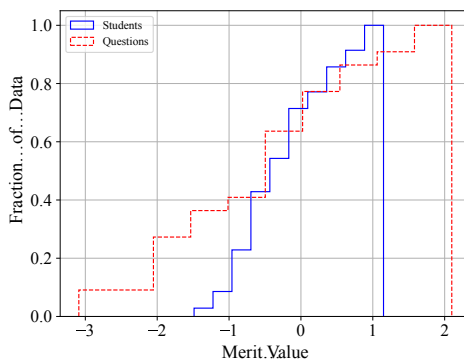
5 Experiments

5.1 Real-World Data

We use the anonymous answer sheets from a previously administered exam with $|S| = 35$ students and $|Q| = 22$ questions. The task assignment graph of the exam is a complete bipartite graph, i.e., each student is assigned with all questions. The corresponding exam result graph happens to be strongly connected, thus we are able to infer student abilities

7:12 Fair Grading Algorithms for Randomized Exams

and question difficulties (Figure 1). Below we study results from counterfactual subgraphs with real exam answers and from data generated according to the model with the inferred abilities and difficulties.



■ **Figure 1** Empirical Cumulative Distribution of Merit Value. We analyze all students and questions under the Bradley-Terry-Luce model and show the empirical cumulative density function of inferred student abilities and question difficulties. The abilities ranges from -1.486 to 1.149 while the difficulties ranges from -3.090 to 2.099.

5.2 Algorithms

Simple Averaging

The grade for student i is its average correctness on assigned questions. See the formal definition in Example 8.

Our Algorithm

The grade for student i is an aggregation of the algorithm's prediction on her performance on each question. All predictions can be classified into four cases, including existing edges (keep the fact as prediction), same component (maximum likelihood estimators), comparable components (answer in line with the path direction) and incomparable components (heuristic as simple averaging). See the formal definition in Section 3.2.

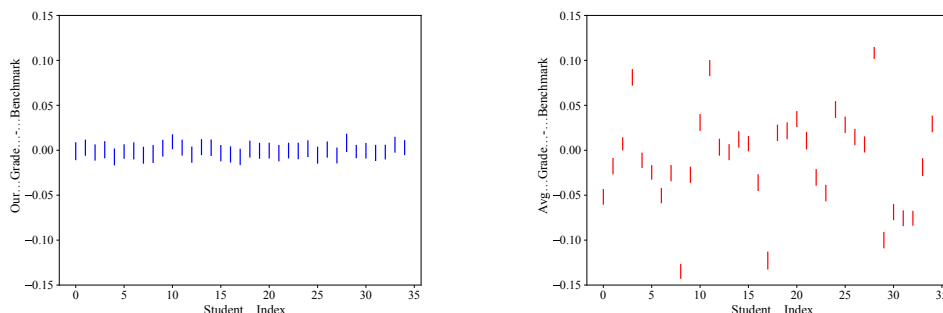
5.3 Ex-post Bias

5.3.1 Simulation 1: A Visualization of Simple Averaging's Ex-post Unfairness

We compare the ex-post bias (Definition 6) between our algorithm and simple averaging given a fixed random task assignment graph. We use inferred parameters of all 35 students and 22 questions according to Figure 1. The task assignment graph is generated with degree $d = 10$, i.e. each student is assigned 10 random questions from the whole question bank. The exam result graph is repeatedly generated according to the model.

Figure 2 shows the performance of two algorithms. The left plot corresponds to our algorithm and the right plot corresponds to simple averaging. In each plot, there are 35 confidence intervals, each corresponding to the difference between the student's expected grade and her benchmark, i.e. $\mathbb{E}_w[\text{alg}_i] - \text{opt}_i$. The confidence intervals in the left plot

are significantly closer to 0, compared to the right plot, which visualizes the intuition that students are facing different overall question difficulties under the random assignment and simple averaging fails to adjust their grades. Instead, our algorithm infers the question difficulties and the student abilities and adjusts their grades accordingly, largely reducing the ex-post bias.



(a) Ex-post Grade Deviation of Our Algorithm. (b) Ex-post Grade Deviation of Simple Averaging.

■ **Figure 2** A Visualization of the Ex-post Grade Deviation with Degree Constraint $d = 10$.

5.3.2 Simulation 2: The Effect of the Degree Constraint

We compare the expected maximum ex-post bias, i.e., $\mathbb{E}_G \left[\max_{i \in S} (\mathbb{E}_w[\text{alg}_i] - \text{opt}_i)^2 \right]$ and the expected average ex-post bias, i.e., $\mathbb{E}_G \mathbb{E}_{i \sim \mathcal{U}(S)} \left[(\mathbb{E}_w[\text{alg}_i] - \text{opt}_i)^2 \right]$ between our algorithm and simple averaging. We use inferred parameters of all 35 students and 22 questions according to Figure 1. For each degree constraint d from 1 to 22, we repeatedly generate task assignment graphs, i.e. each student is assigned d independent questions from the whole question bank. For each task assignment graph, the exam result graph is repeatedly generated according to the model.

Figure 3 shows two algorithms' expected maximum ex-post bias (Figure 3a) and expected average ex-post bias (Figure 3b) under different degree constraints, where our algorithm (blue curve) outperforms simple averaging (red curve) on every degree constraint d . Our algorithm's expected ex-post bias with the degree constraint $d = 5$ is close to simple averaging's with the degree constraint $d = 20$, which means our algorithm can ask 15 fewer questions to each student to achieve the same grading accuracy as simple averaging.

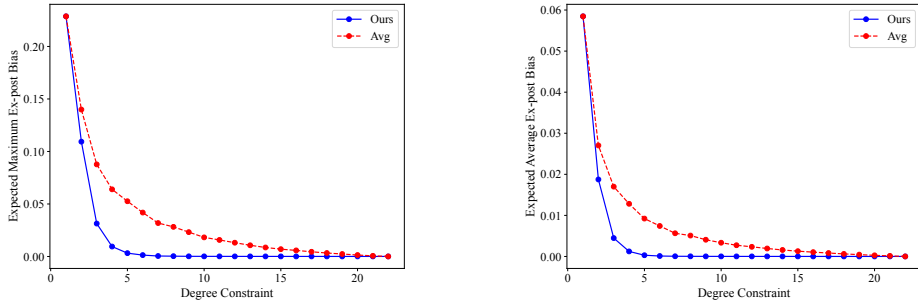
5.4 Ex-post Error and Bias-Variance Decomposition

In this part, we are investigating the expected average ex-post error (Definition 7), i.e., $\mathbb{E}_G \mathbb{E}_i \mathbb{E}_w [(\text{alg}_i - \text{opt}_i)^2]$. Through bias-variance decomposition (the proof is deferred to Appendix A), we relate the ex-post error to the ex-post bias and the variance in the algorithm performance.

► **Theorem 20** (Bias-Variance Decomposition).

$$\mathbb{E}_G \mathbb{E}_i \mathbb{E}_w [(\text{alg}_i - \text{opt}_i)^2] = \mathbb{E}_G \mathbb{E}_i [(\mathbb{E}_w[\text{alg}_i] - \text{opt}_i)^2] + \mathbb{E}_G \mathbb{E}_i \mathbb{E}_w [(\text{alg}_i - \mathbb{E}_w[\text{alg}_i])^2].$$

With the same setting in Section 5.3.1, we show the expected average ex-post error of our algorithm and simple averaging in Table 1. Our algorithm achieves a factor of 8 percent smaller ex-post error in total. But after the decomposition, we can see that our algorithm



(a) Expected Maximum Ex-post Bias. (b) Expected Average Ex-post Bias.

■ **Figure 3** Expected Aggregated Ex-post Bias v.s. Degree Constraint. The scale of expected average ex-post bias is about 4 times smaller than the scale of expected maximum ex-post bias.

achieves a factor of 99 percent smaller ex-post bias with the cost of a factor of 10 percent larger variance. In practice, students will only take the exam once, so inevitably the variance of the algorithm would contribute to the total error. Our algorithm does not focus on how to reduce variance over the noisy answering process, instead, it focuses on the expected performance of the algorithm, i.e., it makes the ex-post bias much closer to zero. To verify that our algorithm does not increase the variance too much, we also run the simulation under “the worst case” of our algorithm, i.e., all students have the same abilities and all questions have the same difficulties. In this setting, our algorithm faces the risk of over-fitting, while simple averaging works perfectly. In Table 2, we can see that both algorithms achieve ex-post biases close to 0, and our algorithm has a factor of 1.6 percent larger variance than simple averaging which is the main contribution to the difference in ex-post errors.

■ **Table 1** Bias-Variance Decomposition in the setting of real-world parameters.

	Ex-post Bias	Variance	Ex-post Error
Ours	0.00004	0.0188	0.0188
Avg	0.00331	0.0170	0.0203
Ours-Avg	-0.00327	0.0018	-0.0015

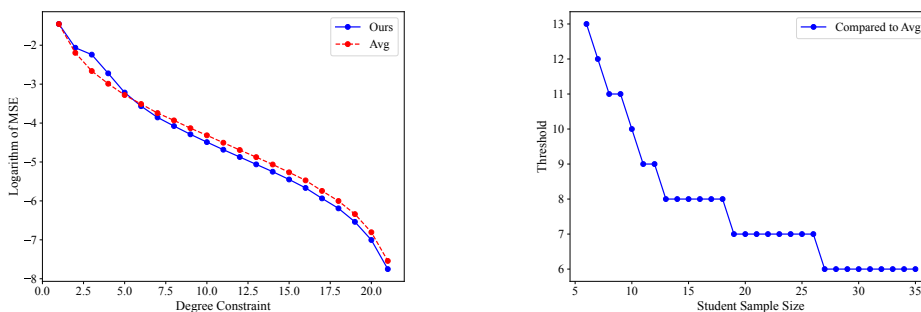
■ **Table 2** Bias-Variance Decomposition in the setting of all-the-same parameters.

	Ex-post Bias	Variance	Ex-post Error
Ours	0.0000500	0.0254	0.0255
Avg	0.0000493	0.0250	0.0250
Ours-Avg	0.0000007	0.0004	0.0005

5.5 Real-World Data Experiment: Cross Validation

We cannot repeat an exam in real world and check the ex-post bias of the algorithms. Thus, we sample part of the data we have as a new exam result graph, and use them to predict the students’ actual average on the data. We randomly split the real-world data into training data and test data. Specifically, for a fixed student sample size d_1 and a degree constraint d_2 , in each repetition, we randomly sample d_1 students and randomly choose d_2 questions and corresponding answers for each student independently as the training data, use our algorithm (Ours) and simple averaging (Avg) to predict every student’s average accuracy on the whole question bank, and calculate the mean squared error. Formally, the mean squared error MSE is defined as $MSE = \mathbb{E}_{X, \tilde{S}} \left[\frac{1}{|\tilde{S}|} \sum_{i \in \tilde{S}} \left(\text{alg}_i - \frac{1}{|Q|} \sum_{j \in Q} w_{ij} \right)^2 \right]$, where X is the training set, \tilde{S} is the sampled student set, alg_i is student i ’s grade given by the algorithm and w_{ij} is the correctness of student i ’s answer to question j .

In Figure 4a, we fix the student sample size $d_1 = 35$, i.e., $\tilde{S} = S$ and change the degree constraint d_2 from 1 to 22 and show the curve of the logarithm of MSE. Our algorithm performs better than simple averaging when the degree constraint d_2 is larger than 5 and has a factor of 16% to 20% smaller MSE compared to simple averaging when the degree constraint d_2 is larger than 10. In Figure 4b, we consider for every possible student sample size d_1 , what the smallest degree constraints d_2 is for our algorithm to perform better than simple averaging. It provides a reference for choosing the grading rule in different situations.



(a) Logarithm of MSE v.s. Degree Constraint. (b) Threshold v.s. Student Sample Size.

■ **Figure 4** Cross Validation.

6 Conclusions

We formulate and study the fair exam grading problem under the Bradley-Terry-Luce model. We propose an algorithm that is a generalization of the maximum likelihood estimation method. To theoretically validate our algorithm, we prove the existence, uniqueness, and the uniform consistency of the maximum likelihood estimators under the Bradley-Terry-Luce model on sparse bipartite graphs. Our algorithm significantly outperforms simple averaging in numerical simulation. On real-world data, our algorithm is better when the students are assigned a sufficient number of questions (i.e., on sufficiently long exams). We provide guidelines for how to choose the grading rule given certain number of students and a fixed exam length.

Our model in this paper mainly considers true-or-false questions, which can be extended to multiple-choice questions and to the case where it can be assumed that students would guess if they cannot solve a question. Our model treats student abilities and question difficulties as one-dimensional, which can be extended to a multi-dimensional model that takes different topics into account. Another potential extension of the model is to introduce different groups of students, so each question might have different difficulties for each group and we could ask for fairness across groups. Our method to treat missing edges across comparable components – which predicts 0 or 1 – needs to be improved, especially in the low-degree environment (i.e., short exam lengths where the exam result graph is unlikely to be strongly connected). Also, it would be important to provide a simple and clear explanation to students for practical use.

References

- 1 Xinming An and Yiu-Fai Yung. Item response theory: What it is and how you can use the irt procedure to apply it. *SAS Institute Inc*, 10(4):364–2014, 2014.
- 2 Haris Aziz. Simultaneously Achieving Ex-ante and Ex-post Fairness, June 2020. doi:10.48550/arXiv.2004.02554.
- 3 Moshe Babaioff, Tomer Ezra, and Uriel Feige. Best-of-Both-Worlds Fair-Share Allocations, March 2022. arXiv:2102.04909.
- 4 Gordon G. Bechtel. Generalizing the Rasch Model for Consumer Rating Scales. *Marketing Science*, 4(1):62–73, February 1985. doi:10.1287/mksc.4.1.62.
- 5 Nikolaus Bezruczko. *Rasch Measurement in Health Sciences*. JAM Press, Maple Grove, Minn, 2005.
- 6 Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons, 1952. doi:10.2307/2334029.
- 7 Luca De Alfaro and Michael Shavlovsky. Crowdgrader: A tool for crowdsourcing the evaluation of homework assignments. In *Proceedings of the 45th ACM technical symposium on Computer science education*, pages 415–420, 2014.
- 8 Craig K. Enders. *Applied Missing Data Analysis*. Methodology in the Social Sciences. Guilford Press, New York, 2010.
- 9 L. R. Ford. Solution of a Ranking Problem from Binary Comparisons. *The American Mathematical Monthly*, 64(8):28–33, 1957. doi:10.2307/2308513.
- 10 Max Fowler, David H. Smith, Chinedu Emeka, Matthew West, and Craig Zilles. Are We Fair? Quantifying Score Impacts of Computer Science Exams with Randomized Question Pools. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 1, SIGCSE 2022*, pages 647–653, New York, NY, USA, February 2022. Association for Computing Machinery. doi:10.1145/3478431.3499388.
- 11 Rupert Freeman, Nisarg Shah, and Rohit Vaish. Best of Both Worlds: Ex-Ante and Ex-Post Fairness in Resource Allocation. In *Proceedings of the 21st ACM Conference on Economics and Computation, EC '20*, pages 21–22, New York, NY, USA, July 2020. Association for Computing Machinery. doi:10.1145/3391403.3399537.
- 12 Shelby J. Haberman. Maximum Likelihood Estimates in Exponential Response Models. *The Annals of Statistics*, 5(5):815–841, September 1977. doi:10.1214/aos/1176343941.
- 13 Shelby J. Haberman. Joint and Conditional Maximum Likelihood Estimation for the Rasch Model for Binary Responses. *ETS Research Report Series*, 2004(1):i–63, June 2004. doi:10.1002/j.2333-8504.2004.tb01947.x.
- 14 John Hamer, Kenneth T. K. Ma, Hugh H. F. Kwong, Kenneth T. K. Ma Hugh, and H. F. Kwong. A Method of Automatic Grade Calibration in Peer Assessment. In *Of Conferences in Research and Practice in Information Technology, Australian Computer Society*, pages 67–72, 2005.
- 15 Ruijian Han, Rougang Ye, Chunxi Tan, and Kani Chen. Asymptotic theory of sparse Bradley–Terry model. *The Annals of Applied Probability*, 30(5):2491–2515, October 2020. doi:10.1214/20-AAP1564.
- 16 David R. Hunter. MM algorithms for generalized Bradley-Terry models. *The Annals of Statistics*, 32(1), February 2004. doi:10.1214/aos/1079120141.
- 17 Won-Chan Lee and Guemin Lee. IRT Linking and Equating. In *The Wiley Handbook of Psychometric Testing*, chapter 21, pages 639–673. John Wiley & Sons, Ltd, 2018. doi:10.1002/9781118489772.ch21.
- 18 Francisco J. Moral and Francisco J. Rebollo. Characterization of soil fertility using the Rasch model. *Journal of soil science and plant nutrition*, 17(2):486–498, June 2017. doi:10.4067/S0718-95162017005000035.
- 19 Georg Rasch. *Probabilistic Models for Some Intelligence and Attainment Tests*. MESA Press, 5835 S, 1993.

- 20 Syed A. Raza, Wasim Qazi, Komal Akram Khan, and Javeria Salam. Social Isolation and Acceptance of the Learning Management System (LMS) in the time of COVID-19 Pandemic: An Expansion of the UTAUT Model. *Journal of Educational Computing Research*, 59(2):183–208, April 2021. doi:10.1177/0735633120960421.
- 21 Ken Reily, Pam Finnerty, and Loren Terveen. Two peers are better than one: Aggregating peer reviews for computing assignments is surprisingly accurate. In *GROUP'09 - Proceedings of the 2009 ACM SIGCHI International Conference on Supporting Group Work*, pages 115–124, January 2009. doi:10.1145/1531674.1531692.
- 22 Alexander Robitzsch. A Comprehensive Simulation Study of Estimation Methods for the Rasch Model. *Stats*, 4(4):814–836, December 2021. doi:10.3390/stats4040048.
- 23 Mehdi S. M. Sajjadi, Morteza Alamgir, and Ulrike von Luxburg. Peer Grading in a Course on Algorithms and Data Structures: Machine Learning Algorithms do not Improve over Simple Baselines, February 2016. doi:10.48550/arXiv.1506.00852.
- 24 Gordon Simons and Yi-Ching Yao. Asymptotics when the number of parameters tends to infinity in the Bradley-Terry model for paired comparisons. *The Annals of Statistics*, 27(3):1041–1060, June 1999. doi:10.1214/aos/1018031267.
- 25 Glenn Waterbury. Missing Data and the Rasch Model: The Effects of Missing Data Mechanisms on Item Parameter Estimation. *Journal of applied measurement*, 20:1–12, May 2019.
- 26 Ting Yan, Yaning Yang, and Jinfeng Xu. Sparse Paired Comparisons in the Bradley-Terry Model. *Statistica Sinica*, 22(3):1305–1318, 2012.
- 27 E. Zermelo. Die Berechnung der Turnier-Ergebnisse als ein Maximumproblem der Wahrscheinlichkeitsrechnung. *Mathematische Zeitschrift*, 29(1):436–460, December 1929. doi:10.1007/BF01180541.

A Omitted Proofs

A.1 Proof of Lemma 12

Proof. Consider any subset of vertices S with size $r \leq \frac{n+m}{2}$. Denote $X = S \cap L, Y = S \cap R, |X| = x$, thus $|Y| = r - x, |L \setminus X| = n - x, |R \setminus Y| = m + x - r$. ∂S is a random variable that can be expressed as $|\partial S| = \sum_{u \in X} \sum_{v \in R \setminus Y} A_{uv} + \sum_{u \in L \setminus X} \sum_{v \in Y} A_{uv}$, where A is the adjacency matrix of the task assignment graph G . Recall that the task assignment graph G is generated by linking $d_{n,m}$ random different vertices in R to each vertex in L . Thus for different $u_1 \neq u_2 \in L$, A_{u_1} is independent with A_{u_2} , while for a fixed $u \in L$, A_u is chosen randomly without replacement. Chernoff bound applies under such conditions, i.e., $\Pr [|\partial S| \leq \frac{1}{2} \mathbb{E}[|\partial S|]] \leq \exp\left(-\frac{\mathbb{E}[|\partial S|]}{8}\right)$. Then we lower bound $\mathbb{E}[|\partial S|]$ by $\mathbb{E}[|\partial S|] = \frac{d_{n,m}}{m} (|X||R \setminus Y| + |L \setminus X||Y|) = \frac{d_{n,m}}{m} (2x^2 + (m - n - 2r)x + nr)$. For the case where $m - n - 2r \leq 0$, i.e., $r \geq \frac{m-n}{2}$, we have

$$\begin{aligned} \mathbb{E}[|\partial S|] &= \frac{d_{n,m}}{m} (2x^2 + (m - n - 2r)x + nr) \\ &\geq \frac{d_{n,m}}{m} \left(-\frac{(m - n - 2r)^2}{8} + nr \right) = \frac{d_{n,m}r}{m} \left(-\frac{1}{2}r - \frac{1}{8} \frac{(m - n)^2}{r} + \frac{1}{2}(n + m) \right) \\ &\geq \frac{d_{n,m}r}{m} \left(-\frac{n + m}{4} - \frac{1}{4} \frac{(m - n)^2}{n + m} + \frac{1}{2}(n + m) \right) = \frac{nd_{n,m}r}{n + m} \end{aligned}$$

For the case where $m - n - 2r > 0$, i.e., $r < \frac{m-n}{2}$, we have

$$\mathbb{E}[|\partial S|] = \frac{d_{n,m}}{m} (2x^2 + (m - n - 2r)x + nr) \geq \frac{nd_{n,m}r}{m} \geq \frac{nd_{n,m}r}{n + m}.$$

Thus for any fixed set S with size $r \leq \frac{n+m}{2}$,

$$\Pr \left[|\partial S| \leq \frac{d_{n,m}nr}{2(n+m)} \right] \leq \Pr \left[|\partial S| \leq \frac{1}{2} \mathbb{E}[|\partial S|] \right] \leq \exp \left(-\frac{\mathbb{E}[|\partial S|]}{8} \right) \leq \exp \left(-\frac{nd_{n,m}r}{8(n+m)} \right).$$

Finally, by union bound,

$$\begin{aligned} \Pr \left[\forall S \subset V, \text{ s.t. } |S| \leq n, \frac{|\partial S|}{|S|} > \frac{nd_{n,m}}{2(n+m)} \right] &= 1 - \Pr \left[\exists S \subset V, \text{ s.t. } |S| \leq n, \frac{|\partial S|}{|S|} \leq \frac{nd_{n,m}}{2(n+m)} \right] \\ &\geq 1 - \sum_{r=1}^{(n+m)/2} \binom{n+m}{r} \exp \left(-\frac{nd_{n,m}r}{8(n+m)} \right) \geq 1 - \sum_{r=1}^{(n+m)/2} \exp \left(-\frac{nd_{n,m}r}{8(n+m)} + r \log(n+m) \right) \\ &\geq 1 - \sum_{r=1}^{(n+m)/2} \exp \left(-\frac{nd_{n,m}r}{16(n+m)} \right) \geq 1 - \exp \left(-\frac{nd_{n,m}}{16(n+m)} + \log(n+m) \right) \geq 1 - \exp \left(-\frac{nd_{n,m}}{32(n+m)} \right) \end{aligned}$$

The third-to-last inequality and the last inequality hold when $d_{n,m} > \frac{32(n+m) \log(n+m)}{n}$. Note that condition (5) implies $\frac{(n+m) \log(n+m)}{nd_{n,m}} \rightarrow 0$ ($n, m \rightarrow \infty$) since $\alpha_{n,m} \geq 0$. Thus for large enough n and m ,

$$\Pr \left[\forall S \subset V, \text{ s.t. } |S| \leq n, \frac{|\partial S|}{|S|} > \frac{nd_{n,m}}{2(n+m)} \right] \geq 1 - \exp \left(-\frac{nd_{n,m}}{32(n+m)} \right) \rightarrow 1 \quad (n, m \rightarrow \infty).$$

A.2 Proof of Theorem 11

Proof. For an edge between vertex i and j in the task assignment graph G , i.e. $A_{ij} = 1$, the corresponding directed edge in the exam result graph G' goes from i to j with probability $\Pr[A'_{ij} = 1] = f(u_i - u_j) \leq \max_{1 \leq i, j \leq n+m} f(u_i - u_j) \leq \frac{1}{1 + \exp(-\alpha_{n,m})} \leq 2^{-\exp(-\alpha_{n,m})}$. By Lemma 12, under condition (5), $\Pr \left[\forall S \subset V, \text{ s.t. } |S| \leq n, \frac{|\partial S|}{|S|} > \frac{nd_{n,m}}{2(n+m)} \right] \rightarrow 1$ ($n, m \rightarrow \infty$). Now consider any subset of vertices $S \subset V$ s.t. $|S| = r \leq \frac{n+m}{2}$. The probability that all edges between S and $V \setminus S$ go in the same direction in G' is no more than $2 \left(2^{-\exp(-\alpha_{n,m})} \right)^{\frac{nd_{n,m}}{2(n+m)}}$. Thus by union bound, the probability that Condition A holds is at least $1 - 2 \sum_{1 \leq r \leq (n+m)/2} \binom{n+m}{r} 2^{-\frac{\exp(-\alpha_{n,m})nd_{n,m}}{2(n+m)}}$ $\geq 1 - 2 \left(\left(1 + 2^{-\frac{\exp(-\alpha_{n,m})nd_{n,m}}{2(n+m)}} \right)^{n+m} - 1 \right)$, which converges to 1 when $n, m \rightarrow \infty$ under condition (5). \blacktriangleleft

A.3 Proof of Lemma 15

Proof. Before proving the vertex expansion property of the task assignment graph $\mathcal{B}(n, m, d_{n,m})$, we first bound the vertex degree by Chernoff bound and union bound,

$$\forall i \in R, \quad \Pr \left[(1 - z_{n,m}) \frac{nd_{n,m}}{m} \leq |N(\{i\})| \leq (1 + z_{n,m}) \frac{nd_{n,m}}{m} \right] \geq 1 - (n+m)^{-4}, \quad (7)$$

where $z_{n,m}$ is defined above as $z_{n,m} = \sqrt{\frac{32m \log(n+m)}{nd_{n,m}}} \rightarrow 0$ ($n, m \rightarrow \infty$) under condition (5).

We define another family of random bipartite graph $\tilde{\mathcal{B}}$. Each graph in $\tilde{\mathcal{B}}(n, m, d_{n,m})$ contains n vertices in the left part, m vertices in the right part, and assigns $d_{n,m}$ random neighbors to each vertex in the left part (multi-edges are allowed). For any $X \subset L$, it's easy to see that $|N(X)|$ in $G \sim \mathcal{B}(n, m, d_{n,m})$ stochastically dominates $|N(X)|$ in $G \sim \tilde{\mathcal{B}}(n, m, d_{n,m})$.

Thus it's sufficient to prove the theorem under $\tilde{\mathcal{B}}(n, m, d_{n,m})$. On the other hand, counting $|N(X)|$ under $\tilde{\mathcal{B}}(n, m, d_{n,m})$ is the same random process as counting the number of non-empty bins after independently throwing $d_{n,m}|X|$ balls u.a.r. into m bins. By linearity of expectation over every bin, we know $\mathbb{E}[|N(X)|] = m \left(1 - \left(1 - \frac{1}{m}\right)^{d_{n,m}|X|}\right)$.

We need several lower bounds of $\mathbb{E}[|N(X)|]$ here. With the fact of $\frac{x}{2} \leq 1 - \exp(-x) \leq x$, $\forall 0 \leq x < 1$, we have $\mathbb{E}[|N(X)|] = m \left(1 - \left(1 - \frac{1}{m}\right)^{d_{n,m}|X|}\right) \geq m \left(1 - \exp\left(-\frac{d_{n,m}|X|}{2}\right)\right) \geq \frac{d_{n,m}|X|}{2}$. Therefore, using Azuma's inequality, we can lower bound $|N(X)|$, i.e.,

$\Pr[|N(X)| \leq (1 - z_{n,m})\mathbb{E}[|N(X)|]] \leq \exp\left(-\frac{z_{n,m}^2(\mathbb{E}[|N(X)|])^2}{2d_{n,m}|X|}\right) \leq (n+m)^{-4|X|}$. Also, when $|X| < m/d_{n,m}$, we have $\left(1 - \left(1 - \frac{1}{m}\right)^{d_{n,m}|X|}\right) \geq \frac{d_{n,m}|X|}{m} \left(1 - \frac{d_{n,m}|X|}{m}\right)$, thus with probability $1 - (n+m)^{-4|X|}$, $|N(X)| \geq (1 - z_{n,m})\mathbb{E}[|N(X)|] \geq (1 - z_{n,m})d_{n,m}|X| \left(1 - \frac{d_{n,m}|X|}{m}\right)$; Similarly when $|X| \geq m/d_{n,m}$, we have $\left(1 - \left(1 - \frac{1}{m}\right)^{d_{n,m}|X|}\right) \geq 1 - e^{-1}$, and $|N(X)| \geq (1 - z_{n,m})\mathbb{E}[|N(X)|] \geq (1 - z_{n,m})(1 - e^{-1})m \geq (1 - z_{n,m} - e^{-1})m$.

The proof for $Y \subset R$ is almost the same except that it's sufficient to use Chernoff bound rather than Azuma's inequality since the independence among the subjects in $N(Y)$, to have $\mathbb{E}[|N(Y)|] = n \left(1 - \left(1 - \frac{|Y|}{m}\right)^{d_{n,m}}\right) \geq n \left(1 - \exp\left(-\frac{d_{n,m}|Y|}{m}\right)\right) \geq \frac{nd_{n,m}|Y|}{2m}$. Using Chernoff bound, we can lower bound $|N(Y)|$, i.e., $\Pr[|N(Y)| \leq (1 - z_{n,m})\mathbb{E}[|N(Y)|]] \leq \exp\left(-\frac{z_{n,m}^2\mathbb{E}[|N(Y)|]^2}{2}\right) \leq (n+m)^{-4|Y|}$. Thus when $|Y| < m/d_{n,m}$, with probability $1 - (n+m)^{-4|Y|}$, $|N(Y)| \geq (1 - z_{n,m})\mathbb{E}[|N(Y)|] \geq (1 - z_{n,m})\frac{nd_{n,m}|Y|}{m} \left(1 - \frac{d_{n,m}|Y|}{m}\right)$; when $|Y| \geq m/d_{n,m}$, with probability $1 - (n+m)^{-4|Y|}$, $|N(Y)| \geq (1 - z_{n,m})\mathbb{E}[|N(Y)|] \geq (1 - z_{n,m})(1 - e^{-1})n \geq (1 - z_{n,m} - e^{-1})n$. \blacktriangleleft

A.4 Proof of Lemma 16

Proof. Pick a subject $i \in \underline{B}_k$. For any task assignment graph G and its adjacency matrix A , the corresponding adjacency matrix A' of the exam result graph is a random variable of A . Specifically, for any $A_{ij} = 1$, A'_{ij} s are independent Bernoulli random variables with probability $f(u_i - u_j)$ to be 1. In other words, $\mathbb{E}[A'_{ij}] = A_{ij}f(u_i - u_j)$. By Chernoff bound,

$$\Pr\left[\left|\sum_j A'_{ij} - \sum_j A_{ij}f(u_i - u_j)\right| \geq \sqrt{2|N(\{i\})| \log(n+m)}\right] \leq 2(n+m)^{-4}.$$

Below we use the above inequality and some analysis of function f to count the number of subjects in $N(\{i\}) \cap \underline{B}_{k+1}$. The fact we use about function f is $f'(x) = \frac{\exp(-x)}{(1+\exp(-x))^2} \leq \frac{1}{4}$

and $f'(x) \geq \frac{\exp(-(\alpha_{n,m}+1))}{(1+\exp(-(\alpha_{n,m}+1)))^2} \geq \frac{\exp(-(\alpha_{n,m}+1))}{4} = c_{n,m}$, $\forall |x| \leq \alpha_{n,m} + 1$. Thus for another

subject j such that $\varepsilon_j \leq \varepsilon_i$, by mean value theorem, we have $f(u_i^* - u_j^*) - f(u_i - u_j) = f'(\xi_{ij})(\varepsilon_i - \varepsilon_j) \leq \frac{1}{4}(\varepsilon_i - \varepsilon_j) \leq \frac{D_k}{4}$, where $\xi_{ij} \in [u_i - u_j, u_i^* - u_j^*]$.

Similarly, for a subject j with $\varepsilon_j > \varepsilon_i + D_{k+1} - D_k$, we have $f(u_i - u_j) - f(u_i^* - u_j^*) = f'(\xi'_{ij})(\varepsilon_j - \varepsilon_i) \geq c_{n,m}(D_{k+1} - D_k)$, where $\xi'_{ij} \in [u_i^* - u_j^*, u_i - u_j]$.

Since $u_i - u_j - D_{K_{n,m}} \leq u_i - u_j - (\varepsilon_j - \varepsilon_i) = u_i^* - u_j^* \leq \xi'_{ij} \leq u_i - u_j$, and $D_{K_{n,m}} \rightarrow 0$ as $n, m \rightarrow \infty$ under condition (6), $|\xi'_{ij}|$ is bounded by $\alpha_{n,m} + 1$ when n and m is large enough, thus $f'(\xi'_{ij}) \geq c_{n,m}$. Therefore, on the one hand,

$$\begin{aligned}
& \sum_{\varepsilon_j > \varepsilon_i} A_{ij} (f(u_i - u_j) - f(u_i^* - u_j^*)) \\
&= \sum_j A_{ij} (f(u_i - u_j) - f(u_i^* - u_j^*)) - \sum_{\varepsilon_j \leq \varepsilon_i} A_{ij} (f(u_i - u_j) - f(u_i^* - u_j^*)) \\
&\leq \sqrt{2N(\{i\}) \log(n+m)} + \frac{1}{4} D_k \sum_{\varepsilon_j \leq \varepsilon_i} A_{ij}.
\end{aligned} \tag{8}$$

On the other hand,

$$\begin{aligned}
& \sum_{\varepsilon_j > \varepsilon_i} A_{ij} (f(u_i - u_j) - f(u_i^* - u_j^*)) \geq \sum_{\varepsilon_j > \varepsilon_i + D_{k+1} - D_k} A_{ij} (f(u_i - u_j) - f(u_i^* - u_j^*)) \\
&\geq c_{n,m}(D_{k+1} - D_k) \sum_{\varepsilon_j > \varepsilon_i + D_{k+1} - D_k} A_{ij}.
\end{aligned} \tag{9}$$

Combining (8) and (9), we have

$$|N(\{i\}) \cap \underline{B}_{k+1}| \geq \sum_{u_j^* - u_j \leq u_i^* - u_i + D_{k+1} - D_k} A_{ij} \geq \frac{c_{n,m}(D_{k+1} - D_k) - \sqrt{\frac{2m \log(n+m)}{(1-z_{n,m})^n d_{n,m}}}}{c_{n,m}(D_{k+1} - D_k) + \frac{1}{4} D_k} |N(\{i\})|.$$

$$\text{For } k < K_{n,m} - 1, \frac{c_{n,m}(D_{k+1} - D_k) - \sqrt{\frac{2m \log(n+m)}{(1-z_{n,m})^n d_{n,m}}}}{c_{n,m}(D_{k+1} - D_k) + \frac{1}{4} D_k} |N(\{i\})| \geq q_{n,m} |N(\{i\})|.$$

$$\text{For } k = K_{n,m} - 1, \frac{c_{n,m}(D_{k+1} - D_k) - \sqrt{\frac{2m \log(n+m)}{(1-z_{n,m})^n d_{n,m}}}}{c_{n,m}(D_{k+1} - D_k) + \frac{1}{4} D_k} |N(\{i\})| \geq \frac{75}{81} |N(\{i\})|. \quad \blacktriangleleft$$

A.5 Proof of Theorem 13

Proof of Theorem 13. Denote $X_k = \underline{B}_k \cap L$ and $Y_k = \underline{B}_k \cap R$. We inductively prove the following fact, for n and m large enough, with probability $1 - (n+m)^{-2}$,

- for $1 \leq k \leq K_{n,m} - 2$, and k is odd, $|X_k|, |Y_k| \geq \left(\frac{n}{m} d_{n,m}\right)^{(k-1)/2}$;
- for $1 \leq k \leq K_{n,m} - 2$, and k is even, $|X_k| \geq \left(\frac{n}{m}\right)^{k/2} d_{n,m}^{(k-1)/2}$ and $|Y_k| \geq \left(\frac{n}{m}\right)^{k/2-1} d_{n,m}^{(k-1)/2}$;
- for $k = K_{n,m} - 1$, $|X_k|, |Y_k| \geq \frac{m}{d_{n,m}}$;
- for $k = K_{n,m}$, $|X_k| > \frac{n}{2}$, $|Y_k| > \frac{m}{2}$.

We will use the following fact,

$$\begin{aligned}
& |Y_{k+1}| \geq |N(X_k) \cap \underline{B}_{k+1}| = |N(X_k)| - |N(X_k) \cap \overline{\underline{B}_{k+1}}| \\
&\geq |N(X_k)| - \sum_{i \in X_k} |N(\{i\}) \cap \overline{\underline{B}_{k+1}}| = |N(X_k)| - \sum_{i \in X_k} (|N(\{i\})| - |N(\{i\}) \cap \underline{B}_{k+1}|), \tag{10}
\end{aligned}$$

and similarly $|X_{k+1}| \geq |N(Y_k) \cap \underline{B}_{k+1}| \geq |N(Y_k)| - \sum_{i \in Y_k} (|N(\{i\})| - |N(\{i\}) \cap \underline{B}_{k+1}|)$, to show the growth of X_k and Y_k respectively.

We only consider n and m large enough. Since $i \in \underline{B}_0$, w.l.o.g. we assume $|X_0| = 1$. If X_0 contains other subjects, we take a subset with size 1. Then by fact (10), (7) and Lemma 16, we know with probability $1 - 4(n+m)^{-4}$ that $|Y_1| \geq |N(X_0) \cap \underline{B}_{k+1}| \geq q_{n,m} |N(X_0)| > 0$. For $1 < k \leq K_{n,m} - 2$, and odd k , we prove inductively. We assume $|X_k| = \left(\frac{n}{m} d_{n,m}\right)^{(k-1)/2}$. If X_k is larger, we pick any subset with size $\left(\frac{n}{m} d_{n,m}\right)^{(k-1)/2}$. Fact (10) show that $|Y_{k+1}| \geq |N(X_k)| - \sum_{i \in X_k} (|N(\{i\})| - |N(\{i\}) \cap \underline{B}_{k+1}|)$.

By Lemma 15 and union bound over all subset of L with size $\left(\frac{n}{m} d_{n,m}\right)^{(k-1)/2}$, it holds with probability $1 - (n+m)^{-3|X_k|}$ that, $|N(X_k)| > (1 - z_{n,m}) \left(1 - \frac{d_{n,m}|X_k|}{m}\right) d_{n,m}|X_k|$.

By Lemma 16 and union bound over all possible subject $i \in X_k$, it holds with probability $1 - 2(n+m)^{-3}$ that, $\forall i \in X_k$, $|N(\{i\}) \cap \underline{B}_{k+1}| \geq q_{n,m} |N(\{i\})|$.

Therefore, with probability $1 - 3(n+m)^{-3}$ we have

$$\begin{aligned} |Y_{k+1}| &\geq |N(X_k)| - \sum_{i \in X_k} (|N(\{i\})| - |N(\{i\}) \cap \underline{B}_{k+1}|) \geq |N(X_k)| - (1 - q_{n,m}) \sum_{i \in X_k} |N(\{i\})| \\ &\geq (1 - z_{n,m}) \left(1 - \frac{d_{n,m} |X_k|}{m}\right) d_{n,m} |X_k| - (1 - q_{n,m}) d_{n,m} |X_k| \\ &\geq |X_k| \left(\frac{m}{n} d_{n,m}\right)^{1/2} \left(\left(\frac{n}{m}\right)^{1/2} (q_{n,m} - z_{n,m}) d_{n,m}^{1/2} - (1 - z_{n,m}) \frac{\left(\frac{n}{m} d_{n,m}\right)^{3/2} |X_k|}{n}\right) \\ &\geq |X_k| \left(\frac{m}{n} d_{n,m}\right)^{1/2} \left(\left(\frac{n}{m}\right)^{1/2} (q_{n,m} - z_{n,m}) d_{n,m}^{1/2} - 1\right) \end{aligned}$$

where the last inequality holds because we assume $|X_k| = \left(\frac{n}{m} d_{n,m}\right)^{(k-1)/2}$. Finally, under condition (6), we have for large enough n and m , $\left(\frac{n}{m}\right)^{1/2} (q_{n,m} - z_{n,m}) d_{n,m}^{1/2} - 1 \geq \left(\frac{n}{m}\right)^{\frac{1}{2}}$, thus $|Y_{k+1}| \geq d_{n,m}^{1/2} |X_k|$. The same calculation applies to the case of $1 < k \leq K_{n,m} - 2$ and even k . Similarly, we can prove for $1 < k \leq K_{n,m} - 2$, $|X_{k+1}| \geq \frac{n}{m} (d_{n,m})^{1/2} |Y_k|$. Therefore, we finish the proof for all $k < K_{n,m}$.

Similarly for $k = K_{n,m}$ and large enough n and m , with probability $1 - 4(n+m)^{-3}$,

$$\begin{aligned} |Y_{K_{n,m}}| &\geq |N(X_{K_{n,m}-1})| - \sum_{i \in X_{K_{n,m}-1}} (|N(\{i\})| - |N(\{i\}) \cap \underline{B}_{K_{n,m}}|) \\ &\geq |N(X_{K_{n,m}-1})| - \left(1 - \frac{75}{81}\right) \sum_{i \in X_{K_{n,m}-1}} |N(\{i\})| \geq (1 - z_{n,m} - e^{-1})m - \frac{6}{81}m > \frac{m}{2}. \end{aligned}$$

The same proof applies for $|X_{K_{n,m}}|$. To summarize, with probability $1 - (n+m)^{-2}$, $|X_{K_{n,m}}| > n/2$ and $|Y_{K_{n,m}}| > m/2$, thus $|\underline{B}_{K_{n,m}}| > (n+m)/2$. By symmetry, $|\overline{B}_{K_{n,m}}| > (n+m)/2$ with probability $1 - (n+m)^{-2}$. Then with probability $1 - 2(n+m)^{-2}$, at least one subject $i \in \underline{B}_{K_{n,m}} \cap \overline{B}_{K_{n,m}}$ lies in both $\underline{B}_{K_{n,m}}$ and $\overline{B}_{K_{n,m}}$. By definition, subject i satisfies $\varepsilon_i - \underline{\varepsilon} \leq D_{K_{n,m}}$ and $\overline{\varepsilon} - \varepsilon_i \leq D_{K_{n,m}}$, thus $\|\mathbf{u}^* - \mathbf{u}\|_\infty \leq \overline{\varepsilon} - \underline{\varepsilon} \leq 2D_{K_{n,m}}$, which tends to 0 under condition (6). ◀

A.6 Proof of Theorem 17

Proof. When the exam result graph is strongly connected, the algorithm calculates the MLEs \mathbf{u}^* and gives student i a grade of $\text{alg}_i = \frac{1}{|Q|} \sum_{j \in Q} f(u_i^* - u_j^*)$, while the ground truth probability of answering a random question correctly is $\text{opt}_i = \frac{1}{|Q|} \sum_{j \in Q} f(u_i - u_j)$. Thus we have

$$\begin{aligned} |\text{alg}_i - \text{opt}_i| &= \left| \frac{1}{|Q|} \sum_j f(u_i^* - u_j^*) - \frac{1}{|Q|} \sum_j f(u_i - u_j) \right| \leq \frac{1}{|Q|} \sum_j |f(u_i^* - u_j^*) - f(u_i - u_j)| \\ &= \frac{1}{|Q|} \sum_j |f'(\xi_{ij})| |\varepsilon_i - \varepsilon_j| \leq \frac{2}{n} \|\mathbf{u} - \mathbf{u}^*\|_\infty \sum_j |f'(\xi_{ij})| \leq \frac{1}{2} \|\mathbf{u} - \mathbf{u}^*\|_\infty, \end{aligned}$$

where the third-to-last equality is because of the mean value theorem, the next-to-last inequality is because $|\varepsilon_i - \varepsilon_j| \leq 2\|\mathbf{u} - \mathbf{u}^*\|_\infty$, and the last inequality is because $|f'(x)| \leq \frac{1}{4}$. Thus $(\text{alg}_i - \text{opt}_i)^2 \leq \frac{1}{4} \|\mathbf{u} - \mathbf{u}^*\|_\infty^2$. ◀

A.7 Proof of Theorem 20

Proof. We prove a stronger argument of the decomposition for any fixed student i and any fixed task assignment graph G ,

$$\begin{aligned}
 \forall i, G, \mathbb{E}_w[(\text{alg}_i - \text{opt}_i)^2] &= \mathbb{E}_w[(\text{alg}_i - \mathbb{E}_w[\text{alg}_i] + \mathbb{E}_w[\text{alg}_i] - \text{opt}_i)^2] \\
 &= (\mathbb{E}_w[\text{alg}_i] - \text{opt}_i)^2 + \mathbb{E}_w[(\text{alg}_i - \mathbb{E}_w[\text{alg}_i])^2] + 2\mathbb{E}_w[(\text{alg}_i - \mathbb{E}_w[\text{alg}_i]) (\mathbb{E}_w[\text{alg}_i] - \text{opt}_i)] \\
 &= (\mathbb{E}_w[\text{alg}_i] - \text{opt}_i)^2 + \mathbb{E}_w[(\text{alg}_i - \mathbb{E}_w[\text{alg}_i])^2] + 2(\mathbb{E}_w[\text{alg}_i] - \text{opt}_i) \mathbb{E}_w[(\text{alg}_i - \mathbb{E}_w[\text{alg}_i])] \\
 &= (\mathbb{E}_w[\text{alg}_i] - \text{opt}_i)^2 + \mathbb{E}_w[(\text{alg}_i - \mathbb{E}_w[\text{alg}_i])^2]. \quad \blacktriangleleft
 \end{aligned}$$

An Algorithmic Approach to Address Course Enrollment Challenges

Arpita Biswas   

Harvard University, Cambridge, MA, USA

Yiduo Ke   

Northwestern University, Evanston, IL, USA

Samir Khuller   

Northwestern University, Evanston, IL, USA

Quanquan C. Liu   

Northwestern University, Evanston, IL, USA

Abstract

Massive surges of enrollments in courses have led to a crisis in several computer science departments - not only is the demand for certain courses extremely high from majors, but the demand from non-majors is also very high. Much of the time, this leads to significant frustration on the part of the students, and getting seats in desired courses is a rather ad-hoc process. One approach is to first collect information from students about which courses they want to take and to develop optimization models for assigning students to available seats in a fair manner. What makes this problem complex is that the courses themselves have time conflicts, and the students have credit caps (an upper bound on the number of courses they would like to enroll in). We model this problem as follows. We have n agents (students), and there are “resources” (these correspond to courses). Each agent is only interested in a subset of the resources (courses of interest), and each resource can only be assigned to a bounded number of agents (available seats). In addition, each resource corresponds to an interval of time, and the objective is to assign non-overlapping resources to agents so as to produce “fair and high utility” schedules.

In this model, we provide a number of results under various settings and objective functions. Specifically, in this paper, we consider the following objective functions: total utility, max-min (Santa Claus objective), and envy-freeness. The total utility objective function maximizes the sum of the utilities of all courses assigned to students. The max-min objective maximizes the minimum utility obtained by any student. Finally, envy-freeness ensures that no student envies another student’s allocation. Under these settings and objective functions, we show a number of theoretical results. Specifically, we show that the course allocation under the time conflicts problem is NP-complete but becomes polynomial-time solvable when given only a constant number of students *or* all credits, course lengths, and utilities are uniform. Furthermore, we give a near-linear time algorithm for obtaining a constant $1/2$ -factor approximation for the general maximizing total utility problem when utility functions are binary. In addition, we show that there exists a near-linear time algorithm that obtains a $1/2$ -factor approximation on total utility and a $1/4$ -factor approximation on max-min utility when given uniform credit caps and uniform utilities. For the setting of binary valuations, we show three polynomial time algorithms for $1/2$ -factor approximation of total utility, envy-freeness up to one item, and a constant factor approximation of the max-min utility value when course lengths are within a constant factor of each other. Finally, we conclude with experimental results that demonstrate that our algorithms yield high-quality results in real-world settings.

2012 ACM Subject Classification Theory of computation → Scheduling algorithms

Keywords and phrases fairness, allocation, matching, algorithms

Digital Object Identifier 10.4230/LIPIcs.FORC.2023.8

Supplementary Material *Software*: <https://github.com/yiduo/CS-499-Khuller>
archived at `swb:1:dir:d79d76fd785cad2f69c6edf8a15ecac2d81bc363`



© Arpita Biswas, Yiduo Ke, Samir Khuller, and Quanquan C. Liu;
licensed under Creative Commons License CC-BY 4.0

4th Symposium on Foundations of Responsible Computing (FORC 2023).

Editor: Kunal Talwar; Article No. 8; pp. 8:1–8:23



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Funding Samir Khuller and Yiduo Ke gratefully acknowledge support from NSF-Award 2216970 (IDEAL Institute).

Acknowledgements We thank Prahlad Narasimhan Kasthurirangan for pointing us to a relevant reference for one of the problems we consider.

1 Introduction

This work addresses a central problem in fair resource allocation in the course allocation setting. In the algorithms community, one of the fairness objectives is to allocate resources among agents to maximize the minimum allocation to any single agent, also known as “Santa Claus” problem. In the course allocation setting, there are additional constraints to the Santa Claus problem, such as a “conflict” graph between the resources, in other words, if there is a conflict edge between two resources, then we cannot allocate that pair of resources to the same agent. Our study was motivated by the course allocation scenario since massive surges in enrollments in CS courses have led to a crisis in several computer science departments - not only is the demand for certain courses extremely high from majors, but the demand from non-majors is also very high. Much of the time, this leads to significant frustration on the part of the students who are unable to get into courses of interest, and this lead to non-uniformity in student happiness as a few students were able to successfully petition faculty to add them to their course, and other students failed to get into any course of interest (leading to further annoyance when finding out that you did not get in, but your friend did). As registration opens up, there is always a mad scramble to enroll in courses. Given the amount of money spent by students on fees, and due to the scale of the problem, we set out to collect the information from students about which courses they want to take, and then developed optimization models for assigning students to available seats. What makes this problem complex is that courses themselves have time conflicts, so a student might be interested in two courses, but if they meet at overlapping times, they can only take one of those courses. Moreover, students have credit caps, that limit how many courses a student can enroll in, and naturally, courses have limited capacity. Students specify a set of courses that they are interested in, and we care about total utility (assigned seats), as well as fairness measured by both the lowest allocation to any student in an assignment and envy-freeness.

While our motivating example was assigning seats to students in a fair manner, this is a pretty general resource allocation problem with some additional constraints capturing conflicts among courses and capacity constraints of students. We represent the conflict using a *conflict graph* where resources are the nodes and an edge between two resources implies that those two resources cannot be assigned to the same student.

The problem when the conflict graph is unrestricted is NP-hard (Appendix A). Thus, we focus on the case of assigning resources that can be represented as intervals. Each interval has a start and end time. We assume that time occurs in discrete integer time steps in increments of 1 beginning with step 0. Overlapping intervals are those that strictly overlap (an interval ending at time 3 does not overlap with another interval that starts at 3). The conflict graph is now determined by the overlapping structure: if two resources (intervals) overlap in time, then there is an edge between them in the corresponding conflict graph.

1.1 Related Work

The problem of allocating resources among a set of n agents with an egalitarian objective (maximizing the total value of items allocated to the worst-off agent) has been well-studied in the literature and is known as the Santa Claus problem. This problem was introduced

by Bansal and Sviridenko [3] and they developed a $O(\log \log n / \log \log \log n)$ approximation algorithm. Later, Davies et al. [16] improved it to a $(4 + \epsilon)$ -approximation. More recently, Chiarelli et al. [15] considered the Santa Claus problem assuming conflicting items represented by a conflict graph. They analyzed the NP-hardness of the problem for specific subclasses of conflict graphs and provided pseudo-polynomial solutions for others. Our work complements their results by providing constant approximate (polynomial time) solutions for interval graphs with uniform and binary valuations for course allocation.

Another well-studied fairness criterion in the fair division literature is envy-freeness [17], where every agent values her allocation at least as much as she values any other agent's allocation. However, envy-freeness does not translate well when the items to be allocated are indivisible (for example, if there is one indivisible item and two students, the item can be allocated to only one student, and the other student would envy). Thus, for indivisible items (such as course seats), an appropriate fairness criterion is envy-freeness up to one item (EF1), defined by Budish [12]. Prior works have shown that an EF1 allocation always exists while allocating non-conflicting budgeted courses [12], under submodular valuations [31], under cardinality constraints [7], conflicting courses with monotone submodular valuations and binary marginal gains over the courses [4, 34], and many more. However, these results do not consider interval graphs to model conflicting courses and thus, the existing EF1 solutions cannot solve the fair course allocation problem that we consider. Recent work by Hummel et al. [24] explored the allocation of conflicting items with EF1 fairness criteria. They showed the existence of EF1 for conflict graphs with small components and refuted the existence of EF1 when the maximum degree of the conflict graph is at least as much as the number of agents. Moreover, they provided a polynomial time EF1 solution when the conflict graph consists of disjoint paths and the valuations are binary. Our work extends their results by providing a polynomial time EF1 solution for interval graphs with binary valuations, which are more general than disjoint-path graphs and capture conflicts between courses.

Fair allocation of intervals has been studied in job scheduling problems, where each job is represented as an interval (with a starting time, deadline, and processing time) and is required to be allocated to machines such that the same machine is not scheduled to run another job at the same time. Fairness notions considered are in terms of load balancing [2], waiting time envy freeness [6], completion time balancing [25], and EF1 among machines [30]. However, these papers allow flexible time intervals, which cannot capture conflicts as graph edges and represent a different problem from our work.

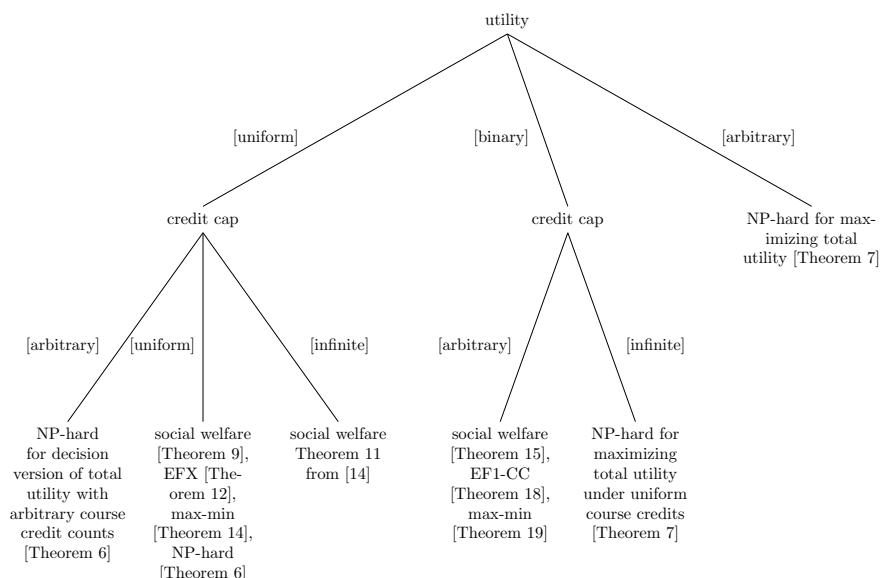
Other related techniques to our fair course allocation problem include equitable coloring [8, 27, 20], bounded max coloring [10, 23], mutual exclusion scheduling [18, 26, 33], although most of these works are only tangentially related to our problem at hand. There have also been many works on approximation algorithms for various different types of conflict models [9, 13, 29, 32] and resource constrained scheduling [5] but none of these works operate in the specific conflict graph and allocation model studied in our paper.

1.2 Summary of Contributions

In this paper, we tackle the problem of *fair* allocation of conflicting resources. We prove that a general version of the problem is NP-hard via a reduction from the independent set problem in Appendix A. This motivates the study of a specific class of conflict graphs, namely interval graphs, which capture the course allocation problem. For interval graphs, we provide polynomial time algorithms to obtain a fair allocation. We establish that, oftentimes, *simple*

algorithms are enough to provide multiple guarantees in terms of efficiency and fairness, specifically, a round robin approach is often sufficient. Figure 1 summarizes our results. Our main results are:

- We first consider *uniform utilities* in Section 3 and show that the course allocation under the time conflicts problem with the objective of maximizing social welfare is NP-complete in general. However, we develop polynomial-time solutions when there are a constant number of students *or* when the credit caps and course lengths are uniform. We further provide solutions that have fairness guarantees, one of which satisfies envy-freeness up to any good (EFX) and the other achieves approximate maxi-min fairness.
- We then investigate *binary utilities and uniform credits for all courses* in Section 4 and develop a $(1/2)$ -approximate solution for the course allocation problem under the time conflicts problem with the objective of maximizing social welfare. We further provide solutions that have fairness guarantees, one of which satisfies envy-freeness up to one good (EF1) and the other achieves approximate max-min fairness.
- Our experimental evaluation demonstrates that our algorithms yield near-optimal solutions on synthetic as well as real-world university datasets.



■ **Figure 1** Overview of results.

2 Preliminaries

In this section, we define our problem as well as the necessary concepts for our results. We first define our main problem which we call the Course Allocation Under Time Conflicts problem (CAUTC). This problem describes an issue almost all universities face: given a set of courses that have meeting times during the week and student preferences over these courses, what is the best way to assign these courses to students? Each course has a seating capacity, after all. From a university’s perspective, filling seats has value (maximizing utility), but we have to balance that with a fairness aspect as well.

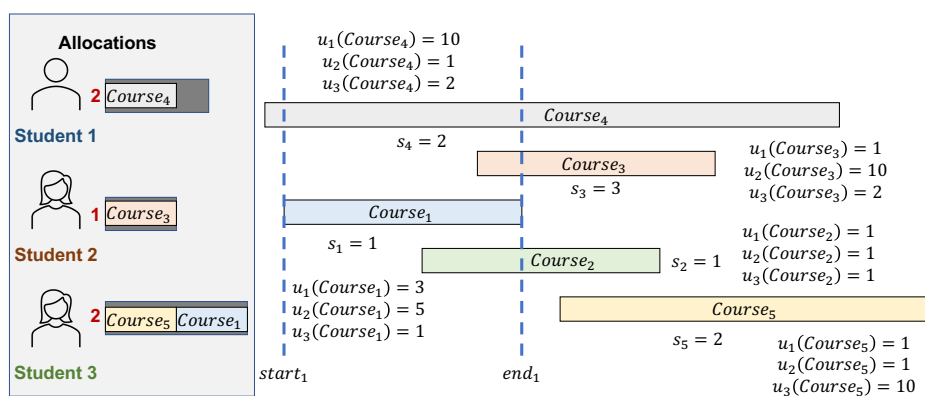
2.1 Course Allocation under Time Conflicts Model

We consider the problem of allocating a set of m courses among a set of n students. Let \mathcal{N} be the set of students and \mathcal{M} be the set of courses. Courses in \mathcal{M} have indices in \mathcal{M} . Each student $i \in \mathcal{N}$ has a non-negative utility for each course $j \in \mathcal{M}$; this utility is denoted by $u_i(j) \geq 0$. C_i represents the maximum number of credits a student i can take. Each course j has a certain number of credits indicated by c_j , a seat capacity of s_j for each $j \in \mathcal{M}$, a start and end time, represented by the tuple $(start_j, end_j)$ and a duration d_j (in units consisting of discrete time steps). Finally, each course j is associated with a seat count s_j . Therefore, the restrictions are:

- A student $i \in \mathcal{N}$ can be matched to courses with the total credits at most C_i (*credit cap*).
- A course $j \in \mathcal{M}$ can be allocated to at most s_j students.
- No student can be allocated a pair of courses that overlap in time.

Although we define the problem in the most general form, for the rest of this paper, we set $c_j = 1$ for all courses. Furthermore, we reduce to the equivalent problem where we make a copy of the course for each seat and create an interval with the same start and end time for each seat of the course. Via this reduction, we also set $s_j = 1$ for all courses.

The course schedule can be represented as an interval graph. We illustrate such a configuration in Figure 2.



■ **Figure 2** An CAUTC instance with 3 students and 5 courses, with one seat per course. All courses conflict with each other except for Course₁ and Course₅. The red numbers students indicate the credit caps for students. The allocation represents a solution for CAUTC-SW (Definition 1).

2.2 Fairness Measures

We first consider the problem of finding an allocation that maximizes the social welfare (total sum of utilities of all the students based on the courses allocated) subject to all the feasibility and non-conflicting constraints. We call this maximization problem CAUTC-SW.

► **Definition 1** (CAUTC-SW). *Given a set of students \mathcal{N} , a set of courses \mathcal{M} , and the set of utility functions \mathcal{U} , CAUTC-SW is the assignment of courses to students such that the social welfare is maximized and the constraints of CAUTC are satisfied.*

In addition to maximizing social welfare, we also consider a number of common fairness measures as constraints. We first define them here but will slightly modify some of these definitions in their respective sections later on in this paper.

We first define the concept of *envy-free up to any good* (EFX). Informally, EFX means that if any agent A were to be envious of any agent B, then A would no longer be envious if any one item were to be removed from agent B's allocation.

► **Definition 2** (Envy-Free Up to Any Good (EFX)). *For all students $i \in \mathcal{N}$, if there exists an $i' \in \mathcal{N}$ such that $u_i(A_{i'}) > u_i(A_i)$, then for all items $x \in A_{i'}$, it follows that $u_i(A_{i'} \setminus x) \leq u_i(A_i)$ or $C_i = \sum_{j \in A_i} c_j$ (student i has reached their credit cap), where A_k denotes the allocation of courses to student k .*

A slightly weaker version of EFX is *envy-free up to one good* (EF1), defined below. Informally, EF1 means that if any agent A were to be envious of any agent B, then A would no longer be envious if a particular item were to be removed from agent B's allocation.

► **Definition 3** (Envy-Free Up to One Good (EF1)). *For all students $i \in \mathcal{N}$, if there exists an $i' \in \mathcal{N}$ such that $u_i(A_{i'}) > u_i(A_i)$, then there exists an item $a \in A_{i'}$ satisfying $u_i(a) > 0$, such that $u_i(A_{i'} \setminus a) \leq u_i(A_i)$ or $C_i = \sum_{j \in A_i} c_j$ (student i has reached their credit cap), where A_k denotes the allocation of courses to student k .*

The problem with only ensuring EF1 is that there is a trivial allocation of courses consisting of giving everyone one course only or no courses. Such an allocation is EF1 since no one envies anyone else by more than one course. However, such an allocation is not a very useful allocation most students would not receive as many courses as they want and there will be many remaining courses. Thus, we need a better measure of envy. A definition from [30] resolves this problem. Suppose all unassigned courses in each iteration were donated to a dummy student, the *charity*, who is unable to envy anyone, but students are able to envy the charity. Then, having the charity resolves the issue of trivial solutions. Specifically, any student i can envy the charity by considering the maximum independent set among the courses in the charity that are desired by i . If such a maximum independent set is larger than the number of courses allocated to i , then i envies the charity. We formally define EF1 Considering Charity (EF1-CC) to be our new notion of envy below.

► **Definition 4** (Envy-Free Up to One Good Considering Charity (EF1-CC)). *Any student i who has reached their credit cap (i.e. $C_i = \sum_{j \in A_i} c_j$) does not envy anyone else. For all other students $i, i' \in \mathcal{N}$ (who have not reached their credit caps) and given an allocation $\mathcal{A} = (A_1, \dots, A_i, \dots, A_n)$ of courses, it holds that $|\{j \mid u_i(j) > 0, j \in A_i\}| \geq |\{j \mid u_i(j) > 0, j \in A_{i'}\}| - 1$. Let D be the set of courses that are unassigned and held by a dummy student defined as the *charity*. Let $MIS_i = MIS(\{j \mid u_i(j) > 0, j \in D\})$ be the maximum independent set of courses in D that are desired by student i . Then, for all students $i \in \mathcal{N}$, it holds that $|\{j \mid u_i(j) > 0, j \in A_i\}| \geq |MIS_i| - 1$.*

Finally, we consider a Santa Claus fairness objective which is to maximize the minimum allocation of courses to any student. For simplicity, we denote this problem as CAUTC-SC.

► **Definition 5** (CAUTC-SC). *Determine an allocation of courses to students $\mathcal{A} = (A_1, \dots, A_n)$ that maximizes the minimum utility of any student subject to the constraints of CAUTC. Namely, we seek to satisfy the following objective $\max_{\mathcal{A}} \left(\min_{i \in \mathcal{N}} \left(\sum_{j \in A_i} u_i(j) \right) \right)$.*

3 Uniform Utilities for Courses

In this section, we discuss the setting where all students have equal, uniform preferences for all courses. In other words, in this section, all students have preference 1 for every course. In this setting, we show a number of hardness, social welfare, and fairness results described in the following sections.

3.1 Hardness of CAUTC-SW under Uniform Utilities

We show that CAUTC-SW is NP-hard (Theorem 6). Subsequently, we consider some variants of the problem that are polynomial-time solvable in the following sections.

► **Theorem 6.** *The CAUTC-SW problem where the utilities are uniform, credit caps are uniform, course are non-overlapping, and number of credits for each course is non-uniform and arbitrary is NP-hard.*

We prove this via a reduction from the 3-partition problem (Appendix B.1).

► **Theorem 7.** *The CAUTC-SW problem where utilities are binary, credit caps are infinite, and number of credits for each course is uniform is NP-hard.*

We prove this via a reduction from the k -coloring problem for circular-arc graphs. The complete proof is in Appendix B.2.

3.2 Maximizing Social Welfare

In this section, we show that, for some more restricted settings, the CAUTC-SW problems are polynomial-time solvable. We first show that when given a constant number of students, we can efficiently solve the most general form of the problem with no restrictions on either the credit caps or the number of credits for each course, and with arbitrary preferences for each student.

■ **Algorithm 1** Round Robin Algorithm for CAUTC-SW.

Require: Set of students \mathcal{N} , set of courses \mathcal{M} , uniform (unit) utilities

Ensure: Assignment of courses to students.

```

1: function ROUNDROBIN( $\mathcal{N}$ ,  $\mathcal{M}$ )
2:   Sort  $\mathcal{M}$  chronologically by earliest finish time.
3:   Initialize student assignments  $\mathcal{A}$  to empty sets.    ▷ each student starts out with no
   courses
4:   for course  $j \in \mathcal{M}$  in sorted order do
5:     Let  $T = \{s \mid |A_s| < C_s, \text{no course in } A_s \text{ conflicts with } j\}$ .
6:     if  $|T| > 0$  then
7:       Let  $s = \min_{s' \in T} (|A_{s'}|)$  (breaking ties by student index).
8:       Update  $A_s = A_s \cup \{j\}$     ▷ Assign course  $j$  to student  $s$ 
9:   return  $\mathcal{A}$ 

```

► **Theorem 8.** *CAUTC-SW is polynomial-time solvable when there are only a constant number of students and credit counts for courses can be distinct but are each $O(1)$.*

The proof of Theorem 8 can be found in Appendix B.3.

► **Theorem 9.** *Algorithm 1 solves CAUTC-SW in $O((n + m) \log n)$ time when there are (1) uniform credits for all courses, i.e. $c_j = c_{j'}$ for all $j, j' \in \mathcal{M}$, (2) uniform course lengths, i.e., $d_j = d_{j'}$ for all $j, j' \in \mathcal{M}$, and (3) uniform utilities i.e., $u_i(j) = u_{i'}(j)$ for all $i, i' \in \mathcal{N}$.*

We prove Theorem 9 via a variation of the greedy-comes-first strategy; we present our full proof in Appendix B.4. When the durations of the courses are not uniform, we can obtain a $(1/2)$ -approximate allocation for CAUTC-SW.

► **Lemma 10.** *There is a $O((n + m) \log n)$ time round-robin algorithm for CAUTC-SW that obtains a $1/2$ -approximation when there are (1) n students, (2) uniform credit caps i.e. for any pair of students $i, i' \in \mathcal{N}$, we have $C_i = C_{i'}$, and (3) uniform utilities i.e. for any pair of students $i, i' \in \mathcal{N}$ and jobs $j, j' \in \mathcal{M}$, we have $u_i(j) = u_{i'}(j')$.*

Proof. We use the same algorithm as before, given in Algorithm 1. However, we use a slightly different analysis which is somewhat more complicated than our utility proof before but with the same essential flavor of proof using D_i, J_i, B_i . Namely, the one additional property we prove is that when $|B_i| + |D_i| \geq |J_i|$, our new greedy algorithm will pick $|J_i|$ instead of $B_i \cup D_i$. Suppose for contradiction that i picked $B_i \cup D_i$ instead of J_i , then i must have picked a course with *earlier or the same end time* as each of the courses in J_i . We now show that $|B_i \cup D_i| \geq |J_i|$. We prove this through the classic greedy stays ahead proof technique. If one were to chronologically order $B_i \cup D_i$ by finish time and also chronologically order J_i by finish time, and call the two ordered sets as P and Q , respectively, and let P_i denote the i -th course in set P ; we will prove that it is always true that for all indices $i \leq |J|$, $f(P_i) \leq f(Q_i)$, where $f(x)$ means the finish time of course x . Also define the start time function of course x as $s(x)$. The base case of $i = 1$ is obviously true due to the nature of the algorithm. Now for the inductive case, assume inductive hypothesis $f(P_i) \leq f(Q_i)$ and we want to prove $f(P_{i+1}) \leq f(Q_{i+1})$. We know that $f(Q_i) \leq s(Q_{i+1})$. Combining this with the inductive hypothesis, we get $f(P_i) \leq s(Q_{i+1})$, so Q_{i+1} is available for our algorithm to choose, and since our algorithm chooses an available course with the earliest end time, $f(P_{i+1}) \leq f(Q_{i+1})$.

Let's assume for the sake of contradiction that $|J| > |B_i| \cup |D_i|$. Through the same argument as in the inductive case above, say $|B_i| \cup |D_i| = p$, then the start time of Q_{p+1} must have a start time later than the finish time of the last course in $|B_i| \cup |D_i|$, i.e. $s(Q_{p+1}) \geq f(P_p)$, but that means our algorithm would have selected Q_{p+1} (some time) after selecting P_p , a contradiction. ◀

For completeness, we state the following form formulation of CAUTC-SW that is solved via an interval coloring algorithm of Carlisle and Lloyd [14].

► **Theorem 11** ([14]). *CAUTC-SW can be solved in polynomial time when there are (1) n students, (2) no credit caps i.e., $C_i = m$, and (3) uniform utilities i.e. for any pair of students $i, i' \in \mathcal{N}$, we have $u_i(j) = u_{i'}(j)$.*

3.3 Guaranteeing Envy-Freeness Up to Any Good

Maximizing seat occupancy is a reasonable objective only from a financial perspective for the university, but oftentimes, maximizing seat occupancy could result in highly unfair schedules for the students. For example, student A might get all of his favorite courses while student B gets none of his desired courses. We, therefore, consider CAUTC-SW under several fairness notions, such as envy-free up to any good (Definition 2) and envy-free up to one good (Definition 3).

► **Theorem 12.** *There is an $O((n + m) \log n)$ -time algorithm for CAUTC-SW that is EFX when there are (1) n students, (2) uniform credit caps i.e. for any pair of students $i, i' \in \mathcal{N}$, we have $C_i = C_{i'}$, and (3) uniform utilities i.e. for any pair of students $i, i' \in \mathcal{N}$ and any pair of jobs $j, j' \in \mathcal{M}$, we have $u_i(j) = u_{i'}(j')$.*

Proof. Our algorithm is the same round robin algorithm given in Algorithm 1. We first prove the following lemma.

► **Theorem 13.** *When student i is no longer able to choose a feasible course, there will be at most $n - 1$ courses that can be assigned after i 's turn and each of these courses is assigned to a different student.*

Proof. Because utilities are uniform, if student i is no longer able to choose a course, this means that all remaining courses conflict with the courses they are assigned. Suppose the last course that is assigned to student i is course j . Because we are assigning courses in Algorithm 1 in a round robin manner in an order determined by non-decreasing end time, all remaining courses (yet to be considered by the algorithm) that can be assigned have end time no earlier than the end time of j . Let this set of courses be A . Since i is no longer able to receive a course, either there remains only $n - 1$ courses or A has at least $n - 1$ courses and at least $|A| - n + 1$ courses in A all conflict with j . Since all courses in A have end time no earlier than the end time of j , these $|A| - n + 1$ courses all conflict with each other. In either of these two cases, at most $n - 1$ courses can be assigned after i 's turn. Furthermore, these courses are assigned to different students. If there are at most $n - 1$ courses in A , then by nature of the algorithm, these courses all have end times later than the end times of courses assigned to students; furthermore, the ending time of the last course assigned to each student can be no later than the end time of j by the nature of our algorithm. Hence, two such courses can be assigned to one student, then one of these courses can be assigned to j . Thus, since we are assigning courses to a student with the fewest number of courses, each of these courses is assigned to a different student. Finally, all additional $|A| - n + 1$ courses all conflict with each other and hence no two of these courses can be assigned to the same student. ◀

Hence, by the time the algorithm completes and by Theorem 13, the cardinalities of all students' allocations are within one of each other, therefore achieving EFX. ◀

3.4 Maximizing Max-Min Objective

In this section, we consider the max-min objective, Santa Claus (SC) problem (Definition 5). We first show that our algorithm in Algorithm 1 gives a $(1/4)$ -approximate CAUTC-SC allocation. Specifically, we prove the following.

► **Lemma 14.** *There is a $O((n + m) \log n)$ time round robin algorithm (Algorithm 1) for CAUTC-SC that obtains a $(1/4)$ -approximation when there are (1) n students, (2) uniform credit caps i.e. for any pair of students $i, i' \in \mathcal{N}$, we have $C_i = C_{i'}$, and (3) uniform utilities i.e. for any pair of students $i, i' \in \mathcal{N}$ and jobs $j, j' \in \mathcal{M}$, we have $u_i(j) = u_{i'}(j')$.*

Proof. Given a set of courses with total utility U , the max-min value of any allocation is at most $\lfloor \frac{U}{n} \rfloor$. We now consider two possible cases with respect to the values of $\lfloor \frac{U}{n} \rfloor$. First, we consider the case when $\lfloor \frac{U}{n} \rfloor \geq 2$. In this case, by Theorem 13, the max-min value of our allocation is at least $\frac{U}{2n} - 1 \geq \frac{U}{4n}$. Now, we consider the case when $\lfloor \frac{U}{n} \rfloor < 2$. In this case, either the max-min value is 0 or the max-min value is 1. If the max-min value is 0, then we trivially obtain our approximation since any allocation will result in the correct approximation. Otherwise, if the max-min value is 1, then there is one student who gets only one course. We show that if the max-min value is 1, then our algorithm also allocates at least one course to every student. The criteria for our algorithm giving one course to each student is that there exists at least n courses. Since our algorithm assigns the courses in a round robin manner, if there are at least n courses, then our algorithm will assign at least one course to each student. In order for the max-min value to be 1, there must exist at least

n courses; hence, the max-min value of allocations given by our algorithm matches that of the value given in OPT. Thus, by the two cases we just showed, the approximation factor is at least $\frac{U}{\frac{4n}{U}} = \frac{1}{4}$. ◀

4 Binary Preferences for Classes with Uniform Credits

In this section, we discuss the setting where students have binary preferences for courses. This is a very realistic setting since it is often the case that students want to take certain courses and not others. We denote the binary preferences of the students as $U : \mathcal{N} \times \mathcal{M} \mapsto \{0, 1\}$, where $u_i(c) = 1$ denotes that the student $i \in \mathcal{N}$ wants to take the course c , and $u_i(c) = 0$ denotes that course c is not desired by student i . If a student has $u_i(c) = 1$, then we say that student i *desires* course c ; otherwise, we say that student i does not desire course c . Each student i has a credit cap denoted by C_i . In this section, all courses have uniform number of credits; i.e. all courses have the same number of credits. Because of this assumption, we can assume all courses are 1 credit each and we scale the credit caps of each student to the maximum number of courses that can fit in the student's schedule.

4.1 Maximizing Social Welfare

We first present an algorithm that gives an approximation for CAUTC-SW given binary preferences. Our algorithm proceeds as follows. Sort the students by credit cap from largest credit cap to smallest (Line 2). Then, we iterate the following procedure. Let the current student be the first student in the sorted order of the students by credit cap with no assigned courses (Line 4). We find an independent set of maximum size among all courses with non-zero utility for the current student (Line 5). For each independent set I and the associated student $i \in \mathcal{N}$, we sort the courses in I and give the first $\max(|I|, C_i)$ courses in I in the sorted order to student i (Lines 7, 8, 9). Finally, we remove the allocated courses from the set of available courses (Line 10).

■ **Algorithm 2** Binary Utilities Algorithm for CAUTC-SW.

Require: Set of students \mathcal{N} , set of courses \mathcal{M} , binary utilities U

Ensure: Assignment of courses to students.

```

1: function MAXINDEPENDENTSETRoundRobin( $\mathcal{N}, \mathcal{M}, U$ )
2:   Sort  $\mathcal{N}$  in non-increasing order by credit cap.
3:   Initialize student assignments  $\mathcal{A}$  to empty sets. ▷ student starts out with no courses
4:   for student  $i \in \mathcal{N}$  in sorted order do
5:     Let  $I = MIS(\{j \mid j \in \mathcal{M}, u_i(j) > 0\})$ .           ▷ Find MIS in remaining courses.
6:     if  $|I| > C_i$  then
7:       Sort  $I$  by end time.
8:       Set  $I \leftarrow I[C_i]$ .           ▷ Resize the MIS to be the first  $C_i$  courses in the MIS.
9:       Set  $A_s \leftarrow I$ .
10:      Update  $\mathcal{M} = \mathcal{M} \setminus I$ .           ▷ Remove assigned courses.
11:  return  $\mathcal{A}$ 

```

► **Theorem 15.** *Algorithm 2 solves CAUTC-SW in $O(n^2)$ time with an $(1/2)$ -approximation when there are n students, arbitrary credit caps C_i for all $i \in \mathcal{N}$, unit credits per course $c_j = 1$ for all $j \in \mathcal{M}$, and binary utilities for all students, i.e. $u_i(j) \in \{0, 1\}$ for all $i \in \mathcal{N}$.*

Proof. In the sorted order of courses by end time in I , if course $j \in \mathcal{M}$ is assigned in OPT and by our algorithm, then we skip this course in our analysis. However, if the course is assigned in OPT but not assigned by our algorithm, then we need to argue that either another course is assigned in its place or that we can *charge* it to another assigned course. For all of the below cases, suppose that course $j \in \mathcal{M}$ is assigned to student $i \in \mathcal{N}$ in OPT but not assigned in our assignment. For simplicity, we denote the assignment produced by our algorithm as \mathcal{A} . Let D_i be the set of courses assigned to student i in \mathcal{A} which were not assigned to any student in OPT; let B_i be the set of courses assigned to i in \mathcal{A} but assigned to $q \neq i \in \mathcal{N}$ in OPT. Finally, let J_i be the set of courses assigned to i in OPT but assigned to no student in \mathcal{A} . We consider all possible cases below.

- If $|D_i| \geq |J_i|$, then for each course in J_i , we can replace it with a course in D_i and achieve the same maximum total utility.
- If $|D_i| < |J_i|$, then we consider two additional cases:
 - It is impossible to have $|B_i| + |D_i| < |J_i|$ since $|J_i|$ is a larger independent set and would have been assigned to i instead of $B_i \cup D_i$.
 - Then, the remaining case is that $|B_i| + |D_i| \geq |J_i|$. This case is the core of our proof. In this case, we know that $|B_i| \geq |J_i| - |D_i|$. We pick an arbitrary set of $|D_i|$ jobs in J_i and replace them each with a unique job in D_i . This does not change the optimum total utility value. Now, we charge each of the remaining $|J_i| - |D_i|$ jobs in J_i to a job in $|B_i|$. We now count the number of “charges” that each course in $|B_i|$ gets. Since $|B_i| \geq |J_i| - |D_i|$ and we do not charge a course in B_i with any other course not in J_i , each course in B_i is charged with at most one charge resulting from a course in J_i .

We now count the number of courses assigned in both OPT and \mathcal{A} as well as the number of charges each course gets. By the cases above, each of these courses gets at most 1 charge. Hence, if each charge is added to the set of allocated courses, the utility increases by at most a factor of two. Hence, our algorithm produces a $(1/2)$ -approximation. ◀

4.2 Guaranteeing Envy-Freeness Up to One Good

Given an allocation of courses to students $\mathcal{A} = (A_1, \dots, A_i, \dots, A_n)$ (where A_i is the set of courses assigned to student i), a student i is said to *envy* student i' if the number of student i 's desirable courses in A_i is less than that in $A_{i'}$, that is, $|\{j \mid u_i(j) > 0, j \in A_i\}| < |\{j \mid u_i(j) > 0, j \in A_{i'}\}|$. Similarly, an allocation \mathcal{A} is called EF1 when for every pair of students $i, i' \in \mathcal{N}$, the following holds: $|\{j \mid u_i(j) > 0, j \in A_i\}| \geq |\{j \mid u_i(j) > 0, j \in A_{i'}\}| - 1$. Note that in the binary valuation setting, EF1 implies that, removing *any* course that i desires from $A_{i'}$ results in i no longer envying i' . We provide an algorithm (Algorithm 3) and prove that this algorithm satisfies the stronger fairness criterion called EF1-CC (Definition 4).

Our algorithm is a simple modification of the round-robin algorithm given in Algorithm 1. The only change we make to the algorithm is that when we perform the round-robin assignment, each course is iteratively assigned to only one of those students who have non-zero utility for the course, in addition to ensuring that the selected student has the minimum number of current courses, has not reached credit cap and has no conflict with the course. Our modified pseudocode is given in Algorithm 3.

Specifically, Algorithm 3 first sorts the courses chronologically by finish time (Line 2). Then, we iterate over the courses one by one in the sorted order of finish time (Line 4). Among the students who have non-zero preference for the course, have not reached their credit caps, and have no conflicts with the course (Line 5), we select a student (breaking ties arbitrarily) with the least number of assigned courses among these students (Line 7). Finally, we assign the course to the student (Line 8).

■ **Algorithm 3** Round Robin Algorithm for EF1-CC Allocation with Binary Utilities.

Require: Set of students \mathcal{N} , set of courses \mathcal{M} , binary utilities U

Ensure: EF1-CC Allocation for Binary Utilities

```

1: function EF1CCROUNDROBIN( $\mathcal{N}$ ,  $\mathcal{M}$ ,  $U$ )
2:   Sort  $\mathcal{M}$  chronologically by earliest finish time.
3:   Initialize student assignments  $\mathcal{A}$  to emptysets.  $\triangleright$  students start out with no courses
4:   for course  $j \in \mathcal{M}$  in sorted order do
5:     Let  $T = \{s \mid u_s(j) = 1, |A_s| < C_s, \text{no course in } A_s \text{ conflicts with } j\}$ .
6:     if  $|T| > 0$  then
7:       Let  $s = \min_{s' \in T} (|A_{s'}|)$  (breaking ties arbitrarily).
8:       Update  $A_s = A_s \cup \{j\}$   $\triangleright$  Assign course  $j$  to student  $s$ 
9:   return  $\mathcal{A}$ 

```

► **Theorem 16.** *Under binary preferences, uniform credits for all courses, and arbitrary credit caps, the round-robin algorithm given in Algorithm 3 produces an EF1 allocation.*

Proof. We prove by induction that for any two students s and s' , student s never envies s' by more than one course throughout the entirety of Algorithm 3. The induction is on the finish time of each course in the schedule of s' among the set of courses for which s has non-zero utility, i.e. we induce on the finish times of the set of courses $L = [j \in A_{s'} \mid u_s(j) > 0]$ sorted from earlier to later times. Notice that L is the set of courses assigned to s' that are desired by s , as courses assigned to s' not desired by s cannot make s envy s' and therefore irrelevant to this proof. Now, for each $i \in [|L|]$, we consider the set of courses assigned to both s and s' which has end time no later than the end time of $L[i]$. For simplicity, we use the phrase *by the time course $L[i]$ ends* to mean that we consider the set of courses held by s and s' with end time no later than $L[i]$.

► **Lemma 17.** *For each course $L[i]$ for all $i \in [|L|]$, at the time $L[i]$ ends, student s envies s' by at most one course.*

Proof. We prove via induction on the i -th course of L which ends at time e_i . The base case is when $i = 1$. Student s trivially envies s' by at most 1 because if s has no courses by the time course $L[1]$ ends then s will only envy s' by 1; otherwise, s will not envy s' .

We assume for the purposes of induction that s envies s' by at most one course by the time $L[i]$ ends. We now prove that s envies s' by at most one course by the time $L[i + 1]$ ends. By our induction hypothesis, there are two cases, when s envies s' by one course when $L[i]$ ends, and when s does not envy s' when $L[i]$ ends. In the latter case, it is only possible for s to envy s' by at most one course by the time $L[i + 1]$ ends since s' has gained at most one additional course which s desires by the time $L[i + 1]$ ends. Now we prove the former case. Let j be the next course (after the course $L[i]$) that the algorithm considers that is assigned to either student s or s' , is desired by s . Then, course j would fit into the current schedule of both s and s' , since j starts after the end time of $L[i]$. Suppose for the sake of contradiction that j is assigned to s' . Since we compare the set of courses that end no later than the end time of $L[i]$, if j is assigned to s' then j has start time later than $L[i]$. Student s envies s' by 1 course among the set of courses she received that end no later than $L[i]$. Then, course j is not assigned to s only if s has a conflicting course (since s has fewer courses than s'); however, this contradicts with j being the next course assigned after $L[i]$ to either s or s' . ◀

Now to prove Theorem 16, we use Lemma 17. Specifically, by the time the last course in L ends, student s envies s' by at most one course. Any course in the schedule of s that ends at a time later than this does not increase the envy s feels towards s' . And due to symmetry, s' similarly does not envy s by more than one course. Similarly, any course assigned to s in between the ending times of $L[i]$ and $L[i + 1]$ does not increase the envy of s . ◀

► **Theorem 18.** *Under binary preferences and uniform credits for all courses, Algorithm 3 produces an EF1-CC allocation.*

Proof. Theorem 16 stated that no student envies another student by more than one course. We are left to show that no student envies charity by more than 1 course. Assume for the sake of contradiction that there is a student s that envies the charity, this means that (1) $|A_s| < C_s$ where c_s is the credit cap for student s , and (2) there is a bigger independent set of courses (name this set I) among the courses assigned to the charity than the number of allocated courses to s , i.e. $|A_s| < |I|$.

First, all courses in I overlap with A_s because if some course $j \in I$ does not conflict with any course in A_s , then our algorithm would have assigned j to s . If we were to sort I and A_s by earliest finish time first and index them by i , observe that for all i , course $A_s[i]$ ends earlier than $I[i]$ due to our algorithm (this can be proven with a very elementary greedy stays ahead induction proof [28]). This means that $|A_s| \geq |I|$ because if there were to be a course $j = E[|A_s| + 1]$, that means j begins after the last course in A_s ends, which means our algorithm would have assigned j to s . ◀

4.3 Maximizing Max-Min Objective

Now, we look at a more general version of CAUTC-SC considering binary utilities and provide the following algorithm that gives a constant factor approximation when the maximum and minimum durations of any course are within a constant factor c of each other. We first describe our algorithm with the pseudocode provided in Algorithm 4. The algorithm proceeds as follows. The courses are sorted by end time (Line 2). Then, in the sorted order of courses, each course is given to a student who has non-zero preference for the course, has not filled up all of their credits (up to their credit cap), has no conflicting courses, and who has the least number of assigned courses among all students who have non-zero preference for the course (Line 6). Suppose we assign course j to a student i . Let d_i be a *dummy course* that we create for each student i . Then, we repeatedly perform the following procedure until no more *augmenting paths* exist (Line 9):

- For each course assigned to student i , draw a directed edge from course j' assigned to student $i' \neq i$ if j conflicts with j' and removing j' means that j does not conflict with any other course assigned to i' and i' has less assigned courses than i (Line 13).
- For each course assigned to student i , draw a directed edge from dummy course $d_{i'}$ to j if j does not conflict with any course assigned to i' and i' has less than or equal to the number of courses assigned to i (Line 14).
- Repeat with the courses assigned to i' and omit all courses assigned to student i from this part of the graph construction.

Once a full directed acyclic graph is drawn using the above procedure, we define an *augmenting path* to be a directed path with the source at a dummy course and sink at a course of i (Line 16). We repeatedly produce a new directed acyclic graph using the above procedure and switch courses between students via an augmenting path until no such augmenting paths remain (Line 18). Then, we proceed with assigning the next item in the sorted order of courses. We prove that our algorithm returns a constant factor approximation of the max-min objective value.

Algorithm 4 Max-Min Assignment of Courses.

Require: Courses \mathcal{M} , students \mathcal{N} , binary utilities U

Ensure: Approximate max-min allocation J

```

1: function FIND-MAX-MIN-ALLOCATION( $\mathcal{M}, \mathcal{N}, U$ )
2:   Sort courses in  $\mathcal{M}$  by end time from earliest to latest.
3:    $D \leftarrow \emptyset$ .
4:   Let  $Q \leftarrow \emptyset$  be a queue of students.
5:   for each course  $j$  in sorted order do
6:     Assign  $j$  to student  $i$  with minimum number of assigned courses, has not reached
       credit cap, where  $u_i(j) > 0$ , and does not have any conflicting courses.
7:     Add  $i$  to the end of  $Q$ .
8:     Set  $AugPath \leftarrow True$ .
9:     while  $AugPath$  do
10:      while  $Q \neq \emptyset$  do
11:        Remove the first student  $i'$  from  $Q$ .
12:        for each course  $j$  assigned to  $i'$  do
13:          Draw directed edge from  $j'$  assigned to student  $b$  to  $j$  if  $j'$  conflicts with  $j$ ,
            removing  $j'$  results in  $j$  conflicting with no course assigned to  $b$  conflicting with  $j$  and  $b$ 
            now has less assigned courses than  $i'$ , and  $b \notin D$ . Add  $b$  to the end of  $Q$ .
14:          Draw a directed edge from  $d_b$  to  $j$  if student  $b$  does not have any courses that
            conflict with  $j$  and  $b$  has at most as many courses as  $i'$ . Add  $b$  to the end of  $Q$ .
15:           $D \leftarrow D \cup i'$ .
16:          Find an augmenting path with source at a dummy course and sink at course
            assigned to  $i$  and reassign courses along augmenting path from sink to source.
17:          if there is no augmenting path then
18:             $AugPath \leftarrow False$ .
19:   return Allocation of courses to students.

```

► **Theorem 19.** *Algorithm 4 achieves a c -factor approximate solution for CAUTC-SC, where c is the maximum ratio between the durations of any two courses.*

Proof. Let S denote the set of students with the minimum number of assigned courses by our algorithm. We compare the allocations of courses assigned to each of the students in S by our algorithm with the allocation of courses assigned to the students by OPT. Let $i \in S$ be one such student. Let A_i be the set of courses allocated to student i by our algorithm and OPT_i be the set of courses allocated to i by OPT. There are four different types of courses assigned to these students that we are concerned with. Courses assigned to i in A_i and not in OPT_i can only make max-min greater; thus, we do not consider such courses. The same holds for courses assigned in A_i and by OPT to another student. Then, courses assigned by OPT but not assigned to A_i must conflict with at least one other course assigned to i . Hence, such courses can be charged to the course that it conflicts. The conflicting course(s) cannot be assigned in OPT_i ; thus, the course in OPT_i can be charged to one of the conflicting courses. The remaining type are courses that are in OPT_i , not in A_i , but are instead assigned to another student by our algorithm. Let j be one such course; then, either

- Course j is assigned to a student i' with *less* assigned courses than i . This scenario is impossible by definition of i as a student with the smallest number of assigned courses.
- Course j is assigned to a student i' with the same or more assigned courses than i . Student i must be assigned a conflicting course to j , as otherwise, when the last course

assigned to i' is assigned to i' , course j would have been transferred to i . Suppose first that i' has a greater number of courses than i and i has no conflicting course with j , then this is a contradiction since j would have been eventually transferred to i . Now suppose i has a course that conflicts j . If this conflicting course has an earlier end time than j , then j can be charged to the conflicting course. Furthermore, any course can conflict with at most c different courses assigned to i in OPT by our assumption of the ratio between the longest class and shortest class. Thus, we charge the course to the conflicting course assigned to A_i ; at most c such courses can be charged to any course in A_i . ◀

5 Experimental Results

In this section, we present a case study with data derived from MS students at Northwestern. We compare the performance of our algorithms Algorithm 3 and Algorithm 4 to those of optimal integer programs (IP) implemented using Gurobi [21] in Python. There are two integer programs of note: one to get the max-min value, and one to get the assignment maximizing the total social welfare given the max-min value T such that every student must receive at least T courses. We will henceforth refer to both of these integer programs that produce the optima as OPT. We implement Algorithm 3 and Algorithm 4 in Python [1]. In Algorithm 4, after looping through each course, exchange path operations are initiated. The graphs of exchange paths were implemented in NetworkX[22] in Python. The experiments are conducted on a Dell PowerEdge R740 with 2 x Intel Xeon Gold 6140 2.3GHz 18 core 36 threads processors, 192GB RAM, dual 10Gbps and 1Gbps NICs.

The dataset was obtained through a Google Form sent out to Master's students who wished to take computer science courses. They could select and rank up to five courses. Since ordinal preferences are beyond the scope of this paper, we only considered the courses they desire (binary valuations).

■ **Table 1** Comparison of utilities.

Datasets		max-min			total utility		
		OPT	Algorithm 3	Algorithm 4	OPT	Algorithm 3	Algorithm 4
real-world data	dataset	1	1	1	744	624	744
	alteration 1	2	1	2	725	623	725
	alteration 2	3	2	3	686	686	686
	alteration 3	2	2	2	760	760	760
synthetic data	example 1	2	2	2	7	7	6
	example 2	3	1	2	6	3	6
	example 3	4	3	3	8	6	6
	example 4	1	1	1	5	4	5
	example 5	1	1	1	6	5	6
	example 6	4	4	4	12	12	12
	example 7	4	4	4	12	12	12
	example 8	4	4	4	12	12	12

In terms of utility and max-min value, both algorithms incurred similar values as that of OPT. Table 1 compares the max-min value between OPT, Algorithm 3, and Algorithm 4. For almost all instances listed in Table 2, Algorithm 3 was much faster than OPT. Since our input data is not too large, we could compute an optimal assignment by solving the corresponding IP using Gurobi, which is not scalable in general.

■ **Table 2** Comparison of runtimes in milliseconds. All runtimes correspond to instances in the corresponding cells in Table 1. There is only one column under *total utility* because Algorithm 3 and Algorithm 4 are executed only once, as opposed to the two different linear programs of OPT.

Datasets		max-min			total utility
		OPT	Algorithm 3	Algorithm 4	OPT
real-world data	dataset	64.3614	44.893709	789.936638	1.883833
	alteration 1	59.0851	44.675743	751.317634	1.818061
	alteration 2	47.1846	38.609633	490.054614	1.447398
	alteration 3	59.0924	44.469586	1214.057334	1.625204
synthetic data	example 1	3.672	0.405452	1.505546	1.518124
	example 2	0.6444	0.177736	0.831938	0.186128
	example 3	2.3873	0.263971	1.145906	0.552925
	example 4	2.5193	0.200845	1.052790	0.524367
	example 5	3.3109	0.247982	1.344810	0.588912
	example 6	1.1906	0.242676	0.667803	0.232696
	example 7	11609.3298	594.707318	34986.057784	877.306024
	example 8	634809.8065	31554.700315	2278023.952813	9592.727642

The results of our experiments demonstrate the effectiveness of our algorithms. Algorithm 3 was able to give near-optimal solutions with a significantly reduced computational cost compared to integer programming, a traditional method. The reduced runtime is a testament to the effectiveness of the algorithms and their potential for practical implementation. The findings of this study highlight the potential for further improvement and optimization of these algorithms (especially Algorithm 4 since its runtime has much room for improvement) making them an attractive option for real-world applications. Although Algorithm 4 is slower than OPT for some of the tested instances, we believe it will be much faster and more scalable on instances larger than what we tested in our experiments.

6 Conclusions and Future Work

We investigated the problem of allocating conflicting resources across n agents, while taking into account both fairness as well as overall utility of the assignment. While resource allocation is extremely well studied, cases when the resources have conflicts have not been well studied from an algorithmic perspective.

Several generalizations of the course allocation problem open up interesting new directions for the fair allocation literature, such as generalizing utilities beyond additive binary and considering non-uniform credits for different courses. Further, each course may be a corresponding collection of time intervals (instead of a single interval). While we assume that our courses meet once a week, this may not be true for the general case where courses might meet on Tues-Thurs or Mon-Wed or Mon-Wed-Fri. If two courses overlap in any of the time windows then there is an edge in the conflict graph between them. However, such considerations would make the problem more challenging since the corresponding conflict graphs would be more complicated than interval graphs.

Going ahead, there are several directions for future research that can extend and improve upon our approach to course allocation. By addressing these challenges, we can develop more effective and fair algorithms for allocating courses to students, and better meet the diverse and evolving needs of students.

References

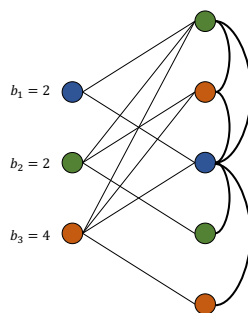
- 1 Fair course allocation implementations. <https://github.com/yiduo/CS-499-Khuller>.
- 2 Miklos Ajtai, James Aspnes, Moni Naor, Yuval Rabani, Leonard J Schulman, and Orli Waarts. Fairness in scheduling. *Journal of Algorithms*, 29(2):306–357, 1998.
- 3 Nikhil Bansal and Maxim Sviridenko. The santa claus problem. In *Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '06, pages 31–40, New York, NY, USA, 2006. Association for Computing Machinery. doi:10.1145/1132516.1132522.
- 4 Nawal Benabbou, Mithun Chakraborty, Ayumi Igarashi, and Yair Zick. Finding fair and efficient allocations for matroid rank valuations. *ACM Transactions on Economics and Computation*, 9(4):1–41, 2021.
- 5 Mohamed Bendraouche, Mourad Boudhar, and Ammar Oulamara. Scheduling: Agreement graph vs resource constraints. *European Journal of Operational Research*, 240(2):355–360, 2015.
- 6 Vittorio Bilò, Angelo Fanelli, Michele Flammini, Gianpiero Monaco, and Luca Moscardelli. The price of envy-freeness in machine scheduling. In *Mathematical Foundations of Computer Science 2014: 39th International Symposium, MFCS 2014, Budapest, Hungary, August 25-29, 2014. Proceedings, Part II 39*, pages 106–117. Springer, 2014.
- 7 Arpita Biswas and Siddharth Barman. Fair division under cardinality constraints. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 91–97, 2018.
- 8 Hans L Bodlaender and Fedor V Fomin. Equitable colorings of bounded treewidth graphs. *Theoretical Computer Science*, 349(1):22–30, 2005.
- 9 Hans L Bodlaender, Klaus Jansen, and Gerhard J Woeginger. Scheduling with incompatible jobs. *Discrete Applied Mathematics*, 55(3):219–232, 1994.
- 10 Flavia Bonomo, Sara Mattia, and Gianpaolo Oriolo. Bounded coloring of co-comparability graphs and the pickup and delivery tour combination problem. *Theoretical Computer Science*, 412(45):6261–6268, 2011.
- 11 Peter Brucker and L. Nordmann. The k -track assignment problem. *Computing*, 52(2):97–122, 1994. doi:10.1007/BF02238071.
- 12 Eric Budish. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy*, 119(6):1061–1103, 2011.
- 13 Yinhui Cai, Guangting Chen, Yong Chen, Randy Goebel, Guohui Lin, Longcheng Liu, and An Zhang. Approximation algorithms for two-machine flow-shop scheduling with a conflict graph. In *Computing and Combinatorics: 24th International Conference, COCOON 2018, Qing Dao, China, July 2-4, 2018, Proceedings 24*, pages 205–217. Springer, 2018.
- 14 Martin C. Carlisle and Errol L. Lloyd. On the k -coloring of intervals. *Discrete Applied Mathematics*, 59(3):225–235, 1995. doi:10.1016/0166-218X(95)80003-M.
- 15 Nina Chiarelli, Matjaž Krnc, Martin Milanič, Ulrich Pferschy, Nevena Pivač, and Joachim Schauer. Fair allocation of indivisible items with conflict graphs. *Algorithmica*, pages 1–31, 2022.
- 16 Sami Davies, Thomas Rothvoss, and Yihao Zhang. A tale of santa claus, hypergraphs and matroids. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2748–2757. SIAM, 2020.
- 17 Colin Fisher. *Resource allocation in the public sector: Values, priorities and markets in the management of public services*. Routledge, 2002.
- 18 Frédéric Gardi. Mutual exclusion scheduling with interval graphs or related classes. part ii. *Discrete applied mathematics*, 156(5):794–812, 2008.
- 19 M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences)*. W. H. Freeman, first edition edition, 1979. URL: <http://www.amazon.com/Computers-Intractability-NP-Completeness-Mathematical-Sciences/dp/0716710455>.

- 20 Guilherme C.M. Gomes and Vinicius F. dos Santos. Kernelization results for equitable coloring**this work was partially supported by capes, cnpq, and fapemig. *Procedia Computer Science*, 195:59–67, 2021. Proceedings of the XI Latin and American Algorithms, Graphs and Optimization Symposium. doi:10.1016/j.procs.2021.11.011.
- 21 Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023. URL: <https://www.gurobi.com>.
- 22 Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using NetworkX. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11–15, Pasadena, CA USA, 2008.
- 23 Pierre Hansen, Alain Hertz, and Julio Kuplinsky. Bounded vertex colorings of graphs. *Discrete Mathematics*, 111(1-3):305–312, 1993.
- 24 Halvard Hummel and Magnus Lie Hetland. Fair allocation of conflicting items. *Autonomous Agents and Multi-Agent Systems*, 36(1):8, 2022.
- 25 Sungjin Im and Benjamin Moseley. Fair scheduling via iterative quasi-uniform sampling. *SIAM Journal on Computing*, 49(3):658–680, 2020.
- 26 Klaus Jansen. The mutual exclusion scheduling problem for permutation and comparability graphs. *Information and Computation*, 180(2):71–81, 2003.
- 27 Henry A Kierstead, Alexandr V Kostochka, Marcelo Mydlarz, and Endre Szemerédi. A fast algorithm for equitable coloring. *Combinatorica*, 30(2):217–224, 2010.
- 28 Jon Kleinberg and Éva Tardos. *Algorithm Design*. Addison Wesley, 2006.
- 29 Daniel Kowalczyk and Roel Leus. An exact algorithm for parallel machine scheduling with conflicts. *Journal of Scheduling*, 20(4):355–372, 2017.
- 30 Bo Li, Minming Li, and Ruilong Zhang. Fair allocation with interval scheduling constraints, 2021. doi:10.48550/arXiv.2107.11648.
- 31 Richard J Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. On approximately fair allocations of indivisible goods. In *Proceedings of the 5th ACM Conference on Electronic Commerce*, pages 125–131, 2004.
- 32 Amin Mallek and Mourad Boudhar. Scheduling on uniform machines with a conflict graph: complexity and resolution. *International Transactions in Operational Research*, 2022.
- 33 Jakub Marecek and Andrew J Parkes. Semidefinite programming in timetabling and mutual-exclusion scheduling. *arXiv preprint*, 2019. doi:10.48550/arXiv.1904.03539.
- 34 Vignesh Viswanathan and Yair Zick. Yankee swap: a fast and simple fair allocation mechanism for matroid rank valuations. *arXiv preprint*, 2022. doi:10.48550/arXiv.2206.08495.

A Maximizing b -Matching with Conflicts

In this section, we justify our model of representing courses as interval graphs by showing that the general problem of assigning courses to students is NP-complete when given arbitrary numbers of time segments (or intervals) for each course. Namely, when each course can take place over any arbitrary number of time periods, then the conflict graph can be represented as any general graph. We now discuss the more general problem of assigning resources to agents where in our specific setting, courses can be modeled as resources and students as agents.

One way to view the problem of maximizing the utility of assigning resources to agents, where each agent is assigned a set of non-conflicting resources, is to realize that any agent’s allocation is an independent set in the conflict graph. Assigning resources to n agents then becomes a maximum graph coloring problem, where the resources have to be colored with one of n different colors so that no two adjacent resources have the same color, but we simply attempt to maximize the number of colored resources (nodes). If the conflict graph has no restrictions or structure, then even the simplest case becomes NP-hard as we show next.



■ **Figure 3** Example b -matching with allocations of resources indicated by the different colors.

A b -matching of any graph is a degree constrained subgraph, where the degree of any node in the subgraph cannot exceed $b(v)$, a specified value. Note that any allocation of goods to agents can be thought of as a b -matching where the edges encode the value of the good to that agent, and the degree constraints model the number of seats in a course (available copies of the good to be assigned to agents) and the degree constraint on the agent nodes corresponds to an upper bound as to how many resources they desire.

► **Definition 20** (*b -Matching with Conflicts (MBMWC)*). Given a bipartite graph $G = (L \cup R, E)$, a length $|L \cup R|$ vector \vec{b} of non-negative integers, and a set of pairs $(a, a') \in F$ denoting conflicts between nodes on the same side (i.e. either $a \in L$ and $a' \in L$, or $a \in R$ and $a' \in R$) such that no node v can be matched to a and a' at the same time, a feasible b -matching with conflicts is one where the conflicts are respected and no node p gets matched to more than $b(p)$ nodes on the other side. A maximum b -matching for MBMWC is a feasible matching of maximum weight.

Even if we simply want to maximize the overall weight of the b -matching (i.e. the sum of everyone's allocation), the problem is NP -hard. This can be shown by a simple reduction from independent set.

► **Definition 21** (*Maximum Independent Set (MIS)*). Given a graph $G = (V, E)$, set of vertices $V' \subseteq V$ is independent if and only if $\forall p, q \in V', (p, q) \notin E$, i.e. no pair of vertices in V' shares an edge. A maximum independent set of a graph is an independent set with maximum cardinality.

Given a graph G and an integer k , asking for the existence of an independent set of size at least k is an NP -complete problem. We prove the difficulty of our problem by a reduction from the Independent Set problem.

► **Theorem 22.** Given a bipartite graph $G = (L \cup R, E)$, a vector \vec{b} , and a set of pairs F denoting conflicts, finding a b -matching satisfying MBMWC is NP -hard.

Proof. Given an instance of maximum independent set problem, $G = (V, E)$, and an integer k , we construct an instance of MBMWC, $H = (L \cup R, E')$ where L consists of one node (agent) v and $R = G$. We then create edges from v to all vertices in R . Let $b(v) = k$ and let $b_u = 1$ for all $u \in R$.

If we have a solution to MBMWC in H of weight k , then the matched vertices in R give a maximum independent set in G of cardinality k . In addition, if the graph G does contain an independent set of size at least k then any subset of k nodes can be safely matched with v (and they form a conflict free set). ◀

B Proofs

B.1 Proof of Theorem 6

Proof. This proof is a reduction from 3-PARTITION [19].

► **Definition 23** (CAUTC-DECISION). *Consider our problem CAUTC-SW in Section 2.1, instead of the objective of maximizing it, the decision version of it is that given the extra parameter k , is there an allocation such that total student utility is k ?*

► **Definition 24** (3-PARTITION). *Given a multiset of numbers, can one partition the numbers into triplets such that the sum of each triplet is equal? More precisely, and with an additional restriction on each number. Given a multiset S of $3m$ positive integers where $\sum_{i \in S} x_i = mT$, and each integer $x_i \in S$ satisfies $T/4 < x_i < T/2$, does there exist a partition of S into m disjoint subsets S_1, S_2, \dots, S_m such that the sum of the x_i values in each set S_j add exactly to T ?*

Given an instance of 3-PARTITION, one can reduce it to an instance of CAUTC-DECISION where utilities are uniform and credit caps are uniform and course credit counts are arbitrary. Let there be m students s_1, s_2, \dots, s_m , each with credit cap T , and let each number $x_i \in S$ from 3-PARTITION represent a course of credit count x_i . No two courses overlap. Every student is interested in every course with uniform utilities. Let $k = mT$. If the solution to CAUTC-DECISION is yes, then the solution to 3-PARTITION is also yes. But first, we have to prove that if there is a solution to CAUTC-DECISION, then each student is allocated exactly three courses. Since the total student utility is $k = mT$ and each student has a credit cap of T , each student is allocated courses whose credits sum to exactly T . Each student must have at least three courses, because each course j has credit $c_j < T/2$. On the other hand, each student must have at most three courses, because each course j has credit $c_j > T/4$. CAUTC-DECISION is therefore NP-hard. ◀

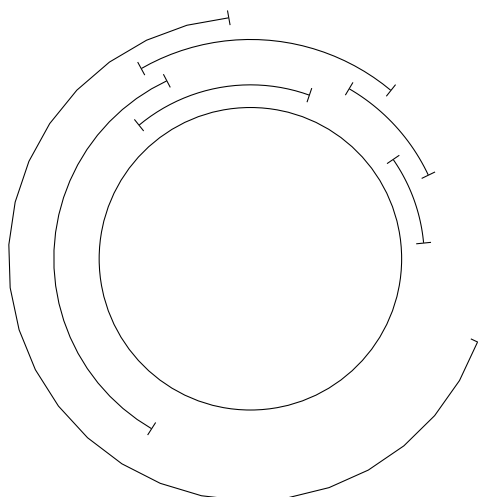
B.2 Proof of Theorem 7

Proof. This proof is based on the reduction from ARC COLORING to the k -track assignment problem by Brucker and Nordmann [11] showing NP-hardness of the k -track assignment problem.

► **Definition 25** (k -coloring problem for circular arc graphs (ARC COLORING)). *Given a positive integer k and a set F of n circular arcs A_1, A_2, \dots, A_n , where each A_i is an ordered pair (a_i, b_i) of positive integers where either $a_i < b_i$ or $b_i < a_i$, can F be partitioned into k disjoint subsets so that no two arcs in the same subset intersect?*

The following simple reduction from ARC COLORING shows that CAUTC is NP-hard: we cut the circle from the k -coloring problem for circular arc graphs at some arbitrary but fixed point t . Without loss of generality we calibrate that as $t = 0$, and the courses I_i have the form $I_i = [s_i, t_i]$, where each s_i and t_i is modulo L , the length of the circle.

Now assume that only the courses I_1, \dots, I_r contain the point $t = 0$ and that $r \leq k$, for if $r > k$, then the k -coloring problem has no solution. We define k students by making them have a utility of 1 only for the courses that overlap with the time interval $[t_j, s_j]$ for $j = 1, \dots, r$ and $[0, L]$ for $j = r + 1, \dots, k$. Now the problem of assigning the remaining courses I_{r+1}, \dots, I_n to these k students is equivalent to the k -coloring problem. ◀



■ **Figure 4** A circular arc model.

B.3 Proof of Theorem 8

Proof. We give a dynamic programming solution for two students, which is easily extendable to any constant k number of students. We sort the courses by non-decreasing start time and use this order to consider the courses in our DP. We define $N(j)$ to be the set of courses that overlap with course j . Given an instance of CAUTC with a constant number of students, for each course $j \in [m]$, course j is either assigned to student 1, to student 2, or to no one. The states of our DP are as follows. For each of the two students, we maintain a counter, p_1 and p_2 , respectively, for the remaining number of credits available to student 1 and 2; we also maintain the set of courses available to students 1 and 2 where t_1 and t_2 denote the earliest time that a course which starts at that time can be assigned to students 1 and 2, respectively. Finally, we maintain a counter j indicating the current course being iterated on.

Each time a course j is assigned to wlog student 1, we subtract the credit count of the course, c_j , from p_1 (the total credit count of the student course j is assigned to), increment t_1 by the duration of course j , that is we update t_1 to $t_1 + d_j$. We define our base case to be

$$OPT[p_1, p_2, t_1, t_2, m + 1] = 0 \quad (1)$$

for any valid p_1, p_2, t_1, t_2 and our initial state is

$$OPT[p_1, p_2, 0, 0, 0]. \quad (2)$$

We therefore have our recurrence scheme as follows:

$$\begin{aligned} OPT[p_1, p_2, t_1, t_2, j] = \max(&OPT[p_1, p_2, t_1, t_2, j + 1], \\ &\mathbb{1}(start_j \geq t_1 \cap c_j \leq p_1) \times (u_1(j) + OPT[p_1 - c_j, p_2, end_j, t_2, j + 1]), \\ &\mathbb{1}(start_j \geq t_2 \cap c_j \leq p_2) \times (u_2(j) + OPT[p_1, p_2 - c_j, t_1, end_j, j + 1])) \end{aligned} \quad (3)$$

We now prove the optimality of our solution via induction. In the base case, course $m + 1$ does not exist, hence, no utility is given for the base case. We now assume for our induction hypothesis that the state for the j -th job is an optimum assignment of courses to students for all valid values of p_1, p_2, t_1, t_2 . Now, we show that the optimum solution is computed for the $(j + 1)$ -st job. For the $(j + 1)$ -st course, it can either be given to student 1 or 2 or given to no one. Wlog suppose the $(j + 1)$ -st course is given to student 1. In this case, if

$start_{j+1} < t_1$ or $c_j > p_1$, then the returned value is 0 since course $j + 1$ cannot be assigned to student 1 in this case. Otherwise, we show that the states are correctly updated. When $j + 1$ is assigned to student 1, the amount of available credits is decreased for student 1 by c_{j+1} and t_1 is increased to end_{j+1} . Since the courses are sorted in non-decreasing order by start time, when course $j + 1$ is being considered, no course with start time earlier than $start_{j+1}$ is being considered. Thus, all courses $j' > j + 1$ have start time $\geq start_{j+1}$ and so will conflict with course $j + 1$ if and only if $start_{j+1} \leq start_{j'} < end_{j+1}$. Hence, setting t_1 to end_{j+1} precisely eliminates the courses $j' > j + 1$ that conflict with course $j + 1$. Since course $j + 1$ has been assigned to student 1, the utility $u_1(j + 1)$ is added. Finally, the counter is incremented to $j + 2$. The case for assigning $j + 1$ to student 2 is symmetric. When $j + 1$ is not given to either student, then no utility is added to the previous values and the counter is incremented to $j + 2$ with no other changes in the state. There are only three different cases for course $j + 1$: it is assigned to either student 1 or 2 or assigned to no one. Using the induction hypothesis and taking the maximum of the three options results in the maximum value for assigning course $j + 1$.

Now we prove the runtime of our DP algorithm. Since $c_j = O(1)$ for all $j \in [m]$, we can upper bound p_1 and p_2 by $O(m)$. We can bound t_1 and t_2 as follows. We only increment each of these counters to an end time of a course. There are at most m distinct end times and thus the total number of values t_1 and t_2 can take is m . Finally, the last counter is upper bounded by m . Hence, there are at most $O(m^5)$ different unique states for our DP and our algorithm takes $O(m^5)$ time. For $s = O(1)$ students, our algorithm would take $O(m^{2s+1})$ time. ◀

B.4 Proof of Theorem 9

Proof. We first prove the optimality of Algorithm 1. In this proof, we use the classical greedy-comes-first strategy. In the sorted order of courses by end time, let J be an optimum assignment of courses to students. We show that our greedy algorithm does not produce a worse assignment than J , thus proving its optimality. We prove this via induction on the k -th course in the order sorted by end time. We aim to show that for all $k \leq m$, the number of courses assigned by the greedy algorithm to each student up to course k is at least the number of courses with index $\leq k$ (in the sorted order) assigned in J to each student.

In the base case, when $k = 1$, no courses have been assigned yet, so either the first course is assigned to some student with a sufficiently large credit cap or no student has a sufficiently large credit cap in which case it also cannot be assigned in J . We assume for our induction hypothesis that our greedy algorithm has assigned at least as many courses up to and including the k -th course to each student as the number of courses in J with index $\leq k$ (in the sorted order by end time) assigned to each student. We now prove this for the $(k + 1)$ -st course. The trivial cases are when the $(k + 1)$ -st course is not in J or if the $(k + 1)$ -st course is assigned by the greedy algorithm. Let the $(k + 1)$ -st course be course j . If the course is in J and it is not assigned by the greedy algorithm to any student, then each student must satisfy at least one of the two following scenarios:

1. Student $i \in [n]$ has not enough remaining credits.
2. Student $i \in [n]$ is assigned a conflicting course.

If Item 1 is true, then student i is assigned as many courses by the greedy algorithm as they were assigned in J ; in other words, student i is assigned the maximum number of courses they can take; this means that the greedy algorithm returned a solution no worse than J , since every student has reached their credit cap (since all courses have the same number of credits), and there is no way to improve upon that.

Otherwise, if Item 1 is not true and Item 2 is true then we consider the course with the *latest* end time that is $\geq \text{end}_j$. Such a course must exist by our greedy algorithm since if no such conflicting course exists, then j would be assigned to i . Let this conflicting course be j' . Then, courses j and j' cannot both be assigned to student i in J . By our induction hypothesis, the greedy algorithm assigned at least as many courses to student i with index $\leq k$ as the number of courses assigned to i in J with index $\leq k$. Suppose wlog that j' is the only course assigned to i that conflicts with j and we remove course j' from student i 's assignment and instead assign j . Then, the number of courses assigned to i cannot increase. Now we argue that removing j' cannot allow another course to be assigned to i . Suppose there exists another course ℓ that is assigned in J and conflicts with j' and does not conflict with j (so that both ℓ and j can be assigned to i if j' is removed). Since all courses have the same duration, it must be the case that if ℓ exists then ℓ has start time earlier than j' and has end time earlier than the start time of j . In that case, j' could not have prevented ℓ from being assigned to i and there exists another course assigned by greedy to i that conflicts with ℓ . Hence, no such ℓ can exist and removing j' and adding j cannot lead to another course ℓ with start time earlier than start_j to be assigned to i . In other words, if ℓ had been assigned to i by greedy, then j would also have been chosen, which contradicts our initial assumption that j and ℓ conflict; and if ℓ hadn't been assigned to i by greedy, it's because a course that starts earlier than ℓ overlaps with it, in which case removing j' does not enable ℓ to be assigned to i .

Finally, courses j' and j cannot both be assigned to the same student in J . Thus, if j' is assigned to a student in J , then j is not assigned to that student. Hence, we only need to consider the case when j' is not assigned in J . By our argument above, at most one course in J is charged to a course assigned by our algorithm; hence, in this case, by what we proved above and by the induction hypothesis the number of courses assigned to i by the greedy algorithm with index $\leq k + 1$ is at least the number of courses assigned to i with index $\leq k + 1$ by the optimum solution J . ◀

Fair Correlation Clustering in Forests

Katrin Casel  

Humboldt-Universität zu Berlin, Germany

Tobias Friedrich  

Hasso Plattner Institute, Universität Potsdam, Germany

Martin Schirneck  

Faculty of Computer Science, Universität Wien, Austria

Simon Wietheger  

Hasso Plattner Institute, Universität Potsdam, Germany

Abstract

The study of algorithmic fairness received growing attention recently. This stems from the awareness that bias in the input data for machine learning systems may result in discriminatory outputs. For clustering tasks, one of the most central notions of fairness is the formalization by Chierichetti, Kumar, Lattanzi, and Vassilvitskii [NeurIPS 2017]. A clustering is said to be fair, if each cluster has the same distribution of manifestations of a sensitive attribute as the whole input set. This is motivated by various applications where the objects to be clustered have sensitive attributes that should not be over- or underrepresented. Most research on this version of fair clustering has focused on centroid-based objectives.

In contrast, we discuss the applicability of this fairness notion to CORRELATION CLUSTERING. The existing literature on the resulting FAIR CORRELATION CLUSTERING problem either presents approximation algorithms with poor approximation guarantees or severely limits the possible distributions of the sensitive attribute (often only two manifestations with a 1:1 ratio are considered). Our goal is to understand if there is hope for better results in between these two extremes. To this end, we consider restricted graph classes which allow us to characterize the distributions of sensitive attributes for which this form of fairness is tractable from a complexity point of view.

While existing work on FAIR CORRELATION CLUSTERING gives approximation algorithms, we focus on exact solutions and investigate whether there are efficiently solvable instances. The unfair version of CORRELATION CLUSTERING is trivial on forests, but adding fairness creates a surprisingly rich picture of complexities. We give an overview of the distributions and types of forests where FAIR CORRELATION CLUSTERING turns from tractable to intractable.

As the most surprising insight, we consider the fact that the cause of the hardness of FAIR CORRELATION CLUSTERING is not the strictness of the fairness condition. We lift most of our results to also hold for the relaxed version of the fairness condition. Instead, the source of hardness seems to be the distribution of the sensitive attribute. On the positive side, we identify some reasonable distributions that are indeed tractable. While this tractability is only shown for forests, it may open an avenue to design reasonable approximations for larger graph classes.

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis; Social and professional topics → Computing / technology policy; Theory of computation → Dynamic programming

Keywords and phrases correlation clustering, disparate impact, fair clustering, relaxed fairness

Digital Object Identifier 10.4230/LIPIcs.FORC.2023.9

Related Version *Full Version:* <https://arxiv.org/abs/2302.11295>



© Katrin Casel, Tobias Friedrich, Martin Schirneck, and Simon Wietheger;
licensed under Creative Commons License CC-BY 4.0

4th Symposium on Foundations of Responsible Computing (FORC 2023).

Editor: Kunal Talwar; Article No. 9; pp. 9:1–9:12



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

In the last decade, the notion of fairness in machine learning has increasingly attracted interest, see for example the review by Pessach and Schmueli [26]. Feldman, Friedler, Moeller, Scheidegger, and Venkatasubramanian [21] formalize fairness based on a US Supreme Court decision on disparate impact from 1971. It requires that sensitive attributes like gender or skin color should neither be explicitly considered in decision processes like hiring but also should the manifestations of sensitive attributes be proportionally distributed in all outcomes of the decision process. Feldman et al. formalize this notion for classification tasks. Chierichetti, Kumar, Lattanzi, and Vassilvitskii [15] adapt this concept for clustering tasks.

In this paper we employ the same disparate impact based understanding of fairness. Formally, the objects to be clustered have a color assigned to them that represents some sensitive attribute. Then, a clustering of these colored objects is called *fair* if for each cluster and each color the ratio of objects of that color in the cluster corresponds to the total ratio of vertices of that color. More precisely, a clustering is *fair*, if it partitions the set of objects into *fair subsets*.

► **Definition 1 (Fair Subset).** *Let U be a finite set of objects colored by a function $c: U \rightarrow [k]$ for some $k \in \mathbb{N}_{>0}$. Let $U_i = \{u \in U \mid c(u) = i\}$ be the set of objects of color i for all $i \in [k]$. Then, a set $S \subseteq U$ is fair if and only if for all colors $i \in [k]$ we have $\frac{|S \cap U_i|}{|S|} = \frac{|U_i|}{|U|}$.*

To understand how this notion of fairness affects clustering decisions, consider the following example. Imagine that an airport security wants to find clusters among the travelers to assign to each group a level of potential risk with corresponding anticipating measures. There are attributes like skin color that should not influence the assignment to a risk level. A bias in the data, however, may lead to some colors being over- or underrepresented in some clusters. Simply removing the skin color attribute from the data may not suffice as it may correlate with other attributes. Such problems are especially likely if one of the skin colors is far less represented in the data than others. A fair clustering finds the optimum clustering such that for each risk level the distribution of skin colors is fair, by requiring the distribution of each cluster to roughly match the distribution of skin colors among all travelers.

The seminal fair clustering paper by Chierichetti et al. [15] introduced this notion of fairness for clustering and studied it for the objectives k -center and k -median. Their work was extended by Bera, Chakrabarty, Flores, and Negahbani [9], who relax the fairness constraint in the sense of requiring upper and lower bounds on the representation of a color in each cluster. More precisely, they define the following generalization of fair sets.

► **Definition 2 (Relaxed Fair Set).** *For a finite set U and coloring $c: U \rightarrow [k]$ for some $k \in \mathbb{N}_{>0}$ let $p_i, q_i \in \mathbb{Q}$ with $0 < p_i \leq \frac{|U_i|}{|U|} \leq q_i < 1$ for all $i \in [k]$, where $U_i = \{u \in U \mid c(u) = i\}$. A set $S \subseteq U$ is relaxed fair with respect to q_i and p_i if and only if $p_i \leq \frac{|S \cap U_i|}{|S|} \leq q_i$ for all $i \in [k]$.*

Following these results, this notion of (relaxed) fairness was extensively studied for centroid-based clustering objectives with many positive results.

For example, Bercea et al. [10] give bicriteria constant-factor approximations for facility location type problems like k -center and k -median. Bandyapadhyay, Fomin and Simonov [6] use the technique of fair coresets introduced by Schmidt, Schwiegelshohn, and Sohler [28] to give constant factor approximations for many centroid-based clustering objectives; among many other results, they give a polynomial-time approximation scheme (PTAS) for fair k -means and k -median in Euclidean space. Fairness for centroid-based objectives seems to be so well understood, that most research already considers more generalized settings, like streaming [28], or imperfect knowledge of group membership [20].

In comparison, there are few (positive) results for this fairness notion applied to graph clustering objectives. The most studied with respect to fairness among those is CORRELATION CLUSTERING, arguably the most studied graph clustering objective. For CORRELATION CLUSTERING we are given a pairwise similarity measure for a set of objects and the aim is to find a clustering that minimizes the number of similar objects placed in separate clusters and the number of dissimilar objects placed in the same cluster. Formally, the input to CORRELATION CLUSTERING is a graph $G = (V, E)$, and the goal is to find a partition \mathcal{P} of V that minimizes the CORRELATION CLUSTERING cost defined as

$$\text{cost}(G, \mathcal{P}) = |\{\{u, v\} \in \binom{V}{2} \setminus E \mid \mathcal{P}[u] = \mathcal{P}[v]\}| + |\{\{u, v\} \in E \mid \mathcal{P}[u] \neq \mathcal{P}[v]\}|. \quad (1)$$

FAIR CORRELATION CLUSTERING then is the task to find a partition into *fair* sets that minimizes the CORRELATION CLUSTERING cost. We emphasize that this is the complete, unweighted, min-disagree form of CORRELATION CLUSTERING. (It is often called *complete* because every pair of objects is either similar or dissimilar but none is indifferent regarding the clustering. It is unweighted as the (dis)similarity between two vertices is binary. A pair of similar objects that are placed in separate clusters as well as a pair of dissimilar objects in the same cluster is called a *disagreement*, hence the naming of the min-disagree form.)

There are two papers that appear to have started studying FAIR CORRELATION CLUSTERING independently¹. Ahmadian, Epasto, Kumar, and Mahdian [2] analyze settings where the fairness constraint is given by some α and require that the ratio of each color in each cluster is at most α . For $\alpha = \frac{1}{2}$, which corresponds to our fairness definition if there are two colors in a ratio of 1 : 1, they obtain a 256-approximation. For $\alpha = \frac{1}{k}$, where k is the number of colors in the graph, they give a $16.48k^2$ -approximation. We note that all their variants are only equivalent to our fairness notion if there are α^{-1} colors that all occur equally often. Ahmadi, Galhotra, Saha, and Schwartz [1] give an $O(c^2)$ -approximation algorithm for instances with two colors in a ratio of 1 : c . In the special case of a color ratio of 1 : 1, they obtain a $3\beta + 4$ -approximation, given any β -approximation to unfair CORRELATION CLUSTERING. With a more general color distribution, their approach also worsens drastically. For instances with k colors in a ratio of 1 : c_2 : c_3 : \dots : c_k for positive integers c_i , they give an $O(k^2 \cdot \max_{2 \leq i \leq k} c_i)$ -approximation for the strict, and an $O(k^2 \cdot \max_{2 \leq i \leq k} q_i)$ -approximation for the relaxed setting².

Following these two papers, Friggstad and Mousavi [23] provide an approximation to the 1 : 1 color ratio case with a factor of 6.18. To the best of our knowledge, the most recent publication on FAIR CORRELATION CLUSTERING is by Ahmadian and Negahbani [3] who give approximations for FAIR CORRELATION CLUSTERING with a slightly different way of relaxing fairness. They give an approximation with ratio $\mathcal{O}(\varepsilon^{-1} k \max_{2 \leq i \leq k} c_i)$ for color distribution 1 : c_2 : c_3 : \dots : c_k , where ε relates to the amount of relaxation (roughly $q_i = (1 + \varepsilon)c_i$ for our definition of relaxed fairness).

All these results for FAIR CORRELATION CLUSTERING seem to converge towards considering the very restricted setting of two colors in a ratio of 1 : 1 in order to give some decent approximation ratio. In this paper, we want to understand if this is unavoidable, or if there is hope to find better results for other (possibly more realistic) color distributions. In order to isolate the role of fairness, we consider “easy” instances for CORRELATION CLUSTERING, and study the increase in complexity when adding fairness constraints. CORRELATION

¹ Confusingly, they both carry the title *Fair Correlation Clustering*.

² Their theorem states they achieve an $O(\max_{2 \leq i \leq k} q_i)$ -approximation but when looking at the proof it seems they have accidentally forgotten the k^2 factor.

CLUSTERING without the fairness constraint is easily solved on forests. We find that FAIR CORRELATION CLUSTERING restricted to forests turns NP-hard very quickly, even when additionally assuming constant degree or diameter. Most surprisingly, this hardness essentially also holds for relaxed fairness, showing that the hardness of the problem is not due to the strictness of the fairness definition.

On the positive side, we identify color distributions that allow for efficient algorithms. Not surprisingly, this includes ratio $1 : 1$, and extends to a constant number of k colors with distribution $c_1 : c_2 : c_3 : \dots : c_k$ for constants c_1, \dots, c_k . Such distributions can be used to model sensitive attributes with a limited number of manifestation that are almost evenly distributed. Less expected, we also find tractability for, in a sense, the other extreme. We show that FAIR CORRELATION CLUSTERING on forests can be solved in polynomial time for two colors with ratio $1 : c$ with c being very large (linear in the number of overall vertices). Such a distribution can be used to model a scenario where a minority is drastically underrepresented and thus in dire need of fairness constraints. Although our results only hold for forests, we believe that they can offer a starting point for more general graph classes. We especially hope that our work sparks interest in the so far neglected distribution of ratio $1 : c$ with c being very large.

Related Work

The study of clustering objectives similar or identical to CORRELATION CLUSTERING dates back to the 1960s [8, 27, 31]. Bansal, Blum, and Chawla [7] were the first to coin the term CORRELATION CLUSTERING as a clustering objective. We note that it is also studied under the name CLUSTER EDITING. The most general formulation of CORRELATION CLUSTERING regarding weights considers two positive real values for each pair of vertices, the first to be added to the cost if the objects are placed in the same cluster and the second to be added if the objects are placed in separate clusters [4]. The recent book by Bonchi, García-Soriano, and Gullo [11] gives a broad overview of the current research on CORRELATION CLUSTERING.

We focus on the particular variant that considers a complete graph with $\{-1, 1\}$ edge-weights, and the min disagreement objective function. This version is APX-hard [13], implying in particular that there is no algorithm giving an arbitrarily good approximation unless $P = NP$. The best known approximation for CORRELATION CLUSTERING is the very recent breakthrough by Cohen-Addad, Lee and Newman [16] who give a ratio of $(1.994 + \epsilon)$.

We show that in forests, all clusters of an optimal CORRELATION CLUSTERING solution have a fixed size. In such a case, CORRELATION CLUSTERING is related to k -BALANCED PARTITIONING. There, the task is to partition the graph into k clusters of equal size while minimizing the number of edges that are cut by the partition. Feldmann and Foschini [22] study this problem on trees and their results have interesting parallels with ours.

Aside from the results on FAIR CORRELATION CLUSTERING already discussed above, we are only aware of three papers that consider a fairness notion close to the one of Chierichetti et al. [15] for a graph clustering objective. Schwartz and Zats [29] consider incomplete FAIR CORRELATION CLUSTERING with the max-agree objective function. Dinitz, Srinivasan, Tsepenekas, and Vullikanti [18] study FAIR DISASTER CONTAINMENT, a graph cut problem involving fairness. Their problem is not directly a fair clustering problem since they only require one part of their partition (the saved part) to be fair. Ziko, Yuan, Granger, and Ayed [32] give a heuristic approach for fair clustering in general that however does not allow for theoretical guarantees on the quality of the solution.



■ **Figure 1** Example forest where a cluster of size 4 and two clusters of size 2 incur the same cost. With one cluster of size 4 (left), the inter-cluster cost is 0 and the intra-cluster cost is 4. With two clusters of size 2 (right), both the inter-cluster and intra-cluster cost are 2.

2 Contribution

We now outline our findings on FAIR CORRELATION CLUSTERING. We start by giving several structural results that underpin our further investigations. Afterwards, we present our algorithms and hardness results for certain graph classes and color ratios. We further show that the hardness of fair clustering does *not* stem from the requirement of the clusters exactly reproducing the color distribution of the whole graph. This section is concluded by a discussion of possible directions for further research.

2.1 Structural Insights

We outline here the structural insights that form the foundation of all our results. We first give a close connection between the cost of a clustering, the number of edges “cut” by a clustering, and the total number of edges in the graph. We refer to this number of “cut” edges as the *inter-cluster* cost as opposed to the number of non-edges inside clusters, which we call the *intra-cluster* cost. Formally, the intra- and inter-cluster cost are the first and second summand of the CORRELATION CLUSTERING cost in Equation (1), respectively. The following lemma shows that minimizing the inter-cluster cost suffices to minimize the total cost if all clusters are of the same size. This significantly simplifies the algorithm development for CORRELATION CLUSTERING.

► **Lemma 3.** *Let \mathcal{P} be a partition of the vertices of an m -edge graph G . Let χ denote the inter-cluster cost incurred by \mathcal{P} on G . If all sets in the partition are of size d , then $\text{cost}(\mathcal{P}) = \frac{(d-1)}{2}n - m + 2\chi$. In particular, if G is a tree, $\text{cost}(\mathcal{P}) = \frac{(d-3)}{2}n + 2\chi + 1$.*

The condition that all clusters need to be of the same size seems rather restrictive at first. However, we prove in the following that in bipartite graphs and, in particular, in forests and trees there is always a minimum-cost fair clustering such that indeed all clusters are equally large. This property stems from how the fairness constraint acts on the distribution of colors and is therefore specific to FAIR CORRELATION CLUSTERING. It allows us to fully utilize Lemma 3 both for building reductions in NP-hardness proofs as well as for algorithmic approaches as we can restrict our attention to partitions with equal cluster sizes.

Consider two colors of ratio 1 : 2, then any fair cluster must contain at least 1 vertex of the first color and 2 vertices of the second color to fulfil the fairness requirement. We show that a minimum-cost clustering of a forest, due to the small number of edges, consists entirely of such minimal clusters. Every clustering with larger clusters incurs a higher cost.

► **Lemma 4.** *Let F be a forest with $k \geq 2$ colors in a ratio of $c_1 : c_2 : \dots : c_k$ with $c_i \in \mathbb{N}_{>0}$ for all $i \in [k]$, $\gcd(c_1, c_2, \dots, c_k) = 1$, and $\sum_{i=1}^k c_i \geq 3$. Then, all clusters of every minimum-cost fair clustering are of size $d = \sum_{i=1}^k c_i$.*

■ **Table 1** Running times of our algorithms for FAIR CORRELATION CLUSTERING on forests depending on the color ratio. Value p is any rational such that $n/p - 1$ is integral; c_1, c_2, \dots, c_k are coprime positive integers, possibly depending on n . Functions f and g are given in the full version.

Color Ratio	1 : 1	1 : 2	1 : ($n/p - 1$)	$c_1 : c_2 : \dots : c_k$
Running Time	$O(n)$	$O(n^6)$	$O(n^{f(p)})$	$O(n^{g(c_1, \dots, c_k)})$

Lemma 4 does not extend to two colors in a ratio of 1 : 1 as illustrated in Figure 1. This color distribution is the only case for forests where a partition with larger clusters can have the same (but no smaller) cost. We prove a slightly weaker statement than Lemma 4, namely, that *there is* always a minimum-cost fair clustering with minimal clusters. This property, in turn, holds not only for forests but for every bipartite graph. Note that in general bipartite graphs there are more color ratios than only 1 : 1 that allow for these ambiguities.

► **Lemma 5.** *Let $G = (A \cup B, E)$ be a bipartite graph with $k \geq 2$ colors in a ratio of $c_1 : c_2 : \dots : c_k$ with $c_i \in \mathbb{N}_{>0}$ for all $i \in [k]$ and $\gcd(c_1, c_2, \dots, c_k) = 1$. Then, there is a minimum-cost fair clustering such that all its clusters are of size $d = \sum_{i=1}^k c_i$. Further, each minimum-cost fair clustering with larger clusters can be transformed into a minimum-cost fair clustering such that all clusters contain no more than d vertices in linear time.*

In summary, the results above show that the ratio of the color classes is the key parameter determining the cluster size. If the input is a bipartite graph whose vertices are colored with k colors in a ratio of $c_1 : c_2 : \dots : c_k$, our results imply that without losing optimality, solutions can be restricted to contain only clusters of size $d = \sum_{i=1}^k c_i$, each with exactly c_i vertices of color i . Starting from these observations, we show in this work that the color ratio is also the key parameter determining the complexity of FAIR CORRELATION CLUSTERING. On the one hand, the simple structure of optimal solutions restricts the search space and enables polynomial-time algorithms, at least for some instances. Additionally, due to the fixed cluster size d , returning *any* fair clustering in a forest can only cause so many mistakes. In fact, this procedure yields an approximation factor decreasing in d and converging to 1 as $d \rightarrow \infty$. Combining this with the fact that FAIR CORRELATION CLUSTERING can be solved in time increasing in d , see Table 1, allows for a PTAS in forests. On the other hand, these insights allow us to show hardness already for very restricted input classes. The technical part of most of the proofs consists of exploiting the connection between the clustering cost, total number of edges, and the number of edges cut by a clustering.

2.2 Tractable Instances

We start by discussing the algorithmic results. The simplest case is that of two colors, each one occurring equally often. We prove that for bipartite graphs with a color ratio 1 : 1 FAIR CORRELATION CLUSTERING is equivalent to the maximum bipartite matching problem, namely, between the vertices of different color. Via the standard reduction to computing maximum flows, this allows us to benefit from the recent breakthrough by Chen, Kyng, Liu, Peng, Probst Gutenberg, and Sachdeva [14]. It gives an algorithm running in time $m^{1+o(1)}$.

The remaining results focus on forests as the input, see Table 1. It should not come as a surprise that our main algorithmic paradigm is dynamic programming. A textbook version finds a maximum matching in linear time in a forests, solving the 1 : 1 case. For general color ratios, we devise much more intricate dynamic programs. We use the color ratio 1 : 2

■ **Table 2** Complexity of FAIR CORRELATION CLUSTERING on trees and general graphs depending on the diameter. The value c is a positive integer, possibly depending on n .

Diameter	Color Ratio	Trees	General Graphs
2, 3	any	$O(n)$	NP-hard
≥ 4	$1 : c$	NP-hard	NP-hard

as an introductory example. The algorithm has two phases. In the first, we compute a list of candidate *splittings* that partition the forest into connected parts containing at most 1 blue and 2 red vertices each. In the second phase, we assemble the parts of each of the splittings to fair clusters and return the cheapest resulting clustering. The difficulty lies in the two phases not being independent from each other. It is not enough to minimize the “cut” edges in the two phases separately. We prove that the costs incurred by the merging additionally depends on the number of parts of a certain type generated in the splittings. Tracking this along with the number of cuts results in a $O(n^6)$ -time algorithm. Note that we did not optimize the running time as long as it is polynomial.

We generalize this to k colors in a ratio $c_1 : c_2 : \dots : c_k$.³ We now have to consider *all* possible colorings of a partition of the vertices such that in each part the i -th color occurs at most c_i times. While assembling the parts, we have to take care that the merged colorings remain compatible. The resulting running time is $O(n^{g(c_1, \dots, c_k)})$ for some (explicit) polynomial g . Recall that, by Lemma 4, the minimum cluster size is $d = \sum_{i=1}^k c_i$. If this is a constant, then the dynamic program runs in polynomial time. If, however, the number of colors k or some color’s proportion grows with n , it becomes intractable. Equivalently, the running time gets worse if there are very large but sublinearly many clusters.

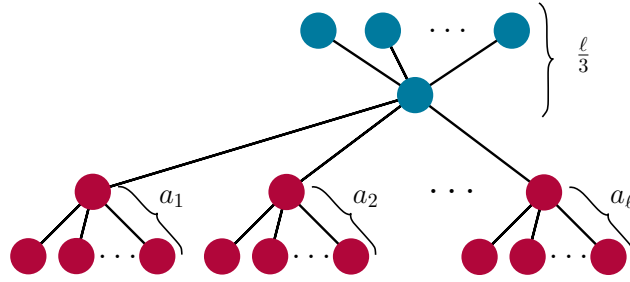
To mitigate this effect, we give a complementary algorithm at least for forests with two colors. Namely, consider the color ratio $1 : \frac{n}{p} - 1$. Then, an optimal solution has p clusters each of size $d = n/p$. The key observation is that the forest contains p vertices of the color with fewer occurrences, say, blue, and any fair clustering isolates the blue vertices. This can be done by cutting at most $p - 1$ edges and results in a collection of (sub-)trees where each one has at most one blue vertex. To obtain the clustering, we split the trees with red excess vertices and distribute those among the remaining parts. We track the costs of all the $O(n^{\text{poly}(p)})$ many cut-sets and rearrangements to compute the one of minimum cost. In total, the algorithm runs in time $O(n^{f(p)})$ for some polynomial in p . In summary, we find that if the number of clusters p is constant, then the running time is polynomial. Considering in particular an integral color ratio $1 : c$,⁴ we find tractability for forests if $c = O(1)$ or $c = \Omega(n)$. We will show next that FAIR CORRELATION CLUSTERING with this kind of a color ratio is NP-hard already on trees, hence the hardness must emerge somewhere for intermediate c .

2.3 A Dichotomy for Bounded Diameter

Table 2 shows the complexity of FAIR CORRELATION CLUSTERING on graphs with bounded diameter. We obtain a dichotomy for trees with two colors with ratio $1 : c$. If the diameter is at most 3, an optimal clustering is computable in $O(n)$ time, but for diameter at least 4, the problem becomes NP-hard. In fact, the linear-time algorithm extends to trees with an arbitrary number of colors in any ratio.

³ The c_i are coprime, but they are not necessarily constants with respect to n .

⁴ In a color ratio $1 : c$, c is not necessarily a constant, but ratios like $2 : 5$ are not covered.



■ **Figure 2** The tree with diameter 4 in the reduction from 3-PARTITION to FAIR CORRELATION CLUSTERING.

The main result in that direction is the hardness of FAIR CORRELATION CLUSTERING already on trees with diameter at least 4 and two colors of ratio $1 : c$. This is proven by a reduction from the strongly NP-hard 3-PARTITION problem. There, we are given positive integers a_1, \dots, a_ℓ where ℓ is a multiple of 3 and there exists some B with $\sum_{i=1}^{\ell} a_i = B \cdot \frac{\ell}{3}$. The task is to partition the numbers a_i into triples such that each one of those sums to B . The problem remains NP-hard if all the a_i are strictly between $B/4$ and $B/2$, ensuring that, if some subset of the numbers sums to B , it contains exactly three elements.

We model this problem as an instance of FAIR CORRELATION CLUSTERING as illustrated in Figure 2. We build ℓ stars, where the i -th one consists of a_i red vertices, and a single star of $\ell/3$ blue vertices. The centers of the blue star and all the red stars are connected. The color ratio in the resulting instance is $1 : B$. Lemma 4 then implies that there is a minimum-costs clustering with $\ell/3$ clusters, each with a single blue vertex and B red ones. We then apply Lemma 3 to show that this cost is below a certain threshold if and only if each cluster consist of exactly three red stars (and an arbitrary blue vertex), solving 3-PARTITION.

2.4 Maximum Degree

The reduction above results in a tree with a low diameter but arbitrarily high maximum degree. We have to adapt our reductions to show hardness also for bounded degrees. The results are summarized in Table 3. If the FAIR CORRELATION CLUSTERING instance is not required to be connected, we can represent 3-PARTITION with a forest of trees with maximum degree 2, that is, a forest of paths. The input numbers are modeled by paths with a_i vertices. The forest also contains $\ell/3$ isolated blue vertices, which again implies that an optimal fair clustering must have $\ell/3$ clusters each with B red vertices. By defining a sufficiently small cost threshold, we ensure that the fair clustering has cost below it if and only if none of the path-edges are “cut” by the clustering, corresponding to a partition of the a_i .

There is nothing special about paths, we can arbitrarily restrict the shape of the trees, as long it is possible to form such a tree with any given number of vertices. However, the argument crucially relies on the absence of edges between the a_i -trees and does not transfer to connected graphs. This is due to the close relation between inter-cluster costs and the number of edges, see Lemma 3. The complexity of FAIR CORRELATION CLUSTERING on a single path with a color ratio $1 : c$ remains open. Notwithstanding, we show hardness for trees in two closely related settings: keeping the ratio $1 : c$ but raising the maximum degree to 5, or having a single path with $n/2$ colors where each color is shared by exactly 2 vertices.

For the case of maximum degree 5 and two colors with ratio $1 : c$, we can again build on the 3-PARTITION machinery. The construction is inspired by how Feldmann and Foschini [22] used the problem to show hardness of computing so-called k -balanced partitions. We adapt it to our setting in which the vertices are colored and the clusters need to be fair.

■ **Table 3** Hardness of FAIR CORRELATION CLUSTERING on trees and forests depending on the maximum degree. The value c is a positive integer, possibly depending on n . The complexity for paths (trees with maximum degree 2) with color ratio $1 : c$ is open.

Max. Degree	Color Ratio	Trees	Forests
2	$1 : c$		NP-hard
≥ 2	$n/2$ colors, 2 vertices each	NP-hard	NP-hard
≥ 5	$1 : c$	NP-hard	NP-hard

For the single path with $n/2$ colors, we reduce from (the 1-regular 2-colored variant of) the PAINT SHOP PROBLEM FOR WORDS [19]. There, a word is given in which every symbol appears exactly twice. The task is to assign the values 0 and 1 to the letters⁵ such that, for each symbol, exactly one occurrence receives a 1, and the number of blocks of consecutive 0s or 1s is minimized. In the translation to FAIR CORRELATION CLUSTERING, we represent the word as a path and the symbols as colors. To remain fair, there must be two clusters containing exactly one vertex of each color, translating back to a 0/1-assignment to the word.

2.5 Relaxed Fairness

One could think that the hardness of FAIR CORRELATION CLUSTERING already for classes of trees and forests has its origin in the strict fairness condition. After all, the color ratio in each cluster must precisely mirror that of the whole graph. This impression is deceptive. Instead, we lift most of our hardness results to RELAXED FAIR CORRELATION CLUSTERING considering the *relaxed fairness* of Bera et al. [9]. Recall Definition 2. It prescribes two rationals p_i and q_i for each color i and allows, the proportion of i -colored elements in any cluster to be in the interval $[p_i, q_i]$, instead of being precisely c_i/d , where $d = \sum_{j=1}^k c_j$.

The main conceptual idea is that, in some settings, the *minimum-cost* solution under a relaxed fairness constraint is *exactly* fair. We show this for the cases in which we reduce from 3-PARTITION. In particular, RELAXED FAIR CORRELATION CLUSTERING with a color ratio of $1 : c$ is NP-hard on trees with diameter 4 and forests of paths, respectively. Furthermore, the transferal of hardness is immediate for the case of a single path with $n/2$ colors and exactly 2 vertices of each color. Any relaxation of fairness still requires one vertex of each color in every cluster, keeping the equivalence to the PAINT SHOP PROBLEM FOR WORDS.

In contrast, algorithmic results are more difficult to extend if there are relaxedly fair solutions that have lower cost than any exactly fair one. We then no longer know the cardinality of the clusters in an optimal solution. As a proof of concept, we show that a slight adaption of our dynamic program for two colors in a ratio of $1 : 1$ still works for what we call α -relaxed fairness.⁶ There, the lower fairness ratio is $p_i = \alpha \cdot \frac{c_i}{d}$ and the upper one is $q_i = \frac{1}{\alpha} \cdot \frac{c_i}{d}$ for some parameter $\alpha \in (0, 1)$. We give an upper bound on the necessary cluster size depending on α , which is enough to find a good splitting of the forest. Naturally, the running time now also depends on α , but is of the form $O(n^{h(1/\alpha)})$ for some polynomial h . In particular, we get a polynomial-time algorithm for constant α . The proof of correctness consists of an exhaustive case distinction already for the simple case of $1 : 1$. We are confident that this can be extended to more general color ratios, but did not attempt it in this work.

⁵ The original formulation [19] assigns colors, aligning better with the paint shop analogy. We change the exposition here in order to avoid confusion with the colors in the fairness sense.

⁶ This should not be confused with the notion of α -fairness in resource allocation [24, 25].

2.6 Summary and Outlook

We show that FAIR CORRELATION CLUSTERING on trees, and thereby forests, is NP-hard. It remains so on trees of constant degree or diameter, and—for certain color distributions—it is also NP-hard on paths. On the other hand, we give a polynomial-time algorithm if the minimum size d of a fair cluster is constant. We also provide an efficient algorithm for the color ratio $1 : c$ if the total number of clusters is constant, corresponding to $c \in \Theta(n)$. For our main algorithms and hardness results, we prove that they still hold when the fairness constraint is relaxed, so the hardness is not due to the strict fairness definition. Ultimately, we hope that the insights gained from these proofs as well as our proposed algorithms prove helpful to the future development of algorithms to solve FAIR CORRELATION CLUSTERING on more general graphs. In particular, fairness with color ratio $1 : c$ with c being very large seems to be an interesting and potentially tractable type of distribution for future study.

As first steps to generalize our results, we give a polynomial-time approximation scheme (PTAS) for FAIR CORRELATION CLUSTERING on forests. This further motivates to study approximation algorithms on more general classes of graphs. Another avenue for future research could be that Lemma 5, bounding the cluster size of optimal solutions, extends also to bipartite graphs. This may prove helpful in developing exact algorithms for bipartite graphs with other color ratios than $1 : 1$. Regarding further graph classes, we suspect that tractability will first have to be examined for the standard (unfair) CORRELATION CLUSTERING before considering additional fairness constraints.

Parameterized algorithms are yet another approach to solving more general instances. When looking at the decision version of FAIR CORRELATION CLUSTERING, our results can be cast as an XP-algorithm when the problem is parameterized by the cluster size d , for it can be solved in time $O(n^{g(d)})$ for some function g . Similarly, we get an XP-algorithm for the number of clusters as parameter. We wonder whether FAIR CORRELATION CLUSTERING can be placed in the class FPT of fixed-parameter tractable problems for any interesting structural parameters. This would require a running time of, e.g., $g(d) \cdot \text{poly}(n)$. There are FPT-algorithms for CLUSTER EDITING parameterized by the cost of the solution [12]. Possibly, future research might provide similar results for the fair variant as well. A natural extension of our dynamic programming approach could potentially lead to an algorithm parameterizing by the treewidth of the input graph. Such a solution would be surprising, however, since to the best of our knowledge even for normal, unfair CORRELATION CLUSTERING⁷ and for the related MAX DENSE GRAPH PARTITION [17] no treewidth approaches are known.

Finally, it is interesting how FAIR CORRELATION CLUSTERING behaves on paths. While we obtain NP-hardness for a particular color distribution from the PAINT SHOP PROBLEM FOR WORDS, the question of whether FAIR CORRELATION CLUSTERING on paths with for example two colors in a ratio of $1 : c$ is efficiently solvable or not is left open. However, we believe that this question is rather answered by the study of the related (discrete) NECKLACE SPLITTING problem, see the work of Alon and West [5]. There, the desired cardinality of every color class is explicitly given, and it is non-constructively shown that there always exists a split of the necklace with the number of cuts meeting the obvious lower bound. A constructive splitting procedure may yield some insights for FAIR CORRELATION CLUSTERING on paths.

⁷ In more detail, no algorithm for complete CORRELATION CLUSTERING has been proposed. Xin [30] gives a treewidth algorithm for *incomplete* CORRELATION CLUSTERING for the treewidth of the graph of all positively and negatively labeled edges.

References

- 1 Saba Ahmadi, Sainyam Galhotra, Barna Saha, and Roy Schwartz. Fair correlation clustering. *CoRR*, arXiv:2002.03508, 2020. ArXiv preprint. doi:10.48550/arXiv.2002.03508.
- 2 Sara Ahmadian, Alessandro Epasto, Ravi Kumar, and Mohammad Mahdian. Fair correlation clustering. In *Proceedings of the 23rd Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 4195–4205, 2020. URL: <https://proceedings.mlr.press/v108/ahmadian20a.html>.
- 3 Sara Ahmadian and Maryam Negahbani. Improved approximation for fair correlation clustering. *CoRR*, abs/2206.05050, 2022. doi:10.48550/arXiv.2206.05050.
- 4 Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: Ranking and clustering. *Journal of the ACM*, 55(5):23:1–23:27, 2008. doi:10.1145/1411509.1411513.
- 5 Noga Alon and Douglas B. West. The Borsuk-Ulam theorem and bisection of necklaces. *Proceedings of the American Mathematical Society*, 98(4):623–628, 1986. doi:10.2307/2045739.
- 6 Sayan Bandyapadhyay, Fedor V. Fomin, and Kirill Simonov. On coresets for fair clustering in metric and euclidean spaces and their applications. In *Proceedings of the 48th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 23:1–23:15, 2021. doi:10.4230/LIPIcs.ICALP.2021.23.
- 7 Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine Learning*, 56(1–3):89–113, 2004. doi:10.1023/B:MACH.0000033116.57574.95.
- 8 Amir Ben-Dor, Ron Shamir, and Zohar Yakhini. Clustering gene expression patterns. *Journal of Computational Biology*, 6(3–4):281–297, 1999. doi:10.1089/106652799318274.
- 9 Suman K. Bera, Deeparnab Chakrabarty, Nicolas J. Flores, and Maryam Negahbani. Fair algorithms for clustering. In *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS)*, pages 4954–4965, 2019.
- 10 Ioana Oriana Bercea, Martin Groß, Samir Khuller, Aounon Kumar, Clemens Rösner, Daniel R. Schmidt, and Melanie Schmidt. On the cost of essentially fair clusterings. In *Proceedings of the 2019 Conference on Approximation for Combinatorial Optimization Problems and the 2019 Conference on Randomization in Computation (APPROX/RANDOM)*, volume 145 of *LIPIcs*, pages 18:1–18:22, 2019.
- 11 Francesco Bonchi, David García-Soriano, and Francesco Gullo. *Correlation Clustering*. Morgan & Claypool Publishers, 2022. doi:10.2200/S01163ED1V01Y202201DMK019.
- 12 Sebastian Böcker and Jan Baumbach. Cluster editing. In *Proceedings of the 9th Conference on Computability in Europe (CiE)*, pages 33–44, 2013. doi:10.1007/978-3-642-39053-1_5.
- 13 Moses Charikar, Venkatesan Guruswami, and Anthony Wirth. Clustering with qualitative information. *Journal of Computer and System Sciences*, 71(3):360–383, 2005. doi:10.1016/j.jcss.2004.10.012.
- 14 Li Chen, Rasmus Kyng, Yang P. Liu, Richard Peng, Maximilian Probst Gutenberg, and Sushant Sachdeva. Maximum Flow and Minimum-Cost Flow in Almost-Linear Time. In *Proceedings of the 63rd Symposium on Foundations of Computer Science (FOCS)*, pages 612–623, 2022. doi:10.1109/FOCS54457.2022.00064.
- 15 Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair clustering through fairlets. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS)*, pages 5036–5044, 2017.
- 16 Vincent Cohen-Addad, Euiwoong Lee, and Alantha Newman. Correlation Clustering with Sherali-Adams. In *Proceedings of the 63rd Symposium on Foundations of Computer Science (FOCS)*, pages 651–661. IEEE, 2022. doi:10.1109/FOCS54457.2022.00068.
- 17 Julien Darlay, Nadia Brauner, and Julien Moncel. Dense and sparse graph partition. *Discrete Applied Mathematics*, 160(16):2389–2396, 2012. doi:10.1016/j.dam.2012.06.004.
- 18 Michael Dinitz, Aravind Srinivasan, Leonidas Tsepenekas, and Anil Vullikanti. Fair disaster containment via graph-cut problems. In *Proceedings of the 25th Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 6321–6333, 2022. URL: <https://proceedings.mlr.press/v151/dinitz22a.html>.

- 19 Thomas Epping, Winfried Hochstättler, and Peter Oertel. Complexity results on a paint shop problem. *Discrete Applied Mathematics*, 136:2-3:217–226, 2004. doi:10.1016/S0166-218X(03)00442-6.
- 20 Seyed A. Esmaeili, Brian Brubach, Leonidas Tsepenekas, and John P. Dickerson. Probabilistic fair clustering. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS)*, pages 12743–12755, 2020.
- 21 Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 259–268, 2015. doi:10.1145/2783258.2783311.
- 22 Andreas E. Feldmann and Luca Foschini. Balanced partitions of trees and applications. *Algorithmica*, 71(2):354–376, 2015. doi:10.1007/s00453-013-9802-3.
- 23 Zachary Friggstad and Ramin Mousavi. Fair correlation clustering with global and local guarantees. In *Proceedings of the 2021 Workshop on Algorithms and Data Structures (WADS)*, pages 414–427, 2021. doi:10.1007/978-3-030-83508-8_30.
- 24 Jonggyu Jang and Hyun Jong Yang. α -Fairness-maximizing user association in energy-constrained small cell networks. *IEEE Transactions on Wireless Communications*, 21(9):7443–7459, 2022. doi:10.1109/TWC.2022.3158694.
- 25 Suchi Kumari and Anurag Singh. Fair end-to-end window-based congestion control in time-varying data communication networks. *International Journal of Communication Systems*, 32(11), 2019. doi:10.1002/dac.3986.
- 26 Dana Pessach and Erez Shmueli. A review on fairness in machine learning. *ACM Computing Surveys*, 55(3):51:1–51:44, 2022. doi:10.1145/3494672.
- 27 Simon Régnier. Sur quelques aspects mathématiques des problèmes de classification automatique. *Mathématiques et Sciences Humaines*, 82:31–44, 1983.
- 28 Melanie Schmidt, Chris Schwiegelshohn, and Christian Sohler. Fair coresets and streaming algorithms for fair k -means. In *Proceedings of the 17th Workshop on Approximation and Online Algorithms (WAOA)*, pages 232–251, 2020. doi:10.1007/978-3-030-39479-0_16.
- 29 Roy Schwartz and Roded Zats. Fair correlation clustering in general graphs. In *Proceedings of the 2022 Conference on Approximation for Combinatorial Optimization Problems and the 2022 Conference on Randomization in Computation (APPROX/RANDOM)*, pages 37:1–37:19, 2022. doi:10.4230/LIPIcs.APPROX/RANDOM.2022.37.
- 30 Xiao Xin. An FPT algorithm for the correlation clustering problem. *Key Engineering Materials*, 474–476:924–927, 2011. doi:10.4028/www.scientific.net/KEM.474-476.924.
- 31 Charles T. Zahn, Jr. Approximating symmetric relations by equivalence relations. *Journal of the Society for Industrial and Applied Mathematics*, 12(4):840–847, 1964. doi:10.1137/0112071.
- 32 Imtiaz Masud Ziko, Jing Yuan, Eric Granger, and Ismail Ben Ayed. Variational fair clustering. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, pages 11202–11209, 2021. doi:10.1609/aaai.v35i12.17336.

Distributionally Robust Data Join

Pranjal Awasthi ✉

Google Research, NY, USA

Christopher Jung ✉

Stanford University, CA, USA

Jamie Morgenstern ✉

University of Washington, Seattle, WA, USA

Abstract

Suppose we are given two datasets: a labeled dataset and unlabeled dataset which also has additional auxiliary features not present in the first dataset. What is the most principled way to use these datasets together to construct a predictor?

The answer should depend upon whether these datasets are generated by the same or different distributions over their mutual feature sets, and how similar the test distribution will be to either of those distributions. In many applications, the two datasets will likely follow different distributions, but both may be close to the test distribution. We introduce the problem of building a predictor which minimizes the maximum loss over all probability distributions over the original features, auxiliary features, and binary labels, whose Wasserstein distance is r_1 away from the empirical distribution over the labeled dataset and r_2 away from that of the unlabeled dataset. This can be thought of as a generalization of distributionally robust optimization (DRO), which allows for two data sources, one of which is unlabeled and may contain auxiliary features.

2012 ACM Subject Classification Theory of computation → Machine learning theory

Keywords and phrases Distributionally Robust Optimization, Semi-Supervised Learning, Learning Theory

Digital Object Identifier 10.4230/LIPIcs.FORC.2023.10

Related Version *Full Version*: <https://arxiv.org/abs/2202.05797>

1 Introduction

For a variety of prediction tasks, a number of sources of data may be available on which to train, each possibly following a distinct distribution. For example, health records might be available from at a number of geographically and demographically distinct hospitals. How should one combine these data sources to build the best possible predictor?

If the datasets S_1, S_2 follow different distributions D_1, D_2 , the test distribution D will necessarily differ from at least one. A refinement of our prior question is to ask for which test distributions, then, can training with S_1, S_2 give a good predictor?

More generally, very common issues of mismatch between training and test distributions (and uncertainty over which test distribution one might face) has led to a great deal of interest in applying tools from distributionally robust optimization (DRO) to machine learning [12, 28, 24, 26]. In contrast to classical statistical learning theory, DRO picks a function f whose maximum loss (over a set of distributions near S) is minimized. This set of potential test distributions, often referred to as the ambiguity or uncertainty set, captures the uncertainty over the test distribution, along with knowledge that the test distribution will be close to the training distribution.

The ambiguity set is usually defined as a set of distributions with distance at most r from the empirical distribution over the training data: $B(\tilde{\mathcal{P}}_S, r) = \{Q : D(\tilde{\mathcal{P}}_S, Q) \leq r\}$ where $\tilde{\mathcal{P}}_S$ is the empirical distribution over training dataset S and D is some distance



© Pranjal Awasthi, Christopher Jung, and Jamie Morgenstern;
licensed under Creative Commons License CC-BY 4.0

4th Symposium on Foundations of Responsible Computing (FORC 2023).

Editor: Kunal Talwar; Article No. 10; pp. 10:1–10:15



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

10:2 Distributionally Robust Data Join

measure between two probability distributions. Then, DRO aims to find a model θ such that for some loss ℓ , $\theta = \arg \min_{\theta} \sup_{\mathcal{Q} \in B(\hat{\mathcal{P}}_S, r)} \mathbb{E}_{(x,y) \sim \mathcal{Q}}[\ell(\theta, (x, y))]$ – that is, minimize the loss over the worst case distribution in the ball of distributions $B(\hat{\mathcal{P}}_S, r)$. The larger r , the more distributions over which DRO hedges its performance, leading to a tension between performance (minimizing worst-case error) and robustness (over the set of distributions on which performance is measured).

In this work, we introduce a natural extension of distributionally robust learning, *two anchor* distributionally robust learning, which we also refer to as the distributionally robust data join problem. Two anchor distributionally robust learning has access to two sources of training data, the first source containing labels, and the second source without labels but with auxiliary features not present in the first source. The optimization is then over the set of distributions close to *both* the labeled and auxiliary data distributions.

Formally, suppose one has two training datasets. The first dataset S_1 consists of feature vectors $\mathcal{X} = \mathbb{R}^{m_1}$ and binary prediction labels for some task $\mathcal{Y} = \{\pm 1\}$. The other dataset S_2 contains feature vectors \mathcal{X} and auxiliary features $\mathcal{A} = \mathbb{R}^{m_2}$ but *not* the labels. The goal is to find a model θ that hedges its performance against any distribution \mathcal{Q} over $(\mathcal{X}, \mathcal{A}, \mathcal{Y})$ whose Wasserstein distance is r_1 away from the empirical distribution over S_1 and r_2 away from that of S_2 . Note that our setting is a strict generalization of semi-supervised setting: for $m_2 = 0$, there are no additional features in the second dataset, and S_2 is simply some additional unlabeled dataset. In contrast to pure semi-supervised settings, our method and setting both allow the learner to take advantage of the additional auxiliary features and to learn a model robust to additional distribution shift. We also emphasize that having the common features x between S_1 and S_2 help learn about the relationship between the auxiliary features a and the label y indirectly. Consider the following example where we actually have one dataset that contains the feature vector, auxiliary features, and the label altogether $S^{\text{combined}} = \{(x_i, a_i, y_i)\}_{i=1}^n$. From this dataset, we may form $S_1 = \{(x_i, a_i)\}_{i=1}^n$ and $S_2 = \{(x_i, y_i)\}_{i=1}^n$ where for every point (x_i, a_i) in S_1 and there's a corresponding (x_i, y_i) such that they share the same feature. In fact, instantiating our framework with $r_1 = 0$ and $r_2 = 0$ corresponds exactly to performing empirical risk minimization over S^{combined} . In other words, the quality of how well feature vectors x 's match between S_1 and S_2 determine how well we may be able to learn the relationship between the auxiliary features a and the label y .

In practice, it is quite common to have the datasets fragmented as our setting captures. For instance, suppose some dataset has been collected at a hospital in order to build a predictive model that is to be used at a nearby hospital. After collecting this data, some other research may find other features that could have been useful for the prediction task but unfortunately were not collected during the construction of this dataset. Fortunately, another nearby hospital may have data that contains both the original features and the useful auxiliary features but does not have labels for this prediction task. Our data join approach allows to find a model that utilizes such auxiliary features and explicitly considers the distribution mismatch between the hospital where the model is deployed and the hospitals from which these two datasets have been collected.

Auxiliary features may be useful not only for improving accuracy of the model but for guaranteeing additional properties including notions of fairness. In the appendix of the full version of the paper, we show that our distributionally robust data join problem encompasses a two-anchor distributionally robust learning instance where one can try to minimize not just the model's overall loss but also penalize the model for its difference in performance across demographic groups, even in situations where demographic information is present only in

one dataset and the label is only present in the other dataset. This extension is motivated by designing equitable predictors (e.g., which equalize false positive rate over a collection of demographic groups) where one training set contains labels for the relevant task but no demographic information, and another training set contains demographic information but may not contain task labels. Such settings are quite common in practice, where demographic data is not collected for every dataset – indeed, collection of demographic data is difficult to do well or sometimes even illegal [1, 15, 32, 34].

The contribution of our work can be summarized as follows:

1. **New Problem Formulation of Distributionally Robust Data Join:** we introduce and precisely formulate the distributionally robust data join problem in Section 2 and exactly characterize its feasibility in Section 3.1.
2. **Application to Fairness:** we further show how our original problem can be slightly modified to capture the problem of enforcing fairness when demographic group information is not available in the original labeled dataset (In the appendix of the full version of the paper).
3. **Tractable Reformulation with an Approximation Guarantee (Theorem 7 in Section 3):** we show how to approximate the distributionally robust data join problem with two convex optimization problems with an approximation guarantee.
4. **Experiments (Section 4):** we design and perform a synthetic experiment that shows how our distributionally robust data join method performs much better than the baselines. Additionally, we show some preliminary results on the experiments on a few real world datasets.

1.1 Related Work

Distributionally Robust Optimization: Prior work has looked at many different ways to define the ambiguity set: characterizing the set with moment and support information [8, 16, 33], or using various distance measures on probability space and defined the ambiguity set to be all the probability measures that are within certain distance ϵ of the empirical distribution: [12] use f-divergence, [18] the Kullback-Leibler divergence, [13] the Prohorov metric, and [28, 3, 2, 14] the Wasserstein distance, [17] chi-square divergence, and so forth. Defining ambiguity sets with divergence measures suffers from the fact that they do not incorporate the underlying geometry between the points – i.e. almost all divergence measures require the distribution in the ambiguity set to be absolutely continuous with respect to the anchor distribution. Therefore, because the distributions in the ambiguity set are simple re-weighting of the anchor distribution, divergence based ambiguity sets don't include distributions where the empirical distributions are perturbed a little bit and hence aren't robust to “black swan” outliers [23]. By contrast, the Wasserstein distance allows one to take advantage of the natural geometry of the points (e.g. L_p space). Furthermore, when we consider ambiguity sets defined by *two* anchor distributions as we do in this work, the two empirical distributions that are the anchors of the ambiguity set are almost surely not continuous with respect to each other. For these reasons, we focus on the Wasserstein distance in this work.

Most relevant to our work from the distributionally robust optimization literature is [28]. They show that regularizing the model parameter of the logistic regression has the effect of robustly hedging the model's performance against distributions whose distribution over just the covariates is slightly different than that of the empirical distribution over the training data. Distributionally robust logistic regression is a generalization of p -norm regularized logistic regression because it allows for not only distribution shift in the covariates but also the distribution shift over the labels. In a couple of real world datasets, they show that distributionally robust logistic regression seems to outperform regularized logistic regression

10:4 Distributionally Robust Data Join

by the same amount that regularized logistic regression outperforms vanilla logistic regression. Our work is a natural extension of this work in that we take additional unlabeled dataset with auxiliary features into account. However, we remark that our contributions go beyond the contributions of [28]. In particular, reasoning about couplings between 3 distributions (labeled dataset, unlabeled dataset, and unknown target dataset) as shown later in Section 2.2 is *a priori* not obvious and rather novel. Existing 2 distribution coupling approach used in [28] (e.g., creating one coupling between labeled and unlabeled, and another between one of these and the test distribution) will not give empirically or theoretically good matchings between all three distributions and will generally also not be computationally tractable in our case. We further discuss new technical difficulties that have to be overcome in order to solve our problem later in Section 3 and the appendix of the full version of the paper. [30] extend [28] by adding a fairness regularization term, but the demographic information is available in the original labeled dataset in their setting unlike our setting.

Semi-supervised Learning: There have been significant advances in semi-supervised learning where the learner has access not only labeled data but also unlabeled data [36, 35, 7]. While our setting is similar to semi-supervised settings, we capture a broader class of possible problems in two ways. First, our approach allows the unlabeled dataset to have additional auxiliary features, and second, we explicitly take distribution shift into account.

Imputation: Numerous imputation methods for missing values in data exist, many of which have few or no theoretical guarantees [11, 27]. Many of these methods work best (or only have guarantees) when data values are missing at random. Our work, on the other hand, assumes all prediction labels are missing from the second dataset and all auxiliary features are missing from the first dataset. Another related problem is the matrix factorization problem which is also referred to as matrix completion problem [25, 22, 4]: here the goal is to find a low rank matrix that can well approximate the given data matrix with missing values. Our problem is different in that we don't make such structural assumption about the data matrix effectively being of low rank, but instead we assume all the auxiliary features are only available from a separate unlabeled dataset.

Fairness: Many practical prediction tasks have disparate performance across demographic groups, and explicit demographic information may not be available in the original training data. Several lines of work aim to reduce the gap in performance of a predictor between groups even without group information for training.

[17] show that the chi-square divergence between the overall distribution and the distribution of any subgroup can be bounded by the size of the subgroup: e.g. for any sufficiently large subgroup, its divergence to the overall distribution cannot be too big. Therefore, by performing distributionally robust learning with ambiguity set defined by chi-square divergence, they are able to optimize for the worst-case risk over all possible sufficiently large subgroups even when the demographic information is not available. [9] provide provably convergence oracle-efficient learning algorithms with the same kind of minimax fairness guarantees when the demographic group information is available.

One may naively think that given auxiliary demographic group information data, the most accurate imputation for the demographic group may be enough to not only estimate the unfairness of given predictor but also build a predictor with fairness guarantees. However, [1] show that due to different underlying base rates across groups, the Bayes optimal predictor for the demographic group information can result in maximally biased estimate of unfairness.

[10] demonstrate that one can rely on a multi-accurate regressor, which was first introduced by [21], as opposed to a 0-1 classifier in order to estimate the unfairness without any bias and also build a fair classifier for downstream tasks. When only some data points are missing demographic information, [19] show how to bypass the need to explicitly impute the missing values and instead rely on some decision tree based approach in order to optimize a fairness-regularized objective function. [20], given two separate datasets like in our setting, show how to construct confidence intervals for unfairness that is consistent with the given datasets via Fréchet and Hoeffding inequalities; our work is different in that we allow a little bit of slack by forming a Wasserstein ball around both datasets and can actually construct a fair model as opposed to only measuring unfairness.

[5] and [6] have shown when the demographic group information is available but possibly noisy, stochastically and adversarially respectively, how to build a fair classifier.

2 Preliminaries

2.1 Notations

We have two kinds of datasets, the auxiliary feature dataset and the prediction label dataset denoted in the following way: $S_A = \{(x_i^A, a_i^A)\}_{i=1}^{n_A}$, $S_P = \{(x_i^P, y_i^P)\}_{i=1}^{n_P}$ where the domain for feature vector x is $\mathcal{X} = \mathbb{R}^{m_1}$, domain for auxiliary features a is $\mathcal{A} = \mathbb{R}^{m_2}$, and the label space is $y \in \mathcal{Y} = \{\pm 1\}$. For any vector $v \in \mathbb{R}^m$ and $d_1, d_2 \in [m]$, we write $v[d_1 : d_2]$ to denote the coordinates from d_1 to d_2 of vector v and $v[d]$ to denote the d th coordinate. We assume both \mathcal{X} and \mathcal{A} are compact and convex. For convenience, we write $S_A^{\mathcal{X}} = \{x : (x, a) \in S_A\}$, $S_P^{\mathcal{X}} = \{x : (x, y) \in S_P\}$ to denote just the feature vectors of the dataset.

Given any dataset $S = \{z_i\}_{i=1}^n$, we will write $\tilde{\mathcal{P}}_S = \frac{1}{n} \sum_{i=1}^n \delta(z_i)$ to denote the empirical distribution over the dataset S where δ is the Dirac delta function. We'll write \mathbb{P}_Z to denote the set of all probability distributions over Z . Similarly, we write $\mathbb{P}_{(Z, Z')}$ to denote a set of all possible joint distributions over Z and Z' . Also, given a joint distribution $\mathcal{P} \in \mathbb{P}_{(Z, Z')}$, we write \mathcal{P}_Z and $\mathcal{P}_{Z'}$ to denote the marginal distribution over Z and Z' respectively, meaning $\mathcal{P}_Z(z) = \int \mathcal{P}(z, dz')$ and $\mathcal{P}_{Z'}(z') = \int \mathcal{P}(dz, z')$. We extend the notation when the joint distribution is over more than two sets: e.g. $\mathcal{P}_{z, z'}((z, z')) = \int \mathcal{P}(z, z', dz'')$ where we have marginalized over Z'' for \mathcal{P} which is a joint distribution over Z, Z', Z'' .

We write the set of all possibly couplings between two distributions $\mathcal{P} \in \mathbb{P}_Z$ and $\mathcal{P}' \in \mathbb{P}_{Z'}$ as $\Pi(\mathcal{P}, \mathcal{P}') = \{\pi \in \mathbb{P}_{(Z, Z')} : \pi_Z = \mathcal{P}, \pi_{Z'} = \mathcal{P}'\}$. For a coupling between more than two distributions, we use the same convention and write $\Pi(\mathcal{P}, \mathcal{P}', \mathcal{P}'')$ for instance.

Given any metric $d : Z \times Z \rightarrow \mathbb{R}$ and two probability distributions $\mathcal{P}, \mathcal{P}' \in \mathbb{P}_Z$, we write the Wasserstein distance between them as $D_d(\mathcal{P}, \mathcal{P}') = \inf_{\pi \in \Pi(\mathcal{P}, \mathcal{P}')} \mathbb{E}_{(z, z') \sim \pi} [d(z, z')]$.

Given some distribution $\mathcal{P} \in \mathbb{P}$ over some set Z , metric $d : Z \times Z \rightarrow \mathbb{R}$, a radius $r > 0$, we will write $B_d(\mathcal{P}, r) = \{\mathcal{Q} \in \mathbb{P} : D_d(\mathcal{P}, \mathcal{Q}) \leq r\}$ to denote the Wasserstein ball of radius r around the given distribution \mathcal{P} . When the metric is obvious from the context, we may simply write $B(\mathcal{P}, r)$.

In our case, the relevant metrics that are used to measure distance between points are

$$\begin{aligned} d_{\mathcal{X}}(x, x') &= \|x - x'\|_p, & d_A((x, a), (x', a')) &= \|x - x'\|_p + \kappa_A \|a - a'\|_{p'} \\ d_P((x, y), (x', y')) &= \|x - x'\|_p + \kappa_P |y - y'| \end{aligned}$$

where $\|v\|_p = (\sum_d |v[d]|^p)^{\frac{1}{p}}$ is some p -norm and $\kappa_A, \kappa_P \geq 0$ are the coefficients that control how much we care about the $\|a - a'\|_{p'}$ and $|y - y'|$. We'll write $\|v\|_{p,*} = \sup_{\|v'\|_p \leq 1} \langle v, v' \rangle$ to denote dual norm for p -norm. Also, for convenience, given any vector v , we'll write

10:6 Distributionally Robust Data Join

$\bar{v}_p = \frac{v}{\|v\|_p}$ and $\bar{v}_{p,*} = \frac{v}{\|v\|_{p,*}}$ to denote the normalized vectors. When it's clear from the context which norm is being used, we write $\|\cdot\|$, $\|\cdot\|_*$, \bar{v} , and \bar{v}_* . Now, we are ready to describe distributionally robust data join problem.

2.2 Distributionally Robust Data Join

We are given an auxiliary dataset S_A and a prediction label dataset S_P . We are interested in a joint distribution \mathcal{Q} over (x, a, y) such that

1. its marginal distribution over (x, a) is at most r_A away from $\tilde{\mathcal{P}}_{S_A}$ in Wasserstein distance:
 $\mathcal{D}_{d_A}(\mathcal{Q}_{\mathcal{X},\mathcal{A}}, \tilde{\mathcal{P}}_{S_A}) \leq r_A$
2. its marginal distribution over (x, y) is at most r_P away from $\tilde{\mathcal{P}}_{S_P}$ in Wasserstein distance:
 $\mathcal{D}_{d_P}(\mathcal{Q}_{\mathcal{X},\mathcal{Y}}, \tilde{\mathcal{P}}_{S_P}) \leq r_P$

Combining them together, the set of distributions we are interested in is

$$\begin{aligned} W(S_A, S_P, r_A, r_P) &= \{\mathcal{Q} \in \mathbb{P}_{(\mathcal{X},\mathcal{A},\mathcal{Y})} : \mathcal{D}_{d_A}(\mathcal{Q}_{\mathcal{X},\mathcal{A}}, \tilde{\mathcal{P}}_{S_A}) \leq r_A, \mathcal{D}_{d_P}(\mathcal{Q}_{\mathcal{X},\mathcal{Y}}, \tilde{\mathcal{P}}_{S_P}) \leq r_P\} \\ &= \{\mathcal{Q} \in \mathbb{P}_{(\mathcal{X},\mathcal{A},\mathcal{Y})} : \mathcal{Q}_{\mathcal{X},\mathcal{A}} \in B_{d_A}(\tilde{\mathcal{P}}_{S_A}, r_A), \mathcal{Q}_{\mathcal{X},\mathcal{Y}} \in B_{d_P}(\tilde{\mathcal{P}}_{S_P}, r_P)\}. \end{aligned}$$

Now, we consider some learning task where the performance is measured according to the worst case distribution in the above set of distributions. We want to find some model parameter θ such that its loss against the worst-case distribution among $W(S_A, S_P, r_A, r_P)$ is minimized:

$$\min_{\theta \in \Theta} \sup_{\mathcal{Q} \in W(S_A, S_P, r_A, r_P)} \mathbb{E}_{(x,a,y) \sim \mathcal{Q}} [\ell(\theta, (x, a, y))]. \quad (1)$$

where $\ell : \Theta \times (\mathcal{X} \times \mathcal{A} \times \mathcal{Y}) \rightarrow \mathbb{R}$ is a convex loss function evaluated at θ . For the sake of concreteness, we focus on logistic loss¹ $\ell(\theta, (x, a, y)) = \log(1 + \exp(-y\langle \theta, (x, a) \rangle))$.

Also, we sometimes make use of the following functions $f(t) = \log(1 + \exp(t))$ and $h(\theta, (x, a)) = f(-\langle \theta, (x, a) \rangle)$ instead of ℓ , as it is more convenient due to not having to worry about y in certain cases: $\ell(\theta, (x, a, +1)) = h(\theta, (x, a))$ and $\ell(\theta, (x, a, -1)) = h(-\theta, (x, a))$. We write the convex conjugate of f as $f^*(b) = \sup_x \langle x^*, x \rangle - f(x)$, which in our case evaluates to $b \log b + (1 - b) \log(1 - b)$ when $b \in (0, 1)$, 0 if $b = 0$ or 1, and ∞ otherwise.

3 Tractable Reformulation

Let us give an overview of this section. Note that the optimization problem in (1) is a saddle point problem. In Section 3.1, we first make the coupling in the optimal transport more explicit in the inner sup term. Then, by leveraging Kantorovich duality, we replace the sup term with its dual problem which is a minimization problem, thereby making the original saddle problem into minimization problem. However, the resulting dual problem has constraints that involve some supremum term, meaning it's an semi-infinite program (i.e. $\sup_{z \in Z} \text{constraint}(z) \leq 0$ is equivalent to $\text{constraint}(z) \leq 0, \forall z \in Z$). Finally, in Section 3.3, we show how each supremum term can be approximated by some other closed-form constraint. And we finally show that the resulting problem can be decomposed into two convex optimization problems and its optimal solution has additional approximation guarantee to the original optimal solution (Theorem 7).

¹ All our results still hold for any other convex loss with minimal modifications

3.1 Formulation through Coupling

We show how to rewrite the problem (1) using the underlying coupling between the “anchor” distributions (S_A, S_P) and $\mathcal{Q} \in W(S_A, S_P, r_A, r_P)$. For simplicity, instead of $\pi((x_i^A, a_i^A), (x_j^P, y_j^P), (x, a, y))$ which is a coupling between $\tilde{\mathcal{P}}_{S_A}$, $\tilde{\mathcal{P}}_{S_P}$, and some joint distribution $\mathcal{Q} \in \mathbb{P}_{\mathcal{X}, \mathcal{A}, \mathcal{Y}}$, we write $\pi_{i,j}^y(x, a) = \pi((x_i^A, a_i^A), (x_j^P, y_j^P), (x, a, y))$. Then, since the “anchor” distributions $\tilde{\mathcal{P}}_{S_A}$ and $\tilde{\mathcal{P}}_{S_P}$ are discrete distributions, we can rewrite the problem (1) as choosing $\theta \in \Theta$ that minimizes the following value:

$$\begin{aligned} \sup_{\pi_{i,j}^{a,y}} & \sum_{i=1}^{n_A} \sum_{j=1}^{n_P} \sum_{y \in \mathcal{Y}} \int_{\mathcal{X}, \mathcal{A}} \ell(\theta, (x, a, y)) \pi_{i,j}^y(dx, da) & (2) \\ \text{s.t.} & \sum_{i=1}^{n_A} \sum_{j=1}^{n_P} \sum_{y \in \mathcal{Y}} \int_{\mathcal{X}, \mathcal{A}} d_A^i(x, a) \pi_{i,j}^y(dx, da) \leq r_A, & \sum_{i=1}^{n_A} \sum_{j=1}^{n_P} \sum_{y \in \mathcal{Y}} \int_{\mathcal{X}, \mathcal{A}} d_P^j(x, y) \pi_{i,j}^y(dx, da) \leq r_P \\ & \sum_{j=1}^{n_P} \sum_{y \in \mathcal{Y}} \int_{\mathcal{X}, \mathcal{A}} \pi_{i,j}^y(dx, da) = \frac{1}{n_A} \quad \forall i \in [n_A], & \sum_{i=1}^{n_A} \sum_{y \in \mathcal{Y}} \int_{\mathcal{X}, \mathcal{A}} \pi_{i,j}^y(dx, da) = \frac{1}{n_P} \quad \forall j \in [n_P] \end{aligned}$$

where $d_A^i(x, a) = d_A((x_i^A, a_i^A), (x, a))$ and $d_P^j(x, y) = d_P((x_j^P, y_j^P), (x, y))$. We defer intuitive explanations and derivation of this problem to the appendix of the full version of the paper. For any fixed parameter θ , we’ll denote the optimal value of the above problem (2) as $p^*(\theta, r_A, r_P)$ and $p^*(r_A, r_P) = \inf_{\theta} p^*(\theta, r_A, r_P)$.

It can be shown that minimizing over the above supremum value in (1) and the optimization problem (2) are equivalent as shown in the following theorem. We also provide a tight characterization of the feasibility of (2). The proof of Theorem 1 and 2 can be found in the appendix of the full version of the paper.

► **Theorem 1.** *For any fixed $\theta \in \Theta$,*

$$p^*(\theta, r_A, r_P) = \sup_{\mathcal{Q} \in W(S_A, S_P, r_A, r_P)} \mathbb{E}_{(x,a,y) \sim \mathcal{Q}} [\ell(\theta, (x, a, y))].$$

► **Theorem 2.** *$D_{d_{\mathcal{X}}}(\tilde{\mathcal{P}}_{S_A^{\mathcal{X}}}, \tilde{\mathcal{P}}_{S_P^{\mathcal{X}}}) \leq r_A + r_P$, if and only if there exists a feasible solution for (2).*

3.2 Strong Duality

We claim that the following problem is the dual to problem (2) and show that strong duality holds between them:

$$\begin{aligned} \inf_{\substack{\alpha_A, \alpha_P, \\ \{\beta_i\}, \{\beta'_j\}}} & \alpha_A r_A + \alpha_P r_P + \frac{1}{n_A} \sum_{i \in [n_A]} \beta_i + \frac{1}{n_P} \sum_{j \in [n_P]} \beta'_j & (3) \\ \text{s.t.} & \sup_{(x,a)} \left(\ell(\theta, (x, a, y)) - \alpha_A d_A^i(x, a) - \alpha_P d_P^j(x, y) \right) \leq \beta_i + \beta'_j \quad \forall i \in [n_A], j \in [n_P], y \in \mathcal{Y} \end{aligned}$$

For fixed θ , we’ll write $d^*(\theta, r_A, r_P)$ to denote the optimal value for the above dual problem (3). As in [28], strong duality directly follows from [29], but to be self-contained, we include the proof in the appendix of the full version of the paper, which follows the same proof structure presented in [31].

► **Theorem 3.** *If there exists a feasible solution for the primal problem (2), then we have that strong duality holds between the primal problem (2) and its dual problem (3): $p^*(\theta, r_A, r_P) = d^*(\theta, r_A, r_P)$ for fixed θ .*

In other words, we have successfully transformed the saddle point problem (1) into a minimization problem over θ and the dual variables $\alpha_A, \alpha_P, \{\beta_i\}$ and $\{\beta'_j\}_j$:

$$\begin{aligned} & \min_{\substack{\theta \in \Theta, \alpha_A, \alpha_P, \\ \{\beta_i\}, \{\beta'_j\}}} \alpha_A r_A + \alpha_P r_P + \frac{1}{n_A} \sum_{i \in [n_A]} \beta_i + \frac{1}{n_P} \sum_{j \in [n_P]} \beta'_j & (4) \\ \text{s.t. } & \max_{y \in \{\pm 1\}} \sup_{(x,a)} (\ell(\theta, (x, a, y)) - \alpha_A d_A^i(x, a) - \alpha_P d_P^j(x, y)) \leq \beta_i + \beta'_j \quad \forall i \in [n_A], j \in [n_P] \end{aligned}$$

3.3 Replacing the sup Term

Note that $\sup_{(x,a)}$ in the constraint makes it hard to actually compute the expression: it's neither concave or convex in terms of (x, a) as it's the difference between convex functions $\ell(\theta, (x, a, y))$ and $\alpha_A d_A^i(x, a) + \alpha_P d_P^j(x, y)$. In that regard, we show how to approximate the sup term in the constraint of dual problem (3) with some closed form expression by extending the techniques used in [28] who study when there's only one "anchor" point – i.e. $\sup_x \ell(\theta, x) - \alpha d_{\mathcal{X}}(x_i, x)$ as opposed to in our case with two anchor points.

First, let's focus only on the terms that actually depend on (x, a) and ignore our dependence on y briefly:

$$\begin{aligned} & \sup_{(x,a)} \ell(\theta, (x, a, y)) - \alpha_A d_A^i(x, a) - \alpha_P d_P^j(x, y) \\ & = \kappa_P \alpha_P |y_j^P - y| + \left(\sup_{(x,a)} h(y\theta, (x, a)) - \alpha_A \|x_i^A - x\|_p - \alpha_P \|x_j^P - x\|_p + \alpha_A \kappa_A \|a_i^A - a\|_{p'} \right). \end{aligned}$$

We obtain an upper bound for the supremum term in the lemma below whose full proof can be found in the appendix of the full version of the paper.

► **Theorem 4.** *Fix any $y \in \mathcal{Y}$ and θ . Write $\theta_1 = \theta[1 : m_1]$ and $\theta_2 = [m_1 + 1 : m_1 + m_2]$. Suppose $p \neq 1$ and $p \neq \infty$. If $\|\theta_1\|_{p,*} \leq \alpha_A + \alpha_P$ and $\|\theta_2\|_{p',*} \leq \kappa_A \alpha_A$, then*

$$\begin{aligned} & \sup_{(x,a)} h(y\theta, (x, a)) - \alpha_A \|x_i^A - x\|_p - \alpha_P \|x_j^P - x\|_p - \alpha_A \kappa_A \|a_i^A - a\|_{p'} \\ & \leq f \left(\left(\frac{\min(\alpha_A, \alpha_P) \|\theta_1\|_* \|x_i^A - x_j^P\|}{\alpha_A + \alpha_P} + \frac{\langle y\theta_1, \alpha_A x_i^A + \alpha_P x_j^P \rangle}{\alpha_A + \alpha_P} \right) + \langle y\theta_2, a_i^A \rangle \right) \\ & \quad - \min(\alpha_A, \alpha_P) \|x_i^A - x_j^P\|_p. \end{aligned}$$

Otherwise, $\sup_{(x,a)} h(y\theta, (x, a)) - \alpha_A \|x - x_i^A\|_p - \alpha_P \|x - x_j^P\|_p - \alpha_A \kappa_A \|a_i^A - a\|_{p'}$ evaluates to ∞ .

Proof Sketch. Similar to [28], we leverage convex conjugacy in order to re-express the sup term. However, because we have multiple anchor points, the re-expression results in an infimal convolution of *two* linear functions with bounded norm constraints as opposed to the case of [28] where they only have to handle a convex conjugate of a *single* linear function with bounded norm constraint and hence find an exact closed form expression. Therefore, in the appendix of the full version of the paper, we develop new techniques where we show (1) infimal convolution of linear functions with norm constraints is convex, (2) obtain a closed form solution of the infimal convolution at two extreme points, and (3) use linear interpolation of these extreme points to obtain an upper-bound, as a line segment of the two extreme points sits above the graph for convex functions. ◀

Equipped with the above upper bound on the supremum term, we can imagine trying to replace the supremum term with the above upper bound in order to get a feasible dual solution to the dual problem (4). However, one may worry that there is a big gap between the original supremum term and our upperbound in Theorem 4.

To this end, we further show that we can in fact approximate the supremum term with one more trick and hence obtain an approximate dual solution. Suppose we write

$$\hat{x}_{i,j} = \begin{cases} x_j^P & \text{if } \alpha_A < \alpha_P \\ x_i^A & \end{cases} \quad \text{and } \hat{\alpha} = \min(\alpha_A, \alpha_P). \text{ Note that by definition, the value}$$

measured at $(\hat{x}_{i,j}, a_i^A)$ is a lower bound on the supremum. In other words, we have

$$\begin{aligned} & h(y\theta, (\hat{x}_{i,j}, a_i^A)) - \alpha_A \|x_i^A - \hat{x}_{i,j}\|_p - \alpha_P \|x_j^P - \hat{x}_{i,j}\|_p = f(\langle y\theta, (\hat{x}_{i,j}, a_i^A) \rangle) - \hat{\alpha} \|x_i^A - x_j^P\|_p \\ & \leq \sup_{(x,a)} h(y\theta, (x, a)) - \alpha_A \|x_i^A - x\|_p - \alpha_P \|x_j^P - x\|_p - \alpha_A \kappa_A \|a_i^A - a\|_{p'} \\ & \leq f \left(\left(\frac{\min(\alpha_A, \alpha_P) \|\theta_1\|_* \|x_i^A - x_j^P\|}{\alpha_A + \alpha_P} + \frac{\langle y\theta_1, \alpha_A x_i^A + \alpha_P x_j^P \rangle}{\alpha_A + \alpha_P} \right) + \langle y\theta_2, a_i^A \rangle \right) - \hat{\alpha} \|x_i^A - x_j^P\|_p. \end{aligned}$$

Now, via Hölder's inequality, we can show the lower bound and the upper bound above on the supremum term are in fact very close, meaning by using either the upper bound or the lower bound, we can approximate the supremum very well. Here's a lemma that shows that the value evaluated at $(\hat{x}_{i,j}, a_i^A)$ is pretty close to the upper bound in Theorem 4:

► **Lemma 5.**

$$\begin{aligned} & f \left(\left(\frac{\min(\alpha_A, \alpha_P) \|\theta_1\|_* \|x_i^A - x_j^P\|}{\alpha_A + \alpha_P} + \frac{\langle y\theta_1, \alpha_A x_i^A + \alpha_P x_j^P \rangle}{\alpha_A + \alpha_P} \right) + \langle y\theta_2, a_i^A \rangle \right) - f(\langle y\theta, (\hat{x}_{i,j}, a_i^A) \rangle) \\ & \leq 2\hat{\alpha} \|x_i^A - x_j^P\|. \end{aligned}$$

In other words, replacing the original supremum constraint with a constraint evaluated at $(\hat{x}_{i,j}, a_i^A)$ will not incur too much additional error. Finally, using the fact that $f(-t) = f(t) + t$ for logistic function f , we can bring back the terms that depend on y and approximate the original supremum constraint in the following manner:

► **Corollary 6.**

$$\begin{aligned} & \left(\max_{y \in \{\pm 1\}} \sup_{(x,a)} \left(\ell(\theta, (x, a, y)) - \alpha_A d_A^i(x, a) - \alpha_P d_P^j(x, y) \right) \right) \\ & - \left(f(\langle y_j^P \theta, (\hat{x}_{i,j}, a_i^A) \rangle) + \max(y_j^P \langle \theta, (\hat{x}_{i,j}, a_i^A) \rangle - \alpha_P \kappa_P, 0) - \hat{\alpha} \|x_i^A - x_j^P\| \right) \\ & \leq 2\hat{\alpha} \|x_i^A - x_j^P\| \end{aligned}$$

In other words, replacing the supremum constraint with the constraint evaluated at $(\hat{x}_{i,j}, a_i^A)$ and using the above trick to remove the max over y will arrive at the following problem, for which we provide an approximation guarantee in Theorem 7.

$$\begin{aligned} & \min_{\alpha_A, \alpha_P, \theta_1, \theta_2, \{\beta_i\}, \{\beta'_j\}} (\alpha_A r_A + \alpha_P r_P) + \frac{1}{n_A} \sum_{i \in [n_A]} \beta_i + \frac{1}{n_P} \sum_{j \in [n_P]} \beta'_j \tag{5} \\ \text{s.t. } & f(y_j^P \langle \theta, (\hat{x}_{i,j}, a_i^A) \rangle) + \max(y_j^P \langle \theta, (\hat{x}_{i,j}, a_i^A) \rangle - \alpha_P \kappa_P, 0) - \hat{\alpha} \|x_i^A - x_j^P\| \\ & \leq \beta_i + \beta'_j \quad \forall i \in [n_A], j \in [n_P] \\ & \|\theta_1\|_* \leq \alpha_A + \alpha_P, \|\theta_2\|_* \leq \kappa_A \alpha_A. \end{aligned}$$

► **Theorem 7.** *We can solve problem (5) by solving two convex optimization problems. And the optimal θ^* for the above problem (5) is such that its objective value for the original problem (1) is at most $2\hat{\alpha} \max_{i \in [n_A], j \in [n_P]} \|x_i^A - x_j^P\|$ greater than the optimal solution:*

$$\begin{aligned} & \sup_{\mathcal{Q} \in \mathcal{W}(S_A, S_P, r_A, r_P)} \mathbb{E}_{(x,a,y) \sim \mathcal{Q}} [\ell(\theta^*, (x, a, y))] - 2\hat{\alpha} \max_{i \in [n_A], j \in [n_P]} \|x_i^A - x_j^P\| \\ & \leq \min_{\theta \in \Theta} \sup_{\mathcal{Q} \in \mathcal{W}(S_A, S_P, r_A, r_P)} \mathbb{E}_{(x,a,y) \sim \mathcal{Q}} [\ell(\theta, (x, a, y))]. \end{aligned}$$

10:10 Distributionally Robust Data Join

■ **Table 1** Average accuracy of each method over 10 experiment runs and standard deviations for synthetic dataset with a distribution shift.

	LR	RLLR	DRLR	DJ
Accuracy	0.4126 ± 0.1049	0.5786 ± 0.3992	0.9068 ± 0.0076	0.9923 ± 0.0057

Just as in [28], two convex optimization problems that problem (5) decomposes into can be solved by IOPT and YALMIP. In addition, we remark that $2\hat{\alpha} \max_{i \in [n_A], j \in [n_P]} \|x_i^A - x_j^P\|$ is a reasonable approximation guarantee because this value should be in the same order as $\alpha_A r_A + \alpha_P r_P$: recall that we have argued in Theorem 2, a feasible solution exists if and only if $D_{d_X}(\tilde{\mathcal{P}}_{S_A^X}, \tilde{\mathcal{P}}_{S_P^X}) \leq r_A + r_P$. Additionally, the worst case pairwise distance can actually be improved with an additional assumption: since any underlying coupling for the Wasserstein distance most likely transports non-zero probability mass between only close points, we can imagine considering only the k-nearest-neighbors of each point as opposed to all possible pairs between two datasets, hence decreasing the approximation error to the maximal pairwise distance between some point and its k-nearest-neighbor. We make this point more formal in the appendix of the full version of the paper.

4 Experiments

We now describe an experimental evaluation of our method on a synthetic dataset and real world datasets. In all our experiments, we use the approach discussed in the appendix of the full version in which we make practically simplifying assumptions in order to solve the problem (5) via projected gradient descent. We use 2-norm throughout the experiments: i.e. $p, p' = 2$.

4.1 Synthetic Data

We briefly discuss how we create the synthetic dataset. We want our synthetic data generation process to encompass the components that are unique to our robust data join setting – namely, distribution shift and auxiliary unlabeled dataset that contains additional features that should help with the prediction task.

To that end, we discuss the data generation process at a high level here and more fully in Appendix B. We have two groups such that the ideal hyperplane that distinguishes the positive and negative points is different for each group. We introduce distribution shift into the setting by having the original labeled training dataset consist mostly of points from the first group and the test dataset consist mostly from the second group. As for specific details of the data generation process that are important for our setting, we have one of the features to carry information regarding which group the point belongs to.

As for the unlabeled dataset with auxiliary features, the points will mostly come from the second group, hence being closer to the test distribution. Furthermore, we include additional features that are present in the unlabeled dataset to be highly correlated with the true label, although this unlabeled dataset doesn't contain the true label of each point.

Because we want our baselines that compare our distributionally robust data join approach (DJ) against to be in the same model class (i.e. logistic regression) as our method for fair comparison, we consider the following baselines:

1. LR: Vanilla logistic regression trained on labeled dataset S_P
2. RLLR: Regularized logistic regression trained on labeled dataset S_P
3. DRLR: Distributionally robust logistic regression trained on S_P

■ **Table 2** Average accuracy of each method over 10 experiment runs and standard deviations for three UCI datasets.

	BC ($m_1 = 5$)	BC ($m_1 = 25$)	IO ($m_1 = 4$)	IO ($m_1 = 25$)	HD	1vs8
DJ	0.9140 \pm 0.0368	0.9281 \pm 0.0155	0.8208 \pm 0.0816	0.7896 \pm 0.04885	0.7495 \pm 0.0374	0.90841 \pm 0.0270
LR	0.9012 \pm 0.0294	0.9140 \pm 0.0393	0.7764 \pm 0.1560	0.7868 \pm 0.0653	0.7286 \pm 0.0504	0.8729 \pm 0.0337
RLR	0.9053 \pm 0.0228	0.9287 \pm 0.0199	0.7915 \pm 0.1417	0.7868 \pm 0.0690	0.7363 \pm 0.0565	0.8953 \pm 0.0250
LRO	0.8789 \pm 0.0318	0.8789 \pm 0.0318	0.7330 \pm 0.0788	0.7330 \pm 0.0788	0.6626 \pm 0.0569	0.7766 \pm 0.0599
RLRO	0.8953 \pm 0.0212	0.8953 \pm 0.0212	0.7377 \pm 0.0800	0.7377 \pm 0.0800	0.6714 \pm 0.0568	0.8710 \pm 0.0450
FULL	0.9684 \pm 0.0143	0.9684 \pm 0.0143	0.8754 \pm 0.0764	0.8754 \pm 0.0764	0.8319 \pm 0.0311	0.9495 \pm 0.0222

The result of this experiment can be found in Table 1. There are few plausible reasons as to why our approach (DJ) does extremely well in this synthetic experiment. Our distributionally robust data join is definitely taking advantage of the proximity of unlabeled dataset to the test distribution in that the majority of points are both from the second group. Although regularized and distributionally robust logistic regression is trying to be robust against some form of distribution shift, the set of distributions they are hedging against may be too big as they are hedging against all distributions that are close to the empirical distribution over the labeled dataset. By contrast, the set of distributions that distributionally robust data join may be smaller because it’s hedging against the set of distributions that are close to the labeled dataset *and* the unlabeled dataset. Finally, auxiliary features in the unlabeled dataset are providing information very relevant for the prediction task.

4.2 UCI Datasets

Here we discuss some experiments we have run and show that as a proof of concept, our distributionally robust data join framework has the potential to be practical empirically. However, we remark unlike in the synthetic data experiment, we do not introduce any distribution shift (i.e. training and test are iid samples from the same distribution) and also choose the additional features for the unlabeled dataset in an arbitrary way because of our lack of contextual expertise of the features in each dataset. Therefore, the gaps between our method and the baselines we consider are not as impressive as the performance gap we see in the synthetic experiments.

We use four UCI datasets for our real world dataset experiment: Breast Cancer dataset (BC), Ionosphere dataset (IO), Heart disease dataset (HD), and Handwritten Digits dataset with 1’s and 8’s (1vs8). We provide more details about these datasets in Appendix B. For all these datasets, each experiment run consists of the following: (1) randomly divide the dataset into $S_{\text{train}} = \{(x_i, a_i, y_i)\}_{i=1}^{n_{\text{train}}}$ and S_{test} , (2) create the prediction label dataset and auxiliary dataset where v data points belong to both datasets: $S_P = \{(x_i, y_i)\}_{i=1}^{n_P+v}$ and $S_A = \{(x_i, a_i)\}_{i=n_P+1}^{n_{\text{train}}}$.

We compare our method of joining S_A and S_P , which we denote as DJ, to the following baselines:

1. LR: Logistic regression trained on S_P
2. RLR: Regularized logistic regression on S_P
3. LRO: Logistic regression on overlapped data $\{(x_i, a_i, y_i)\}_{i=n_P+1}^{n_P+v}$
4. RLRO: Regularized logistic regression on overlapped data $\{(x_i, a_i, y_i)\}_{i=n_P+1}^{n_P+v}$.
5. FULL: full training on $\{(x_i, a_i, y_i)\}_{i=1}^{n_{\text{train}}}$

where FULL is simply to show the highest accuracy we could have achieved if the labeled dataset actually had the auxiliary features and the unlabeled dataset had the labels. The results of the experiment can be found in Table 2, and we include further details of the

experiment in Appendix B. Without any distribution shift, the distributionally robust data join method is solving a somewhat harder problem than the other baselines because of its hedging against other nearby distributions. Yet it can be seen that the use of the additional auxiliary features through our data join method helps achieve better accuracy than the baselines.

References

- 1 Pranjali Awasthi, Alex Beutel, Matthäus Kleindessner, Jamie Morgenstern, and Xuezhi Wang. Evaluating fairness of machine learning models under uncertain and incomplete information. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 206–214, 2021.
- 2 Jose Blanchet, Yang Kang, and Karthyek Murthy. Robust wasserstein profile inference and applications to machine learning. *Journal of Applied Probability*, 56(3):830–857, 2019.
- 3 Jose Blanchet and Karthyek Murthy. Quantifying distributional model risk via optimal transport. *Mathematics of Operations Research*, 44(2):565–600, 2019.
- 4 Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009.
- 5 L Elisa Celis, Lingxiao Huang, Vijay Keswani, and Nisheeth K Vishnoi. Fair classification with noisy protected attributes: A framework with provable guarantees. In *International Conference on Machine Learning*, pages 1349–1361. PMLR, 2021.
- 6 L Elisa Celis, Anay Mehrotra, and Nisheeth K Vishnoi. Fair classification with adversarial perturbations. *arXiv preprint*, 2021. [arXiv:2106.05964](https://arxiv.org/abs/2106.05964).
- 7 Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.
- 8 Erick Delage and Yinyu Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations research*, 58(3):595–612, 2010.
- 9 Emily Diana, Wesley Gill, Michael Kearns, Krishnaram Kenthapadi, and Aaron Roth. Minimax group fairness: Algorithms and experiments. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 66–76, 2021.
- 10 Emily Diana, Wesley Gill, Michael Kearns, Krishnaram Kenthapadi, Aaron Roth, and Saeed Sharifi-Malvajerdi. Multiaccurate proxies for downstream fairness. *arXiv preprint*, 2021. [arXiv:2107.04423](https://arxiv.org/abs/2107.04423).
- 11 A Rogier T Donders, Geert JMG Van Der Heijden, Theo Stijnen, and Karel GM Moons. A gentle introduction to imputation of missing values. *Journal of clinical epidemiology*, 59(10):1087–1091, 2006.
- 12 John C Duchi and Hongseok Namkoong. Learning models with uniform performance via distributionally robust optimization. *The Annals of Statistics*, 49(3):1378–1406, 2021.
- 13 Emre Erdoğan and Garud Iyengar. Ambiguous chance constrained problems and robust optimization. *Mathematical Programming*, 107(1):37–61, 2006.
- 14 Peyman Mohajerin Esfahani and Daniel Kuhn. Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations. *Mathematical Programming*, 171(1):115–166, 2018.
- 15 Allen Fremont, Joel S Weissman, Emily Hoch, and Marc N Elliott. When race/ethnicity data are lacking. *RAND Health Q*, 6:1–6, 2016.
- 16 Joel Goh and Melvyn Sim. Distributionally robust optimization and its tractable approximations. *Operations research*, 58(4-part-1):902–917, 2010.
- 17 Tatsunori Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. Fairness without demographics in repeated loss minimization. In *International Conference on Machine Learning*, pages 1929–1938. PMLR, 2018.

- 18 Zhaolin Hu and L Jeff Hong. Kullback-leibler divergence constrained distributionally robust optimization. *Available at Optimization Online*, 2013.
- 19 Haewon Jeong, Hao Wang, and Flavio P Calmon. Fairness without imputation: A decision tree approach for fair prediction with missing values. *arXiv preprint*, 2021. [arXiv:2109.10431](https://arxiv.org/abs/2109.10431).
- 20 Nathan Kallus, Xiaojie Mao, and Angela Zhou. Assessing algorithmic fairness with unobserved protected class using data combination. *Management Science*, 2021.
- 21 Michael P Kim, Amirata Ghorbani, and James Zou. Multiaccuracy: Black-box post-processing for fairness in classification. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 247–254, 2019.
- 22 Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.
- 23 Daniel Kuhn, Peyman Mohajerin Esfahani, Viet Anh Nguyen, and Soroosh Shafieezadeh-Abadeh. Wasserstein distributionally robust optimization: Theory and applications in machine learning. In *Operations research & management science in the age of analytics*, pages 130–166. INFORMS, 2019.
- 24 Jaeho Lee and Maxim Raginsky. Minimax statistical learning with wasserstein distances. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 2692–2701, 2018. URL: <https://proceedings.neurips.cc/paper/2018/hash/ea8fcd92d59581717e06eb187f10666d-Abstract.html>.
- 25 Andriy Mnih and Russ R Salakhutdinov. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264, 2008.
- 26 Hamed Rahimian and Sanjay Mehrotra. Distributionally robust optimization: A review. *arXiv preprint*, 2019. [arXiv:1908.05659](https://arxiv.org/abs/1908.05659).
- 27 Patrick Royston. Multiple imputation of missing values. *The Stata Journal*, 4(3):227–241, 2004.
- 28 Soroosh Shafieezadeh-Abadeh, Peyman Mohajerin Esfahani, and Daniel Kuhn. Distributionally robust logistic regression. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1576–1584, 2015. URL: <https://proceedings.neurips.cc/paper/2015/hash/cc1aa436277138f61cda703991069eaf-Abstract.html>.
- 29 Alexander Shapiro. On duality theory of conic linear problems. In *Semi-infinite programming*, pages 135–165. Springer, 2001.
- 30 Bahar Taskesen, Viet Anh Nguyen, Daniel Kuhn, and Jose Blanchet. A distributionally robust approach to fair classification. *arXiv preprint*, 2020. [arXiv:2007.09530](https://arxiv.org/abs/2007.09530).
- 31 Cédric Villani. *Topics in optimal transportation*. American Mathematical Soc., 2003.
- 32 Joel S Weissman and Romana Hasnain-Wynia. Advancing health care equity through improved data collection. *The New England journal of medicine*, 364(24):2276–2277, 2011.
- 33 Wolfram Wiesemann, Daniel Kuhn, and Melvyn Sim. Distributionally robust convex optimization. *Operations Research*, 62(6):1358–1376, 2014.
- 34 Yan Zhang. Assessing fair lending risks using race/ethnicity proxies. *Management Science*, 64(1):178–197, 2018.
- 35 Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009.
- 36 Xiaojin Jerry Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005.

A Possible Negative Societal Impact and Limitations

We do not foresee any direct negative societal impact of our work. However, just as other distributionally robust optimization methods, our robust guarantees may come at the price of achieving slightly worse accuracy. However, we note that this trade-off between more robustness and higher utility can be controlled by setting r_A and r_P appropriately. On a related note, another limitation of our approach is that it requires specifying r_A and r_P ; one needs to have some knowledge about how “far” the distributions (i.e. labeled dataset, unlabeled dataset with auxiliary features, and test distribution) may be, which is a limitation as in other methods that require setting some hyperparameters appropriately.

B Missing Details from Section 4

All the experiments were performed on one of the authors’ personal computer, MacBook Pro 2017, and every experiment took less than an hour.

We note that as it’s standard in practice to output the last iterate instead of the averaged iterate, we use the last iterate of the projected gradient descent instead of the averaged one for all our experiments. Now, the total number of points and the features for each dataset is here along with where the dataset can be found:

1. BC (<https://archive.ics.uci.edu/ml/datasets/breast+cancer>): 569 points with 30 features
2. IO (<https://archive.ics.uci.edu/ml/datasets/ionosphere>): 351 points with 34 features
3. HD (<https://archive.ics.uci.edu/ml/datasets/Heart+Disease>): 300 points with 13 features
4. 1vs8 (https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_digits.html#sklearn.datasets.load_digits): This is a copy of the test dataset from <https://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits>). It originally contains 1797 points with 64 points. But after filtering out all the digits except for 1’s and 8’s, there are 356 points.

For every dataset, we preprocess the data by standardizing each feature – that is, removing the mean and scaling to unit variance.

We take the common feature to be the first 5 features for (BC, HD) and 4 for IO – i.e. $m_1 = 5$ and 4 respectively. For 1vs8, we have $m_1 = 32$, the first half bits of the 8x8 image. And the remaining features are the auxiliary features \mathcal{A} : $m_2 = 25, 30, 8$, and 32 for BC, IO, HD, and 1vs8 respectively. For all datasets, we set the test size to be 30% of the entire dataset. Then, we set $(n_P, v) = (20, 5), (20, 10), (30, 5), (30, 10)$ for BC, IO, HD, 1vs8 respectively. In other words, we imagine the total number of points in our labeled sets S_P and the number of features to be very small. For BC and IO, we also try a case when the number of common features is a lot more (i.e. $m_1 = 25$).

Now we report the best regularization penalties that maximize the accuracy of RLR and RLRO respectively over all experiment runs at the granularity level of 10^{-2} . The best regularization penalty for RLR and RLRO were $\lambda = (0.07, 0.04)$ for BC ($m_1 = 5$), $(0.04, 0.04)$ for BC ($m_1 = 25$), $(0.02, 0.02)$ for IO ($m_1 = 4$), $(0.01, 0.02)$ for IO ($m_1 = 25$), $(0.08, 0.03)$ for HD, and $(0.08, 0.08)$ for 1vs8. The parameters for data join used for each of the datasets can be found in the table below:

For all of the methods (logistic regression, regularized logistic regression, distributionally robust logistic regression, and our distributionally robust data join), the learning rate used was $7 * 10^{-2}$ and the total number of iterations was 1500.

■ **Table 3** Parameters used for distributionally data join (DJ) for UCI datasets.

	BC ($m_1 = 5$)	BC ($m_1 = 25$)	IO ($m_1 = 4$)	IO ($m_1 = 25$)	HD	1vs8
r_A	0.65	1.65	0.3	1.5	0.65	1.85
r_P	0.65	1.65	0.3	1.5	0.65	1.85
κ_A	5	5	10	5	10	5
κ_P	2.5	2.5	5	2.5	5	15
k	1	1	1	1	1	1

Finally, we describe how we generated the data that was used to test how well DJ handles distribution shift. First, define

$$\beta_1 = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0] \quad \text{and} \quad \beta_2 = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1].$$

For the first group $g = 1$, the positive points and negative points were drawn from a multivariate normal distribution with mean β_1 and $-\beta_1$ respectively both with the standard deviation of 0.2:

$$x|y = +1, g = 1 \sim N(\beta_1, 0.2) \quad \text{and} \quad x|y = -1, g = 1 \sim N(-\beta_1, 0.2).$$

For the second group $g = 2$, the positive points and negative points were drawn from a multivariate normal distribution with mean β_2 and $-\beta_2$ respectively both with the standard deviation of 0.3:

$$x|y = +1, g = 2 \sim N(\beta_2, 0.2) \quad \text{and} \quad x|y = -1, g = 2 \sim N(-\beta_2, 0.2).$$

Now, for the first dataset $S_1 = \{(x_j^1, y_j^1)\}_{j=1}^{n_1}$, we had the number of points from group 1 and from group 2 was 400 and 20 respectively. And we had it so that the number of positive and negative points in each group was exactly the same: i.e. 200 positive and negative points for group 1, and 10 positive and 10 negative points for group 2.

For the second dataset, $S_2 = \{(x_i^2, y_i^2)\}_{i=1}^{n_2}$, the number of points from group 1 and from group 2 was 200 and 2000 respectively. The number of positive and negative points in each group was exactly the same once again here.

Our labeled dataset will be the first two coordinates of the first dataset, meaning $m_1 = 2$:

$$S_P = \{(x_j^1[0:2], y_j^1)\}_{j=1}^{n_1}.$$

Then, we will randomly divide the second dataset so that the 70% of it will be used as unlabeled dataset S_A and the other 30% is to be used as the test dataset S_{test} .

$$S_A = \{x_i^2\}_{i=1}^{0.7n_2} \quad \text{and} \quad S_{\text{test}} = \{(x_i^2, y_i^2)\}_{i=0.7n_2+1}^{n_2}.$$

Note that $m_2 = 10$.

The baselines that we consider for this synthetic data experiment are

1. Logistic regression trained (LR) on S_P
2. Regularized regression trained (RLR) on S_P with $\lambda = 10$
3. Distributionally logistic regression (DLR) trained on S_P with $r = 100, \kappa = 10$.

Resistance to Timing Attacks for Sampling and Privacy Preserving Schemes

Yoav Ben Dov


Weizmann Institute of Science, Rehovot, Israel

Liron David  

Weizmann Institute of Science, Rehovot, Israel

Moni Naor  

Weizmann Institute of Science, Rehovot, Israel

Elad Tzalik 

Weizmann Institute of Science, Rehovot, Israel

Abstract

Side channel attacks, and in particular timing attacks, are a fundamental obstacle for secure implementation of algorithms and cryptographic protocols. These attacks and countermeasures have been widely researched for decades. We offer a new perspective on resistance to timing attacks.

We focus on sampling algorithms and their application to differential privacy. We define sampling algorithms that do not reveal information about the sampled output through their running time. More specifically: (1) We characterize the distributions that can be sampled from in a “time oblivious” way, meaning that the running time does not leak any information about the output. We provide an optimal algorithm in terms of randomness used to sample for these distributions. We give an example of an efficient randomized algorithm \mathcal{A} such that there is no subexponential algorithm with the same output as \mathcal{A} that does not reveal information on the output or the input, therefore we show leaking information on either the input or the output is unavoidable. (2) We consider the impact of timing attacks on (pure) differential privacy mechanisms. It turns out that if the range of the mechanism is unbounded, such as counting, then any *time oblivious* pure DP mechanism must give a useless output with constant probability (the constant is mechanism dependent) and must have infinite expected running time. We show that up to this limitations it is possible to transform *any* pure DP mechanism into a time oblivious one.

2012 ACM Subject Classification Mathematics of computing → Random number generation; Theory of computation → Cryptographic primitives; Theory of computation → Generating random combinatorial structures

Keywords and phrases Differential Privacy

Digital Object Identifier 10.4230/LIPIcs.FORC.2023.11

Funding Research supported in part by grants from the Israel Science Foundation (no.2686/20), by the Simons Foundation Collaboration on the Theory of Algorithmic Fairness and by the Israeli Council for Higher Education (CHE) via the Weizmann Data Science Research Center.

1 Introduction

There is always a gap between the way an algorithm is specified and described mathematically and how it is implemented in a physical device and environment. Physical systems often leak information to the environment, for example the power usage, heat radiation, running time and much more. This leakage, in turn, can make systems which are secure in the theoretical model, susceptible to attacks in practice which make them completely insecure. These attacks are called “side channel attacks.” In this work we focus on timing attacks, i.e. attacks that exploit the running time leakage.



© Yoav Ben Dov, Liron David, Moni Naor, and Elad Tzalik;

licensed under Creative Commons License CC-BY 4.0

4th Symposium on Foundations of Responsible Computing (FORC 2023).

Editor: Kunal Talwar; Article No. 11; pp. 11:1–11:23

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

11:2 Resistance to Timing Attacks for Sampling and Privacy Preserving Schemes

One important field which is sensitive to timing attacks is *Differential Privacy* (DP) [9, 10]. DP deals with analyzing data sets in a way which protects the privacy of an individual contributor to the collection. Informally, for an algorithm to be differentially private it needs to have “close” output distributions on two data sets which differ by a single entity¹. In this work we consider the question of defining security against timing attacks. To this end, we define new notions of resistance to timing attacks in the realm of sampling and differential privacy, provide resistant constructions, and prove their security.

We formally define *time oblivious* DP mechanisms. We show that time oblivious pure DP mechanisms have some undesirable properties and therefore the recommendation is to use approximate DP in practice. Nevertheless, we show that if one can tolerate those properties in applications, then any pure DP mechanism can be transformed to a time oblivious pure DP mechanism with similar privacy guarantee.

1.1 A Very Brief History of Timing Attacks

Side channel attacks and in particular timing attacks have long history and we would not attempt to survey it (see the companion paper [6] for more details). For instance, an early work by Lipton and Naughton [21] showed a way to exploit timing information to compromise the performance of dictionaries that employ universal hash functions. The work of Kocher [20], showing how the running time of certain implementations of RSA and Diffie Hellman schemes leaks information which can be used to break the systems and put the issue in the forefront of research in the area.

One of the most efficient lattice based digital signature schemes is BLISS, suggested by Ducas, Durmus, Lepoint and Lyubashevsky [7]. This scheme uses a bimodal Gaussian sampler and was shown to be vulnerable to timing attacks, and in particular the *sampling component* is not independent of the secret-key [11, 3], as well as other attacks. These vulnerabilities might be the reason the scheme did not emerge as an option in the Post-Quantum NIST standardization process.

The differential privacy (DP) setting has had its own share of issues with respect to leaky implementations. Starting with Mironov [22] who showed that the problems of finite precision arithmetic imply that pioneering implementations of differentially private databases actually do not satisfy the desired properties. To address this Balcer and Vadahn [2] considered designing DP-algorithms that can be implemented in *strict* polynomial time. More recently, Andryscio et al. [1] showed that various concrete implementations differentially private mechanisms are vulnerable to timing channels. Ilvennto [13] suggested implementing the *Exponential Mechanism* with “Base-2 Differential Privacy,” which meant it could be implemented with finite precision, but left open the issue of timing attack resilience.

1.2 Prevention Techniques

The main approach to prevent timing attacks is to use *fixed time algorithms*, often called in the literature “constant time algorithms,” meaning algorithms that run the same amount of time on all inputs.

There are two main drawbacks to this solution. First, in order for the algorithm to run in fixed time on all the inputs, we need to know the worst case running time, a task that is often challenging on its own. The second one is that even if we do know the running time,

¹ There are two variants to differential privacy, pure and approximate. In the case of pure DP, the results should be close pointwise.

in many cases there is a very large gap between *best case and worst case running time*, or even *average case and worst case running time*, and by making the algorithm run in the worst case time on all inputs, we create huge overheads. It is also worth mentioning that the second caveat can make many protocols and algorithms impractical and not usable when efficiency is critical.

In addition, the survey in Section 1.1 demonstrates that the task of making an algorithm run in fixed time is more subtle and challenging than meets the eye. Timing information can leak from response times of the server, from I/O calls, from reading RAM memory or cache memory and many more possibilities. In order for the algorithm to be truly and fully fixed time, one must make sure to make everything fixed time, which is often very challenging, and goes against hardware and software optimizations.

A common technique to thwart timing attacks in the public-key context is “blinding,” first suggested by Chaum [4] in the context of signatures, where a value v is mapped into a random looking one u prior to the encryption or signature, in a manner that allows to retrieve the desired signature or encryption from the encryption or signature on u . Kocher [20] suggested using blinding to make RSA implementations secure against timing attacks. The blinding works by multiplying the input x by a fresh random element r of the group \mathbb{Z}_N^* , i.e. a random element which is co-prime to N . To decode, a multiplication by the group inverse r^{-1} is done at the end of the computation. Note that simply using the same r for many inputs will not work, as the attack suggested by Kocher can recover r over time, and even recover the exponent without knowing r . Hence, fresh r needs to be chosen in each round. This example goes to show that using blinding as a technique to protect against timing attack is often a subtle task, and that if implemented naively or incorrectly can lead to a false sense of security.

A general approach to preventing leakage is to employ techniques from secure multi-party computation, and split the input into various parts where leaking *almost* all of them does not leak the actual values. It was first suggested in Ishai, Sahai and Waters [14] for thwarting probing attacks (see [16] for a survey). This can be thought of as the “moral equivalent” of blinding for a general functions. However, in case of timing, given that what is leaked is a function of *all* parties (at the very least, the sum of their running times), it is not clear that that it solves the problem. Nevertheless, it does point to the issue of the number of random bits used to generate a sample.

1.3 Our Contributions and Technical Overview

Our goal in this work is to investigate the landscape of algorithms and systems that can be implemented in a manner resistant to timing attacks, but we wish to expand the ‘Procrustean bed’ of fixed time algorithms. We provide foundational treatment to the subject as well as many algorithms and separation results.

In Section 2 We focus on the security against timing attacks in an information theoretic manner and focuses on randomized algorithms and in particular on *sampling algorithms*. We define sampling algorithms secure against timing attacks, “time oblivious” sampling algorithms. Our main result of this section is a characterization of the the distributions that can be sampled without leaking *any* information on the output.

► **Theorem 2.3.** *Let \mathcal{D} be a discrete probability distribution. Then \mathcal{D} has a time oblivious generating algorithm if and only if the following holds:*

1. \mathcal{D} has finite support.
2. \mathcal{D} is rational, i.e. all the probabilities of possible outputs are rational.

We also give an optimal time oblivious algorithm to sample from a rational distribution of finite support, where optimal means that any other time oblivious algorithm has slower running time.

In Section 3 we concentrate on the implication of these results to *Differential Privacy*, i.e. what happens to such mechanisms when their running time (or the number of random bits used) is leaked (recall from Section 1.1 that there is a history of leakage problems in DP implementations). Since many differential privacy mechanisms work by taking the input and adding to it noise generated by some distribution with an infinite support, it is clear from the discussion above that this approach is futile when trying to resist timing leakage.

We formally define *time oblivious DP mechanisms*:

► **Definition 3.1.** *Let $\mathcal{M}: \mathcal{C} \rightarrow \mathcal{R}$ be a randomized algorithm. We say \mathcal{M} is (ε, δ) -differentially private time oblivious mechanism if for every pair of neighboring datasets D and D' , every subset $S \subseteq \mathcal{R} \times \mathbb{N}$*

$$\mathbb{P}[(\mathcal{M}(D), \mathcal{T}(\mathcal{M}(D))) \in S] \leq e^\varepsilon \cdot \mathbb{P}[(\mathcal{M}(D'), \mathcal{T}(\mathcal{M}(D'))) \in S] + \delta.$$

We show that the situation is more complex. Some techniques, such as randomized response, work here provided the biased coin flipped is rational. In case the range is unbounded, as in counting in a database whose size is not known, then it is impossible to guarantee *useful* results with very high probability. That is, for any (time oblivious pure DP mechanism with an unbounded range there is a $\rho > 0$ s.t. for most databases the mechanism outputs *useless results with probability at least ρ* (See Claim 3.4). But on the other hand, we can take any DP mechanism and find a pointwise close (for each database) DP mechanism that is time oblivious as we prove in Theorem 3.6:

► **Theorem 3.6.** *Let \mathcal{M} be any ε -pure DP mechanism with a discrete range \mathcal{R} . For any $\gamma > 0$, $\varepsilon' > \varepsilon$ there is a time oblivious ε' pure DP mechanism \mathcal{M}_{obl} such that $\|\mathcal{M}_{obl}(D) - \mathcal{M}(D)\|_{TV} < \gamma$.*

Finally, we supplement Section 2 with Appendix A which deals with the problem of sampling a satisfying assignment of a DNF formula in a time oblivious way. We show how to convert the well known *non* time oblivious algorithm into a time oblivious algorithm while preserving the run-time. In addition, we show that leaking information on the formula is unavoidable. Specifically, we show that *any* time oblivious algorithm for sampling a satisfying assignment from a DNF formula that “hides the formula” must run in exponential time, and therefore we show an *inherent exponential gap* for hiding the input of a randomized algorithm.

► **Theorem A.4.** *Sampling a uniform satisfying assignment of a DNF formula in an input hiding time oblivious way cannot be done efficiently and requires $\Omega(2^n)$ bits in expectation.*

2 Time Oblivious Sampling: Definitions and Characterization

The challenge of designing an algorithm that generates a distribution \mathcal{D} using a sequence of unbiased coins $C_1, C_2, \dots \sim \text{Bernoulli}(\frac{1}{2})$ was studied in the seminal work of Knuth and Yao [19] in the mid 1970s. They described a greedy algorithm to generate \mathcal{D} and showed that it is optimal in terms of the expected number of coin flips. In addition they discuss properties of the algorithm, such as expected number of coins used, optimality, computational efficiency and more. A detailed discussion about their work can be found in Chapter 15 (“The random bit model”) of Devroye [5]

Their work appeared many years before the public discussion on side channel attacks and they did not address this issue. One possible timing attack is to measure the number of coin flips used by the algorithm. To see how this information may be useful, consider the example of sampling from $Geo(\frac{1}{2})$ by tossing coins until we get for the first time “heads”, and the number of tosses is the output generated. Clearly in this example, an adversary who knows the number of coin flips knows exactly what element was sampled. Such leakage can compromise cryptographic systems which rely on private randomness.

Another scenario where such leakage may be problematic occurs in the context of Differential Privacy (DP) [10]. Informally, the requirement of DP is for neighbouring datasets, to have “close” output distribution. By close we mean up to some multiplicative and additive factor. Removing the additive factor gives a stronger notion of DP, called pure DP. One main motivation for DP is to be able to get meaningful statistics from data, while preserving the privacy of the individual. A very common technique to achieve DP algorithms is to sample from some noise distribution, for example the Laplace distribution, and add it to the actual result.

As pointed by Balcer and Vadhan [2], the runtime of the noise generation can leak information about the noise that was generated, which in turn can make the noisy output not as hiding as well as in the idealized world. This may compromise the DP guarantee of the algorithm, especially if it is a pure DP algorithm.

Therefore, in order for the algorithm to satisfy the DP definition, it is *essential* that the running time of the noise generator will not give information to an adversary regarding the value sampled. This is discussed more thoroughly in Section 3.

Distributed sampling

A case where the number of bits used clearly leaks is when the generation is done distributively, using some sort of multi-party computation (e.g. when creating a root key or in the context of differential privacy [8]). In this case, the amount of communication (in bits) between the parties is directly related to the number of bits needed for the generation, so to keep the value generated hidden we need to make sure that the number of random bits consumed does not leak information.

2.1 Preliminaries, Notation and Definition

Let \mathcal{D} be a discrete distribution on \mathbb{N} , with $d_i = \mathbb{P}[\mathcal{D} = i]$. We say \mathcal{D} is rational if each d_i is a rational number, and \mathcal{D} has finite support if: $\text{supp}(\mathcal{D}) := \{j \mid d_j > 0\}$ is a finite set.

For a randomized algorithm \mathcal{A} let \mathcal{R} denote its random tape and $\mathcal{R}[i] \sim \text{Bernoulli}(\frac{1}{2})$ is the i^{th} bit in the tape. Then \mathcal{R}_n is the n -bit random string: $\mathcal{R}[0]\mathcal{R}[1] \dots \mathcal{R}[n-1]$. We assume a randomized algorithm \mathcal{A} reads the tape sequentially, and after reading \mathcal{R}_n decides deterministically, after a finite amount of steps, whether to return an output or read $\mathcal{R}[n]$ from the tape. The output distribution of \mathcal{A} over random tape \mathcal{R} will be denoted by $\mathcal{O}(\mathcal{A})$, and the number of bits read from the random tape by \mathcal{A} is the random variable (R.V.) $\mathcal{T}(\mathcal{A})$.

A useful way to view the generation of $\mathcal{O}(\mathcal{A})$ by Knuth and Yao is to consider \mathcal{R} as defining a random walk on an infinite binary tree in which going left corresponds to reading a 0 and going right corresponds to reading a 1, for this reason we consider a binary sequence $a_0 \dots a_n$ as a node. Since \mathcal{A} is deterministic given $\mathcal{R}_{n+1} = a_0 \dots a_n$, then for the node $a_0 \dots a_n$ in the tree \mathcal{A} either halt and outputs, or \mathcal{A} “walks” to either $a_0 \dots a_n 0$ or $a_0 \dots a_n 1$ with equal probability, $\frac{1}{2}$.

In light of the view above we define the i^{th} level of the tree to be the set of all binary sequences of length i . We will abuse the notation and use $a_0 \dots a_{i-1}$ to denote the corresponding integer associated with the binary sequence. We say that $a_0 \dots a_{i-1}$ precedes $b_0 \dots b_{i-1}$ if it is smaller as an integer. We say that \mathcal{A} outputs on a binary sequence $a_0 \dots a_{i-1}$ if conditioned on $\mathcal{R}_i = a_0 \dots a_{i-1}$ the algorithm reads the first i bits of \mathcal{R} and outputs before reading $\mathcal{R}[i]$. The i^{th} level is called an output level of \mathcal{A} if \mathcal{A} outputs on some sequence $a_0 \dots a_{i-1}$ in that level. The set of all output levels of \mathcal{A} is denote by $\mathcal{L}(\mathcal{A})$. Finally $\mathcal{T}(\mathcal{A})$ will denote the distribution of the number of bits \mathcal{A} read from the random tape \mathcal{R} .

When considering the runtime of a randomized algorithms an important resource is *the number of random bits that are read from the random tape*. Having this number be independent of the instance generated is a prerequisite to time obliviousness. Furthermore, once the required number of random bits has been read, the problem is a deterministic computation of a mapping and this can be performed in some worst case time for the given size. So at least in principle there is an independent implementation.

Motivated by the discussion above, we make the following definition:

► **Definition 2.1.** We say \mathcal{A} is a **time oblivious generating algorithm** if its output distribution and running time distribution are independent, meaning $\mathcal{O}(\mathcal{A})$ and $\mathcal{T}(\mathcal{A})$ are independent random variables.

2.2 Characterization of Time Oblivious Distributions

A natural question to ask is what distributions can be generated in a time oblivious way? For example, can we generate the distribution $Geo(\frac{1}{2})$ that was discussed at the beginning of the section? What about sampling a biased coin where the probability of '1' is p and '0' otherwise? Can we do it for all values of p ? We focus on the exact model, meaning we want to sample from the *exact* distribution and not some approximation.² Consider the following “separating bit” algorithm for sampling a biased coin with bias p : Let $p = 0.p_1p_2\dots$ be the binary expansion of p , we toss fair coins x_1, x_2, \dots to generate a number x between 0 and 1, $x = 0.x_1x_2\dots$. We stop in the first index i where $p_i \neq x_i$ we return x_i . We call the algorithm “separating bit” because we toss coins until we find the first index the separates the binary representation of p from the binary representation of the number x we generate.

Notice that:

▷ **Claim 2.2.** Let p be the bias of the generated coin then:

1. The “separating bit” algorithm is not time oblivious.
2. The expected number of random bits read by the algorithm is 2.

Proof.

1. If the algorithm produced an output after a single coin flip, we know that the result of the coin flip was $1 - p_1$ and therefore the output is $\mathbb{1}_{p_1=1}$.
2. Each coin has probability $\frac{1}{2}$ to be the separating bit, therefore the number of bits read is $Geo(\frac{1}{2})$ and the expected amount of bits read is 2. ◁

While Claim 2.2 shows that the separating bit algorithm is not time oblivious for all p , it does not mean there is no time oblivious algorithm to toss a p -biased coin to some values of $p \neq 1/2$. We show that we can toss a p biased coin iff p is rational. More generally, we will show:

² For an example of previous work on exact sampling, consider Feldman et al. [12] who addressed the issue of how many different types of coins one needs in order to generate a die roll in an exact manner.

► **Theorem 2.3.** *Let \mathcal{D} be a discrete probability distribution. Then \mathcal{D} has a time oblivious generating algorithm if and only if the following holds:*

1. \mathcal{D} has finite support.
2. \mathcal{D} is rational, i.e. all the probabilities of possible outputs are rational.

This characterization answers the questions above, we cannot sample from $Geo(\frac{1}{2})$ in a time oblivious way, and we can generate a biased coin if and only if p is rational. We will prove the theorem by showing each direction separately, and begin by showing:

► **Lemma 2.4.** *If a distribution \mathcal{D} has a time oblivious algorithm, then it is rational with finite support.*

Proof. Let \mathcal{A} be a time oblivious generating algorithm for \mathcal{D} . Since \mathcal{A} has output distribution \mathcal{D} , it means in particular that it outputs at some level, i.e. $\mathcal{L}(\mathcal{A})$, is not empty. Let k be the first output level of \mathcal{A} . Denote by m_k the number of output nodes of length k , and by e_j the number of nodes which output j in that level. The number of sequences of length k is 2^k and so we get that: $0 < m_k \leq 2^k < \infty$. Observe that the probability to get an output j in level k is exactly $\frac{e_j}{m_k}$. Since \mathcal{A} is time oblivious, conditioning on $\mathcal{T}(\mathcal{A}) = k$ yields the same output distribution \mathcal{D} and so for all j we have $d_j = \frac{e_j}{m_k}$ and therefore $d_j \in \mathbb{Q}$ and \mathcal{D} is rational. Notice that $d_j = \frac{e_j}{m_k}$ means that $d_j > 0$ implies that $d_j \geq \frac{1}{m_k}$ and therefore $|\text{supp}(\mathcal{D})| \leq m_k$. We get that \mathcal{D} is rational with finite support. ◀

We will later prove the other direction of Theorem 2.3 by describing a time oblivious algorithm to output \mathcal{D} . In the binary tree view described in Section 2.1 being time oblivious means that in each output level the conditional distribution given that the level was reached is \mathcal{D} (i.e. the output nodes of the level are distributed exactly according to \mathcal{D}). From now on, in light of Lemma 2.4, if \mathcal{D} is generated by a time oblivious algorithm we may assume $\text{supp}(\mathcal{D}) = \{1, \dots, n\}$ and use the notation $d_j = \frac{p_j}{q_j}$ for the output distribution and let $q := \text{LCM}(q_1, \dots, q_n)$.³ We now prove some properties of general time oblivious algorithms. We will need the following definition:

► **Definition 2.5.** *A node $a_0 \dots a_n$ is called reachable if no prefix $a_0 \dots a_m$ with $m < n$ is an output node.*

The following lemma applies for any time oblivious algorithm \mathcal{A} :

► **Lemma 2.6.** *Let \mathcal{A} be a time oblivious algorithm with a finite and rational output distribution \mathcal{D} where the LCM of the probabilities is q . Let t_k be the number of nodes in level k that are either unreachable or output nodes, and m_k the number of output nodes at level k , then:*

1. The number m_i of output nodes in level i is a multiple of q .
2. t_k satisfies: $t_k = \sum_{i=0}^k 2^{k-i} m_i$. In particular t_k is a multiple of q .

Proof. Proof in Appendix B. ◀

A corollary of Lemma 2.6 is:

► **Corollary 2.7.** *Let \mathcal{D} be a discrete distribution. \mathcal{D} can be generated in finite worst case complexity if and only if \mathcal{D} is rational with finite support and $q = 2^k$.*

Proof. Proof in Appendix B. ◀

³ The LCM is the Least Common Multiple of a set of natural numbers, that is the smallest natural number that is divisible by the given set of numbers.

11:8 Resistance to Timing Attacks for Sampling and Privacy Preserving Schemes

We now describe a time oblivious algorithm that generates \mathcal{D} . Following Lemma 2.4, we assume that the support of \mathcal{D} is $[n]$, and \mathcal{D} is given by a list of length n of the probabilities $d_i = \frac{p_i}{q_i}$ for co-prime p_i and q_i . Since the distribution has finite support, $q = LCM(q_1, \dots, q_n)$ can be computed efficiently (and we assume is part of the input). By Lemma 2.6 we know that any time oblivious algorithm should output a multiple of q elements in each level. Observe that if a level has q reachable nodes then we can partition these nodes into sets $\{S_i\}_{i=1}^n$, where $|S_i| = d_i \cdot q = \frac{p_i}{q_i} \cdot q \in \mathbb{N}$ and output i at S_i .

With this in mind, the algorithm works as follows: the algorithm iterates level by level and if there are q reachable nodes then it assign them q outputs similarly to the S_i 's above. Picking arbitrary reachable nodes may be inefficient (even though it will be optimal in terms of randomness, the formal definition may be found in Definition 2.12). To make the algorithm efficient the algorithm picks consistently the q left most reachable nodes in each level. This is now formally described in Algorithm 1:

■ **Algorithm 1** Gen(\mathcal{D}, \mathcal{R}).

```

1:  $q \leftarrow LCM(q_1, \dots, q_k)$ 
2:  $n \leftarrow 0$ 
3: while True do
4:    $n \leftarrow n + 1$ 
5:   if  $(2^n \bmod 2q) \geq q$  then ▷ If there are  $q$  reachable nodes in level  $n$ 
6:     if  $(\mathcal{R}_n \bmod 2q) \leq q - 1$  then ▷ Is  $\mathcal{R}_n$  one of the  $q$  leftmost reachable nodes
7:       Return GetValue( $\mathcal{D}$ ,  $(\mathcal{R}_n \bmod 2q)$ )
```

■ **Algorithm 2** GetValue(\mathcal{D}, j).

```

1:  $q \leftarrow LCM(q_1, \dots, q_k)$ 
2:  $s_0 \leftarrow 0$ 
3: for  $i = 1$  to  $k$  do
4:    $s_i \leftarrow s_{i-1} + q \cdot \frac{p_i}{q_i}$ 
5: Return  $i$  such that  $s_{i-1} \leq j < s_i$  ▷ Binary Search the value of  $i$ 
```

The following lemma justifies the comments in Algorithm 1 and specifies properties of its output nodes:

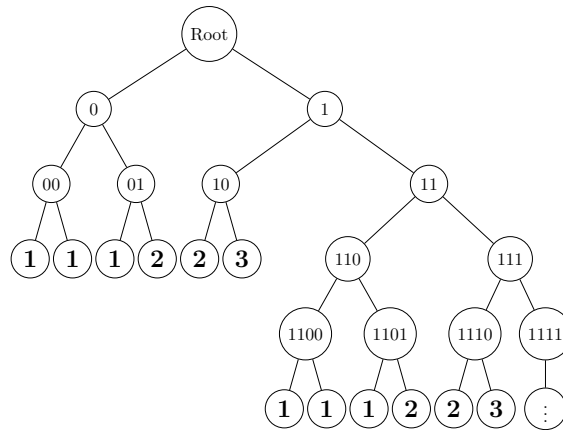
► **Lemma 2.8.**

1. The number of reachable nodes in level n is exactly $2^n \bmod 2q$.
2. Let $a_0 \dots a_{n-1}$ be the node in the tree that represents reading the bits a_0, \dots, a_{n-1} from the randomness tape. The number of reachable nodes preceding a reachable node $a_0 \dots a_{n-1}$ of Algorithm 1 is $a_0 \dots a_{n-1} \bmod 2q$.
3. Let t_n be the number of nodes in level n which are either unreachable or output nodes. For all n we have: $2^n - t_n < q$.

Proof. Proof in Appendix B. ◀

We now finish the proof of Theorem 2.3 and show that Algorithm 1 is time oblivious:

▷ **Claim 2.9.** Algorithm 1 is time-oblivious algorithm and generates \mathcal{D} .



■ **Figure 1** Distribution Generating Tree of \mathcal{D}_0 using Algorithm 1.

Proof. By Parts 1,3 of Lemma 2.8 at each level n of Algorithm 1 there are q output nodes that correspond to the binary expansion of $\{\lfloor \frac{2^n}{2q} \rfloor \cdot 2q, \lfloor \frac{2^n}{2q} \rfloor \cdot 2q + 1, \dots, \lfloor \frac{2^n}{2q} \rfloor \cdot 2q + q - 1\}$ and Algorithm 2 outputs i on the nodes that correspond to the numbers $\{\lfloor \frac{2^n}{2q} \rfloor \cdot 2q + s_{i-1}, \dots, \lfloor \frac{2^n}{2q} \rfloor \cdot 2q + s_i - 1\}$.

Therefore we get that $\mathbb{P}[\mathcal{O}(\mathcal{A}) = i \mid \mathcal{T}(\mathcal{A}) = n] = \frac{1}{q} \cdot q \frac{d_i}{q_i} = d_i$ which implies that $\mathcal{O}(\mathcal{A})$ is \mathcal{D} when conditioning on $\mathcal{T}(\mathcal{A}) = n$. Thus $\mathcal{O}(\mathcal{A})$ and $\mathcal{T}(\mathcal{A})$ are independent random variables.

◁

To demonstrate how the algorithm works, we show an example of the generating tree for the distribution: $\mathcal{D}_0 = \{\frac{1}{2}, \frac{1}{3}, \frac{1}{6}\}$.

► **Example 2.10.** The LCM q of the denominators of \mathcal{D}_0 is 6 and so:

$$\mathcal{D}_0 = \left\{ \frac{3}{6}, \frac{2}{6}, \frac{1}{6} \right\}$$

This gives the following distribution generating tree:

2.3 Randomness Efficiency

We now show the sampling algorithm is the most efficient one in terms of randomness. To formalize what it means to be most efficient in terms of randomness, for an algorithm \mathcal{A} let $\ell_i(\mathcal{A})$ be the probability that \mathcal{A} produces an output on level i and denote by $S_k = \sum_{i=1}^k \ell_i(\mathcal{A})$, the probability that the algorithm will produce an output up until, and including, level k .

► **Definition 2.11.** Let \mathcal{A} and \mathcal{A}' be two algorithms with the same output distribution \mathcal{D} . We say \mathcal{A} (weakly) dominates \mathcal{A}' in efficiency if for all k : $S_k(\mathcal{A}') \leq S_k(\mathcal{A})$. Furthermore, we say \mathcal{A} strictly dominates \mathcal{A}' if \mathcal{A} weakly dominates \mathcal{A}' and there exists some k for which $S_k(\mathcal{A}') < S_k(\mathcal{A})$.

With this definition we can now define what it means for an algorithm to be “optimal”:

► **Definition 2.12.** An algorithm \mathcal{A} that generates \mathcal{D} is **optimal**, if for all algorithms \mathcal{A}' that generate \mathcal{D} , \mathcal{A} (weakly) dominates \mathcal{A}' .

► **Remark 2.13.** For non-negative integer valued random variable X : $\mathbb{E}[X] = \sum_{k=1}^{\infty} \mathbb{P}[X \geq k]$ and therefore if \mathcal{A} weakly dominates \mathcal{A}' then $\mathbb{E}[\mathcal{T}(\mathcal{A})] \leq \mathbb{E}[\mathcal{T}(\mathcal{A}')]$ which means that if \mathcal{A} is optimal then the expected number of random bits used by \mathcal{A} is the minimum possible.

11:10 Resistance to Timing Attacks for Sampling and Privacy Preserving Schemes

We will show that Algorithm 1 is optimal. First we need:

► **Observation 2.14.** *Let t_k be the number of unreachable or output nodes level k , then $S_k = \frac{t_k}{2^k}$.*

Proof. Proof in Appendix B. ◀

► **Theorem 2.15.** *Let \mathcal{D} be a finite and rational distribution where the LCM of the denominators of the probabilities is q , then Algorithm 1 is an optimal time oblivious algorithm for generating \mathcal{D} .*

Proof. Proof in Appendix B. ◀

In addition to its optimality Algorithm 1 is essentially unique in the following sense:

▷ **Claim 2.16.** All optimal algorithms have the same number of output nodes in each level as Algorithm 1.

Proof. Let \mathcal{A} be Algorithm 1 and let \mathcal{A}' be another optimal time oblivious algorithm for \mathcal{D} . Since \mathcal{A} and \mathcal{A}' are both optimal we have: $\forall k : S_k(\mathcal{A}) = S_k(\mathcal{A}')$. By definition of S_k this implies $\forall k : \ell_k(\mathcal{A}) = \ell_k(\mathcal{A}')$. Therefore \mathcal{A} and \mathcal{A}' have the same output levels with q output nodes at each output level. ◀

We can now estimate the number of random bits Algorithm 1 reads from the tape:

► **Lemma 2.17.** *The expected number of bits read from the tape, $\mathbb{E}[\mathcal{T}(\mathcal{A})]$, by Algorithm 1 is $\log_2 q + \Theta(1)$.*

Proof. Proof in Appendix B. ◀

We now analyze the complexity of Algorithm 1. Recall that $n = |\text{supp}(\mathcal{D})|$. We assume the input is a list of numbers (p_i, q_i) , the LCM q and the size of the support $n = |\text{supp}(\mathcal{D})|$:

Preparation time worst case $O(n)$. The array of s_i described in Algorithm 2 can be computed ahead of time in $O(n)$ running time by iteratively applying $s_i = s_{i-1} + q \cdot \frac{p_i}{q_i}$.

Sampling time expected $O(\log q)$. Remember that from Lemma 2.17 we know that the expected number of bits the algorithm reads is $\log_2 q + \Theta(1)$ bits. The algorithm takes extra $O(\log q)$ time to binary search what to output in Algorithm 2 so the sampling time is $O(\log q)$. One should keep in mind that the first level that is an output level is $\lceil \log_2 q \rceil$, and therefore the loop in Algorithm 1 may start at that level. Similar analysis to Lemma 2.17 implies that the expected number of iterations executed by the loop is $\Theta(1)$.

Space complexity worst case $O(\log q)$. We can deduce $O(\log q)$ space complexity in expectation since $\mathcal{T}(\mathcal{A})$, the expected number of bits read from the tape, is $\log_2 q + O(1)$ and the memory needed for $\mathcal{R}_n, 2^n$ is proportional to $\mathcal{T}(\mathcal{A})$. To get $O(\log q)$ worst case, notice that the algorithm doesn't actually need to know $\mathcal{R}_n, 2^n$ but only $(\mathcal{R}_n \bmod 2q), (2^n \bmod 2q)$ and this can be maintained using $O(\log q)$ bits.

Algorithm 1 is also oblivious in the much stronger sense: all reachable nodes have the same control-flow i.e. they execute each line in the algorithm exactly the same number of times, and in the same order. Therefore, given a fixed time implementation of $\bmod 2q$, and a fixed time implementation that compares two numbers of size up to $2q$ the running time of this implementation of \mathcal{A} will not leak information on the output⁴.

⁴ Fixed time mod operations are required since in some circumstance even having the same control flow does not guarantee fixed time, as was shown by the Hertzbleed attack [24]

Notice that the parameters above cannot be improved. Preparation time is essential for general \mathcal{D} , since any sampling algorithm must read the probabilities of the outputs. The sampling time cannot be improved by Lemma 2.17, it also cannot be worst case by Corollary 2.7. Space complexity $\Omega(\log q)$ is needed to represent output range of size $\Omega(q)$.

From the discussion in this section, we conclude that defending against time attacks may cost unbounded slowdown, i.e.:

► **Observation 2.18.** *For every n there exist a distribution \mathcal{D} for which $\mathbb{E}[\mathcal{T}(\mathcal{A})]/\mathbb{E}[\mathcal{T}(\mathcal{A}')] \geq n$ where \mathcal{A}' is the optimal generation algorithm for \mathcal{D} (not necessarily time oblivious) and \mathcal{A} is any time oblivious algorithm for \mathcal{D} .*

Proof. Consider the distribution \mathcal{D} of a biased coin with $p = \frac{1}{2^{2n}}$. Since the separating bit algorithm takes 2 random bits in expectation we know that $\mathbb{E}[\mathcal{T}(\mathcal{A}')] \leq 2$ since \mathcal{A}' is optimal.

Let \mathcal{A}'' be Algorithm 1. Since it is optimal, we deduce from Remark 2.13 that $\mathbb{E}[\mathcal{T}(\mathcal{A})] \geq \mathbb{E}[\mathcal{T}(\mathcal{A}'')]$. \mathcal{A}'' has one output level which is $2n$ and therefore $\mathbb{E}[\mathcal{T}(\mathcal{A})] \geq 2n$. By dividing these two inequalities we conclude that $\mathbb{E}[\mathcal{T}(\mathcal{A})]/\mathbb{E}[\mathcal{T}(\mathcal{A}')] \geq n$. ◀

We refer the interested reader to Appendix A where we consider time oblivious randomized algorithms that does not leak information on their input, as well as their input from the running time. We show that the algorithm of Karp and Luby [17, 18] to sample a satisfying assignment from a DNF formula, can be transformed to be time oblivious and efficient, but there is no efficient time oblivious algorithm that does not leak x from the running time.

2.4 Approximate Time Oblivious Sampling a Distribution

The “time oblivious” condition may sometimes be too strict, since many distributions used in real life applications do not have a finite support or rational probabilities. Moreover, even if the distribution does have finite support and rational probabilities, in some situations Algorithm 1 can be considerably slower than the Knuth-Yao sampler (that is optimal among all samplers in the random coin flips model). In other situations the time oblivious sampler has similar number of coin flips used to the Knuth-Yao sampler, e.g. when sampling a uniform integer in $[n]$. Therefore it is desired to have a definition of “approximate obliviousness.”

We stress that whether using an approximation is an appropriate solution depends on the specific application of the sample in a randomized algorithm as well as on the approximation’s guarantee. For example, if \mathcal{M} is a pure DP mechanism, and the sampling time used by \mathcal{M} leaks a small amount of information on the database, then \mathcal{M} may not be pure DP given the running time. On the other hand this sort of approximation can be applied to an approximate DP mechanism with a small cost to δ . We will consider time oblivious DP mechanism in more detail in Section 3.

We suggest the following definition for “approximate time oblivious sampling” that preserves the privacy of the sample even when the running time has leaked. Note that for sampling algorithm \mathcal{A} we let $\mathcal{O}(\mathcal{A})$ be its output. The definition essentially says that given the running time the conditional distribution is close to the original in a point-wise sense.

► **Definition 2.19.** *Let \mathcal{D} be a distribution and $X \sim \mathcal{D}$. An algorithm \mathcal{A} is (ϵ, δ) -approximate time oblivious sampler of \mathcal{D} if for any $T \subseteq \mathbb{N}$ and for any $S \subseteq \text{supp}(\mathcal{D})$:*

$$e^{-\epsilon} \mathbb{P}[X \in S] - \delta \leq \mathbb{P}[\mathcal{O}(\mathcal{A}) \in S \mid \mathcal{T}(\mathcal{A}) \in T] \leq e^{\epsilon} \mathbb{P}[X \in S] + \delta.$$

The main benefit of considering the approximation above is that, as we shall see, the sampling complexity will *not* depend on the LCM of the distribution. If $\delta = 0$ then we say \mathcal{A} is a pure time oblivious sampler of \mathcal{D} and if $\delta > 0$ then it is an approximate time oblivious sampler. Allowing (pure) type of approximation yield that all distributions of finite support can be sampled approximately:

11:12 Resistance to Timing Attacks for Sampling and Privacy Preserving Schemes

▷ **Claim 2.20.** Let \mathcal{D} be a discrete distribution with support size n and let

$$H := H(\mathcal{D}) = \max_{x \in \text{supp}(\mathcal{D})} \log \left(\frac{1}{\mathbb{P}[x]} \right).$$

Then there exist an ε -pure time oblivious sampler of \mathcal{D} that uses $H + \log \frac{1}{\varepsilon} + O(1)$ random bits in the worst case.

Proof. Proof in Appendix B ◁

We note that in the pure variant of Definition 2.19, only distributions of finite support can be sampled, as $\text{supp}(\mathcal{D})$ must contain all the outputs in the first output level of \mathcal{A} . We will use this observation again in Section 3. We also note that the number of coin flips used is essentially tight, since a pure sampler that outputs after k coin tosses must satisfy $He^\varepsilon \geq 2^k$, otherwise the maximizer of H would not satisfy the pure sampling condition.

We can replace of the dependence on $H(\mathcal{D})$, by a dependence on $\delta > 0$ if we allow the use of an (ε, δ) -approximate time oblivious sampler:

▷ **Claim 2.21.** Let \mathcal{D} be a discrete distribution with $\text{supp}(\mathcal{D}) = [n]$, then there exist a $(0, \delta)$ -approximate time oblivious sampler that uses $\log n + \log \frac{1}{\delta} + O(1)$ bits worst case.

Proof. Proof in Appendix B. ◁

Many distributions used in applications are of infinite support, e.g. the geometric distribution, discrete Laplace, etc. With respect to Definition 2.19 one must consider an approximate time oblivious sampler (that is not pure). Notice that a consequence of Definition 2.19 is that every subset $S \subseteq \text{supp}(\mathcal{D})$ of measure $> \delta$ must satisfy that $\text{supp}(\mathcal{O}(\mathcal{A})) \cap S \neq \emptyset$. This suggests that the the support of the distribution generated by the sampler should be of measure at least $1 - \delta$.

In general consider a distribution \mathcal{D} and let $S_\delta \subseteq \text{supp}(\mathcal{D})$ denote a subset of minimal size that satisfies for $X \sim \mathcal{D}$: $\mathbb{P}[X \in S_\delta] \geq 1 - \delta$.

▷ **Claim 2.22.** For any distribution \mathcal{D} there exist an $(0, \delta)$ approximate time oblivious sampler that uses $\log(|S_{\delta/2}|) + \log \frac{1}{\delta} + O(1)$ random bits in the worst case.

Proof. To ease the notation we assume that $\text{supp}(\mathcal{D}) = \mathbb{N}$, X is a random variable with distribution \mathcal{D} , and $p_i = \mathbb{P}[X = i]$. Also assume $S_{\delta/2} = [n]$ for $n = |S_{\delta/2}|$. Let \mathcal{D}' be the distribution supported on $[n]$ in which the probability to get i is q_i where:

$$q_i = \begin{cases} p_i + \mathbb{P}[X \notin [n]], & \text{if } i = 1 \\ p_i, & \text{otherwise} \end{cases}$$

By applying the sampler of Claim 2.21 to \mathcal{D}' we obtain a sampler that uses $\log n + \log \frac{1}{\delta} + O(1)$ random coins. Notice that since the time is fixed and $\varepsilon = 0$ then Definition 2.19 coincides with total variation (TV) distance of distributions. Therefore since by construction \mathcal{D} and \mathcal{D}' are of TV distance $\frac{\delta}{2}$ and \mathcal{D}' and the distribution produced by Claim 2.21 is of distance $\frac{\delta}{2}$ the sampler distribution is of TV distance δ and it outputs on a single time, therefore satisfies Definition 2.19. ◁

We remark that for distributions of infinite support tail bounds imply bounds on S_δ and therefore imply bounds on the efficiency of an approximate time oblivious sampler. For example, let \mathcal{D} be a discrete distribution on \mathbb{Z} satisfying $\mathbb{P}[|X| > t] \leq c_1 \exp(c_2 t)$ for some positive constants c_1, c_2 and t large enough (i.e. \mathcal{D} is discrete, sub-exponential distribution).

The tail bound implies that $|S_\delta| = O(\log \frac{1}{\delta})$ and thus the approximate sampler is efficient and needs $O(\log \frac{1}{\delta})$ random bits by Claim 2.22.

This in particular applies to the sub-exponential distributions such as geometric and Laplace which are commonly used in differential privacy.

3 Differential Privacy Mechanisms and Timing Attacks

We gave a full characterization on the distributions that can be sampled in a time oblivious way. This can be interpreted as a negative result for time oblivious sampling, since many algorithms use distributions that are not of finite support and also assume that elements may have irrational probabilities of being an outcome. A prominent example in the context of differential privacy (DP) is the application of a $Laplace(\epsilon)$ R.V. that has an infinite support. It is often sampled inside a differentially private mechanism, but by Theorem 2.3 it cannot be sampled by a time oblivious algorithm. Therefore the execution of the mechanism will actually leak information that will make it non-DP (at least in the *pure* sense).

This may be considered as the conceptual starting point of the work of Balcer and Vadhan [2]. In their work, they addressed the problem of DP mechanisms implemented on “finite computers” to address the issues of timing attacks and infinite output range. They constructed DP algorithms on finite range, with worst case running time to handle those issues and emphasized that each sampling must be of a distribution \mathcal{D} which is finite an rational.

One issue that motivates the formal discussion of the sampler as part of the timing of a mechanism is that by Corollary 2.7 we get that rounding to rationals, as done in [2], isn’t enough and the rounding must be to *dyadic* rationals; this was first observed in the work of [8] in relation to DP. This issue is clearly dependent on the model of computation, but shows the delicacy needed in the rounding procedure to ensure worst case running time.

Another issue is that the sampling running time of an algorithm is affected by the running time of the sampler it uses. By Observation 2.18 it may be that a proposed algorithm originally uses samples that take expected $O(1)$ time to generate, but using the time oblivious samples must run in running time $\omega(1)$. This may affect total performance of an algorithm.

The main aim of this section is to define time oblivious DP mechanisms and specifically, time oblivious pure DP mechanisms and investigate their properties. Even though time oblivious sampling is restrictive, we show that time oblivious pure DP mechanism are far more flexible. In particular, we prove that any pureDP mechanism can be transformed to a time oblivious with almost the same guarantees. We begin with the definition of time oblivious DP mechanism.

► **Definition 3.1.** *Let $\mathcal{M}: \mathcal{C} \rightarrow \mathcal{R}$ be a randomized algorithm. We say \mathcal{M} is (ϵ, δ) -differentially private time oblivious mechanism if for every pair of neighboring datasets D and D' , every subset $S \subseteq \mathcal{R} \times \mathbb{N}$*

$$\mathbb{P}[(\mathcal{M}(D), \mathcal{T}(\mathcal{M}(D))) \in S] \leq e^\epsilon \cdot \mathbb{P}[(\mathcal{M}(D'), \mathcal{T}(\mathcal{M}(D'))) \in S] + \delta.$$

In the definition above, if $\delta = 0$ the mechanism is said to be *pure* time oblivious DP, otherwise it is approximate time oblivious DP. \mathcal{C} denotes the set of possible database (rather than a distribution), and we assume that \mathcal{C} is connected as a graph with respect to the neighbouring relation. The time oblivious DP condition does not impose any condition on $\mathcal{M}(D)$ in terms of finiteness or rationality. The time oblivious DP condition means that levels in the distribution generating trees from Section 2.2 satisfy that the conditional distribution on neighbouring datasets are almost the same up to the prescribed parameters.

11:14 Resistance to Timing Attacks for Sampling and Privacy Preserving Schemes

The following lemma says that the time oblivious condition implies that all databases have the same outputs at each level.

► **Lemma 3.2.** *Let \mathcal{M} be a time oblivious ε -pure DP mechanism on a connected set of databases, then for any two databases D and D'' :*

$$\text{supp}(\mathcal{M}(D)|\mathcal{T}(\mathcal{M}(D)) = t) = \text{supp}(\mathcal{M}(D'')|\mathcal{T}(\mathcal{M}(D'')) = t).$$

Proof. By the connectivity of \mathcal{C} it's enough to show this for neighbouring databases D, D' . If $r \in \mathcal{R}$ is in the support of the conditional distribution on level n of \mathcal{M} on D then by the pure DP condition:

$$0 < \mathbb{P}[\mathcal{M}(D) = r \mid \mathcal{T}(\mathcal{M}(D)) = t] \leq e^\varepsilon \cdot \mathbb{P}[\mathcal{M}(D') = r \mid \mathcal{T}(\mathcal{M}(D')) = t]$$

Therefore $\mathbb{P}[\mathcal{M}(D') = r \mid \mathcal{T}(\mathcal{M}(D')) = t] > 0$ and r is in the support of the conditional distribution on level t of $\mathcal{M}(D')$. ◀

Pure DP mechanism for approximate counting

One fundamental problem considered in the DP literature is counting. That is, given a database D output an approximation to the number of elements satisfying some property. We now devise a time oblivious pure DP mechanism for approximate counting up to a factor 2. This example captures critical properties of time oblivious DP algorithms.

Let $c(D)$ be the true count of D . For each database we need to construct a tree and by Lemma 3.2 at each level we must have the same outputs for all possible databases D . The mechanism will have at level $i + 2$ only one possible output which is 2^i . If $2^k \leq c(D) < 2^{k+1}$ the mechanism will have one output node in each of the levels $2, \dots, k + 1, k + 4, k + 5, \dots$, at level $k + 2$ the algorithm outputs on $2^{k+1} - c(D) + 1$ of the nodes the value 2^k , while at level $k + 3$ the algorithm outputs 2^{k+1} on $2c(D) - 2^{k+1} + 1$ nodes. Notice that

$$\sum_{i=2}^{k+1} 2^{-i} + (2^{k+1} - c(D) + 1)2^{-(k+2)} + (2c(D) - 2^{k+1} + 1)2^{-(k+3)} + \sum_{i=k+4}^{\infty} 2^{-i} = \frac{3}{4}.$$

Therefore by adding another output at the 2^{nd} level which will always be 1 we get a probability distribution on output nodes and with probability at least $\frac{1}{4}$ the output is either 2^k or 2^{k+1} so this algorithm approximates $c(D)$ up to factor 2 with constant probability. Moreover, notice that neighboring databases have identical trees up to 1 node in one level $k + 2$ and 2 nodes in level $k + 3$. This algorithm is $\ln 3$ pure DP since at each output level the number of output nodes changes by at most 2 and since there is at least one output node of value 2^i at level $i + 2$ we get that if there are s output nodes at the level then it's enough to have ε satisfy $s + 2 \leq e^\varepsilon s$ which holds for $\varepsilon = \ln 3$ since $s \geq 1$.

The mechanism above has several drawbacks, which we will soon show must exist in all time oblivious pure DP mechanisms. One drawback is that with constant probability the mechanism outputs an irrelevant output since with probability $\frac{1}{4}$ the output is 1 for any database. Another drawback is that the amount of randomness used scales with $c(D)$ in contrast to the not time oblivious algorithm of adding a discrete Laplace noise to $c(D)$.

We turn to specifying and proving the claims stated above. We begin with showing that a time oblivious pure DP mechanism on infinitely many databases gives an irrelevant output with some constant probability. To define what this means we assume we have a *utility function* $\mathbf{U}: \mathcal{C} \times \mathcal{R} \rightarrow \{0, 1\}$ for which $\mathbf{U}(D, r) = 1$ if r is useful information of D . We say that \mathbf{U} is *sofic* if for any $r \in \mathcal{R}$ there are only finitely many databases D such that $\mathbf{U}(D, r) = 1$.

► **Definition 3.3.** For a mechanism \mathcal{M} the utility of the mechanism on a database D is defined by $\mathbf{U}_{\mathcal{M}}(D) = \mathbb{E}[\mathbf{U}(\mathcal{M}(D))]$, we omit \mathcal{M} when it is clear from context. The guaranteed utility of a mechanism \mathcal{M} is $G_{\mathbf{U}}(\mathcal{M}) := \inf_{D \in \mathcal{C}} \mathbf{U}(D)$.

We show that with some constant probability the algorithm gives an irrelevant answer for most databases using the definition of guaranteed utility.

▷ **Claim 3.4.** Let \mathbf{U} be a sofic utility function. Then any time oblivious pure DP mechanism \mathcal{M} with infinitely many possible databases has $G_{\mathbf{U}}(\mathcal{M}) \leq 1 - \frac{1}{2^s}$ where s is the first output level of \mathcal{M} .

Proof. Proof in Appendix B. ◁

Similarly to guaranteed utility, we define the guaranteed expected runtime of the algorithm to be $G_{\mathcal{T}}(\mathcal{M}) = \sup_{D \in \mathcal{C}} \mathbb{E}[\mathcal{T}(\mathcal{M}(D))]$. Even though it seems natural to require for a time oblivious ε -pure DP to have a $G_{\mathcal{T}}(\mathcal{M})$ finite (with some parameter that may depend on ε) it cannot be done without serious consequences on the utility of the mechanism.

► **Proposition 3.5.** Let \mathbf{U} be a sofic utility function and \mathcal{M} is a pure DP mechanism on infinitely many possible databases. Then $G_{\mathcal{T}}(\mathcal{M})$ is finite implies that $G_{\mathbf{U}}(\mathcal{M}) = 0$

Proof. Proof in Appendix B. ◀

We remark that all proofs above can be easily modified to handle a utility function with range $[0, 1]$ such that for every $r \in \mathcal{R}$ and $x > 0$ there are finitely many D with $\mathbf{U}(r, D) > x$.

It is natural to wonder how does the expressive power of pure DP mechanisms compare to the expressive power of time oblivious pure DP mechanisms, i.e. does a ε pure DP algorithm has a time oblivious counterpart?

We now get to the main result of this section. We prove that *any* ε -pure DP mechanism \mathcal{M} has a time oblivious pure DP mechanism with almost the same security guarantee and utility. Therefore this shows that pure DP mechanisms *can be defended against timing attacks* with a small cost to their utility and and security guarantees.

► **Theorem 3.6.** Let \mathcal{M} be any ε -pure DP mechanism with a discrete range \mathcal{R} . For any $\gamma > 0$, $\varepsilon' > \varepsilon$ there is a time oblivious ε' pure DP mechanism \mathcal{M}_{obl} such that $\|\mathcal{M}_{obl}(D) - \mathcal{M}(D)\|_{TV} < \gamma$.

Proof. The idea is to use a similar construction to the approximate counting mechanism described above by providing that each level in the tree will correspond to a different element in \mathcal{R} . Intuitively we will view the levels of the tree as an “abacus”, and the change of the probability distributions of neighbouring datasets will correspond to change in the number of beads that outputs in each layer.

We assume $\mathcal{R} = \{r_i\}_{i=1}^{\infty}$, and we start by picking s , the depth of the first input level which we will decide later, and an integer t which corresponds to the number of output nodes that must be in every level.

The element r_i will be output at level $s+i$. For the mechanism \mathcal{M}_{obl} to have outputs close to $\mathcal{M}(D)$ for any D we need that the probability to output at the level that corresponds to r_i is roughly $p_i = \mathbb{P}[\mathcal{M}(D) = r_i]$; let q_i denote the probability to output r_i in $\mathcal{M}_{obl}(D)$. We pick $q_i \approx \frac{t}{2^{s+i}} + \frac{2^s - t}{2^s} p_i$. Define $N_i^+ = \lceil (2^{s+i} - t2^i)p_i \rceil + t$ and $N_i^- = \lfloor (2^{s+i} - t2^i)p_i \rfloor + t - 1$ to be the number of output nodes in level $s+i$ with the two options for estimating q_i from below and from above. Notice that by the choice of N^+ and N^- we know that

$$1 + \frac{1}{2^s} \geq \sum 2^{-(s+i)} N_i^+ \geq 1 \geq \sum 2^{-(s+i)} N_i^- \geq 1 - \frac{1}{2^s}.$$

11:16 Resistance to Timing Attacks for Sampling and Privacy Preserving Schemes

Therefore, there exists a signing of the N_i (this is done by looking at the binary expansion of the sum above) that gives a probability distribution. Notice that the following inequalities hold

$$\frac{N_i^+}{2^{s+i}} \geq \mathbb{P}[\mathcal{M}_{obl}(D) = r_i] \geq \frac{N_i^-}{2^{s+i}}.$$

To bound the privacy guarantee of \mathcal{M}_{obl} we need to bound $N_i^+(D)/N_i^-(D')$ for neighboring database D, D' , where we abuse the notation to describe which N_i belongs to which database. But since \mathcal{M} is ε -pure DP we get that $N_i^+(D)/N_i^-(D') \leq \frac{te^\varepsilon+1}{t-1}$. Therefore by picking t large enough such that $\frac{te^\varepsilon+1}{t-1} \leq e^{\varepsilon'}$ we can ensure the DP guarantee and by picking s large enough that implies $\frac{t+1}{2^s} < \gamma$ (The extra masking of t nodes at each output level $+ 1$ for rounding) the proof follows. \blacktriangleleft

► **Remark 3.7.** Notice that since $\|\mathcal{M}_{obl}(D) - \mathcal{M}(D)\|_{TV} < \gamma$ it follows that for any utility function (not necessarily sofic) $\mathbf{U}: \mathcal{C} \times \mathcal{R} \rightarrow [0, 1]$, the utility of $\mathbf{U}_{\mathcal{M}_{obl}}(D) \geq \mathbf{U}_{\mathcal{M}}(D) - \gamma$.

The proof above produces a tree for $\mathcal{M}_{obl}(D)$ each $D \in \mathcal{C}$ by picking the signing for the N_i . We leave it as an open question to provide mechanisms $\mathcal{M}(D)$ for which the time oblivious pure DP mechanism $\mathcal{M}(D)$ is efficiently computable from the input D .

By Theorem 3.6 we get that any pure DP mechanism with a discrete range has a time oblivious pure DP with mechanisms with similar privacy guarantee. We saw in Proposition 3.5 that infinite guaranteed expected running time cannot be avoided for pure DP mechanism. Approximate DP mechanisms do not suffer from this issues.

▷ **Claim 3.8.** There is a $(0, \delta)$ mechanism for approximate counting up to an additive factor $\frac{1}{\delta}$ with expected number of used bits $O(\log \frac{1}{\delta})$.

Proof. Return the true count $+ a$ uniform number in $[1, \dots, \frac{1}{\delta}]$, with the uniform number sampled by Algorithm 1. \blacktriangleleft

4 Future Work and Open Problems

Our investigation focused on the question of time oblivious sampling, and we used a combinatorial lens to give a full characterization of the distributions that can be sampled from in a time oblivious way. These distributions have a finite support and rational weights. The combinatorial perspective also helped us define an algorithm to sample from such distributions and show it is optimal in a strong sense of the randomness used. In certain situations distributions are not given explicitly but are given in a rather succinct form. A notable example is sampling via a Markov Chain process which have found numerous applications in TCS. This lead us to ask:

► **Question 1.** *Is it possible to efficiently convert a Markov Chain algorithm, which samples from a finite and rational distribution, into a time oblivious algorithm with exactly the same output distribution?*

A natural consumer of our results is the area of differential privacy. Here we had both bad news and good news: the bad news are for mechanisms that we require to produce useful results (where the utility is some arbitrary but non trivial function of a points in the domain and range). It is impossible to guarantee usefulness with high probability ($(1 - g(n))$ where g is a function whose limit is 0). On the other hand we argued that it is possible to take any mechanisms and make it time oblivious without changing the distribution by much (and hence its utility) while preserving its differential privacy.

► **Question 2.** *One important question is whether the process of turning a DP mechanism into a timing oblivious one can preserve the computational efficiency of the scheme.*

In this paper we revisited the issues arising from timing attacks and specifically investigated Time oblivious sampling, where the emphasis was an exactly producing a distribution. In a companion paper we explore notions and methods for protecting keyed functions from timing attacks [6].

References

- 1 Marc Andryscio, David Kohlbrenner, Keaton Mowery, Ranjit Jhala, Sorin Lerner, and Hovav Shacham. On subnormal floating point and abnormal timing. In *2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, May 17-21, 2015*, pages 623–639. IEEE Computer Society, 2015. doi:10.1109/SP.2015.44.
- 2 Victor Balcer and Salil P. Vadhan. Differential privacy on finite computers. In Anna R. Karlin, editor, *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, volume 94 of *LIPICs*, pages 43:1–43:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPICs.ITCS.2018.43.
- 3 Jonathan Bootle, Claire Delaplace, Thomas Espitau, Pierre-Alain Fouque, and Mehdi Tibouchi. LWE without modular reduction and improved side-channel attacks against BLISS. In *Advances in Cryptology – ASIACRYPT 2018, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part I*, volume 11272 of *Lecture Notes in Computer Science*, pages 494–524. Springer, 2018. doi:10.1007/978-3-030-03326-2_17.
- 4 David Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *Advances in Cryptology: Proceedings of CRYPTO '82, Santa Barbara, California, USA, August 23-25, 1982*, pages 199–203. Plenum Press, New York, 1982. doi:10.1007/978-1-4757-0602-4_18.
- 5 Luc Devroye. *Non-Uniform Random Variate Generation*. Springer, 1986. doi:10.1007/978-1-4613-8643-8.
- 6 Yoav Ben Dov, Liron David, Moni Naor, and Elad Tzalik. Resistance to timing attacks revisited: Protecting the keys. Technical report, Weizmann Institute of Science, Rehovot, 2023.
- 7 Léo Ducas, Alain Durmus, Tancrede Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal gaussians. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013. Proceedings, Part I*, volume 8042 of *Lecture Notes in Computer Science*, pages 40–56. Springer, 2013. doi:10.1007/978-3-642-40041-4_3.
- 8 Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology – EUROCRYPT 2006, St. Petersburg, Russia, May 28 – June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 486–503. Springer, 2006. doi:10.1007/11761679_29.
- 9 Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, volume 3876 of *Lecture Notes in Computer Science*, pages 265–284. Springer, 2006. doi:10.1007/11681878_14.
- 10 Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014. doi:10.1561/04000000042.
- 11 Thomas Espitau, Pierre-Alain Fouque, Benoît Gérard, and Mehdi Tibouchi. Side-channel attacks on BLISS lattice-based signatures: Exploiting branch tracing against strongswan and electromagnetic emanations in microcontrollers. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 – November 03, 2017*, pages 1857–1874. ACM, 2017. doi:10.1145/3133956.3134028.

- 12 David Feldman, Russell Impagliazzo, Moni Naor, Noam Nisan, Steven Rudich, and Adi Shamir. On dice and coins: Models of computation for random generation. *Inf. Comput.*, 104(2):159–174, 1993. doi:10.1006/inco.1993.1028.
- 13 Christina Ilvento. Implementing the exponential mechanism with base-2 differential privacy. In *CCS '20: 2020 ACM SIGSAC Conference on Computer and Communications Security, November 9-13, 2020*, pages 717–742. ACM, 2020. doi:10.1145/3372297.3417269.
- 14 Yuval Ishai, Amit Sahai, and David A. Wagner. Private circuits: Securing hardware against probing attacks. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003 Proceedings*, volume 2729 of *Lecture Notes in Computer Science*, pages 463–481. Springer, 2003. doi:10.1007/978-3-540-45146-4_27.
- 15 Mark Jerrum, Leslie G. Valiant, and Vijay V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theor. Comput. Sci.*, 43:169–188, 1986. doi:10.1016/0304-3975(86)90174-X.
- 16 Yael Tauman Kalai and Leonid Reyzin. A survey of leakage-resilient cryptography. In Oded Goldreich, editor, *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*, pages 727–794. ACM, 2019. doi:10.1145/3335741.3335768.
- 17 Richard M. Karp and Michael Luby. Monte-carlo algorithms for enumeration and reliability problems. In *24th Annual Symposium on Foundations of Computer Science, Tucson, Arizona, USA, 7-9 November 1983*, pages 56–64. IEEE Computer Society, 1983. doi:10.1109/SFCS.1983.35.
- 18 Richard M. Karp, Michael Luby, and Neal Madras. Monte-carlo approximation algorithms for enumeration problems. *J. Algorithms*, 10(3):429–448, 1989. doi:10.1016/0196-6774(89)90038-2.
- 19 Donald E. Knuth and Andrew C. Yao. The complexity of nonuniform random number generation. *Algorithms and Complexity: New Directions and Recent Results*, edited by J.F. Traub, 1976.
- 20 Paul C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *Advances in Cryptology – CRYPTO '96, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer, 1996. doi:10.1007/3-540-68697-5_9.
- 21 Richard J. Lipton and Jeffrey F. Naughton. Clocked adversaries for hashing. *Algorithmica*, 9(3):239–252, 1993. doi:10.1007/BF01190898.
- 22 Ilya Mironov. On significance of the least significant bits for differential privacy. In *the ACM Conference on Computer and Communications Security, CCS'12, Raleigh, NC, USA, October 16-18, 2012*, pages 650–661. ACM, 2012. doi:10.1145/2382196.2382264.
- 23 Gerald Tenenbaum. *Introduction to analytic and probabilistic number theory*, volume 163 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, Rhode Island, third edition. edition, 2015.
- 24 Yingchen Wang, Riccardo Paccagnella, Elizabeth Tang He, Hovav Shacham, Christopher W Fletcher, and David Kohlbrenner. Hertzbleed: Turning power side-channel attacks into remote timing attacks on x86. In *Usenix Security '22: 31st USENIX Security Symposium 2022*, pages 679–697. Usenix, 2022.

A The Good News and the Bad News: DNF Sampling

Recall that a DNF formula on n variables is a formula of the form $C_1 \vee C_2 \vee \dots \vee C_m$ where each term C_i has $1 \leq \ell_i \leq n$ literals and is of the form $C_i = y_1^i \wedge y_2^i \wedge \dots \wedge y_{\ell_i}^i$ where each y_j^i is a literal. It is worth noting that the task of finding a satisfying assignment to a DNF formula is easy and straightforward, however counting the number of satisfying assignments is $\#P$ -complete. Jerrum, Valiant and Vazirani [15] provide a general discussion on generating a uniform object given an efficient algorithm that approximates to the counting problem.

Karp and Luby [17, 18] tackle the other direction. They describe an efficient algorithm for sampling a uniformly satisfying assignment to a DNF formula and use it for approximate counting.

Let $\phi = C_1 \vee \dots \vee C_m$ be a DNF formula with n variables, and let S_i be the set of satisfying assignments to term C_i . Let ℓ_i be the number of literals in C_i and note that $|S_i| = 2^{n-\ell_i}$. Further denote by $S = \sum_i |S_i|$ and by $p_i = \frac{|S_i|}{S}$, which is the relative weight of C_i . The algorithm works as follows:

1. Sample a clause with probability proportional to its weight, meaning clause C_i is chosen with probability p_i .
2. Sample a random assignment $\pi \sim S_i$.
3. Let k be the number of clauses that are satisfied by π . With probability $\frac{1}{k}$ return π , else restart the algorithm from the beginning.

In order to convert the algorithm to be time oblivious, for Step 1, notice that the running time may leak information about the chosen clause. The LCM q of the distribution in step 1 is $\sum_j |S_j|$. We can therefore use Algorithm 1 with $q = \sum_j |S_j|$ and conclude that Step 1 is done in a time oblivious way. Note that $\forall i: |S_i| \leq 2^n$ and there are m clauses, therefore the expected running time of this step is $O(n + \log m)$.

Next, the running time of Step 2 may leak information about the chosen assignment. If the clause has ℓ literals, one can simply choose a uniform assignment to the $n - \ell$ remaining literals. However, this does reveal what ℓ is, which yields information about what clause was chosen in the case where the clauses are of different sizes. To hide ℓ , we toss n coins and ignore the first ℓ , thus making Step 2 of the algorithm run in time independent of ℓ and the chosen assignment, making it time oblivious. This gives us a running time of n always.

Finally, for Step 3, we need to toss a $\frac{1}{k}$ -biased coin. To make this step time oblivious we need to hide the value of k , which is the number of satisfied clauses by the assignment π chosen in Step 2. Since there are m clauses, we know that $1 \leq k \leq m$. In order to hide what k is, we need to take a value q which is divisible by all numbers from 1 to m , i.e. $q = LCM(1, \dots, m)$. With this q we can run Algorithm 1 and output heads on the first $\frac{q}{k}$ output nodes, and tails on the rest. This makes the distribution of the running time of Step 3 independent of k , and overall Step 3 is time oblivious. The following is a standard number theoretic estimate (which may be found in [23] p.37):

► **Fact A.1.** $\ln(LCM(1, 2, \dots, n)) = \Theta(n)$.

By Lemma 2.17, we need $O(\log q) = \log(LCM(1, \dots, m))$ bits in expectation, instead of $O(\log m)$. By Fact A.1 we conclude that the expected running time of Step 3 is $O(m)$. Notice that this is an exponential increase relative to the $O(\log m)$ bits that is needed to generate a uniform element of $\{1, \dots, m\}$.

Finally the expected number of iterations of the algorithm before it produces an output is at most m , since k is bounded above by m . Putting it all together we get:

► **Proposition A.1.** *There exists an efficient time oblivious algorithm for uniform DNF sampling running in expected time $O(mn + m^2)$*

While we showed that the running time of the algorithm above does not leak information about what assignment was chosen, it does reveal information about the structure of the formula. For example, the running time leaks information about the number of literals and number of clauses in the formula. When the formula is public information this leakage does not give any new information, but there might be cases where we wish to hide the structure of the formula in addition to what assignment was chosen.

It is therefore natural to ask: is there an efficient time oblivious algorithm for DNF sampling that hides the formula as well as the chosen assignment?

A.1 An Exponential Lower Bound for Hiding the Formula

We formalise what it means to “hide the structure of the formula”:

► **Definition A.2.** Let $x \in \{0, 1\}^n$ be an input to a randomized algorithm \mathcal{A} . We say $\mathcal{A}(x)$ is an *input hiding time oblivious generating algorithm* if its running time distribution is independent of both the output distribution and the input x .

The motivation for the definition above is to defend against timing attacks in situations where a randomized algorithm runs on an input which we would like to be kept secret, as well as the output. We show that *DNF* sampling cannot be done efficiently in an input hiding time oblivious way, we will show a different problem which cannot be done efficiently, and then get the result on *DNF* via a reduction.

Consider the problem: given $x \in \{1, \dots, 2^n\}$ sample a random element in $\{1, \dots, x\}$. This task can be done efficiently using $O(\log x)$ random bits. However, in the input hiding setting we wish to hide x in addition to the sample. This problem cannot be solved efficiently and in a time oblivious way.

► **Lemma A.3.** The task of sampling a random element $k \in \{1, \dots, x\}$ given $x \in \{1, \dots, 2^n\}$ in an input hiding time oblivious way takes at least $\Omega(2^n)$ random bits.

Proof. Let $\ell_m(x = i)$ be the probability that the algorithm outputs on level m given $x = i$. In order for the sample to be time oblivious, we need the running time to be independent both of the choice of x and k . This means that the running time distribution of the algorithm should be the same for all values of x . In particular: for all m and i , $\ell_m(x = i) = \ell_m(x = 1)$

From the observation above we know that the algorithm must have the same output levels, and same number of output nodes for all values of x . Let q be the number of nodes in the first output level. We get that q must be divisible by all values from 1 to 2^n , and therefore $q \geq LCM(1, \dots, 2^n)$. By Fact A.1 we need $\Theta(2^n)$ to represent $LCM(1, 2, \dots, 2^n)$, and the result follows. ◀

Applying the lemma, we get the following theorem on *DNF* sampling:

► **Theorem A.4.** Sampling a uniform satisfying assignment of a *DNF* formula in an input hiding time oblivious way cannot be done efficiently and requires $\Omega(2^n)$ bits in expectation.

Proof. Given $x \in \{1, \dots, 2^n\}$ notice that we can write a formula with the literals a_0, \dots, a_{n-1} consisting of at most n clauses and x satisfying assignments. We denote the formula corresponding to x by ϕ_x . If $x = 2^k$ we set $\phi_x = \overline{a_0} \wedge \dots \wedge \overline{a_{(n-1)-k-1}} \wedge a_{n-1-k}$. It is easy to check that for $x = \sum_{i \in S} 2^i$ for some $S \subseteq \{0, 1, \dots, n-1\}$ one can set $\phi_x = \bigvee_{i \in S} \phi_{2^i}$ and obtain in such a way for each $x \in \{1, \dots, 2^n\}$ a formula with x satisfying assignments and at most n clauses.

We know by Lemma A.3 that to sample a uniform element from $\{1, \dots, x\}$ given x any algorithm must use $\Omega(2^n)$ bits in expectation in case we want to algorithm to be input hiding time oblivious case and by Proposition A.1 there is a time oblivious algorithm that samples from the formulas above using $O(n^2)$ random bits. Therefore we conclude that hiding the formula cannot be done efficiently. ◀

B Missing Proofs

B.1 Missing Proofs in Section 2

Proof of Lemma 2.6. If \mathcal{A} is time oblivious then the output distribution $\mathcal{O}(\mathcal{A})$ is \mathcal{D} when conditioning on $\mathcal{T}(\mathcal{A}) = k$, therefore: $\frac{p_j}{q_j} = d_j = \frac{e_j}{m_k}$ where m_k is the number of output nodes in the k^{th} level and e_j is the sequences of length k on which \mathcal{A} outputs j . We get that $e_j = m_k \cdot \frac{p_j}{q_j}$ and as each e_j is a natural number we get that for all j : $q_j \mid m_k$ and therefore $q \mid m_k$. This concludes Part 1.

For Part 2, notice that in the binary tree each output node in level l has 2^{k-l} descendants in level k that it makes unreachable, and also that an output node can not have another output node as a descendent, therefore $t_k = \sum_{i=0}^k 2^{k-i} m_i$. By the first part each m_i is divisible by q and therefore t_k is a multiple of q . ◀

Proof of Corollary 2.7. If $q = 2^k$ then we can clearly output all elements in level k since \mathcal{D} is rational with finite support. For the other direction, we may assume that the algorithm reads exactly t bits from \mathcal{R} . This also implies that \mathcal{D} is rational with finite support. By Lemma 2.6 $q \mid 2^t$ and therefore $q = 2^k$ for some k . ◀

Proof of Lemma 2.8. For Part 1, observe that the statement is true if $q = 2^m$ for some m . The test $2^n \geq q$ and the test $(2^n \bmod 2q) \geq q$ are exactly the same until the first output level. Since q is a power of 2 then all the nodes in level m will be output nodes and there will not be any more output nodes.

For q that is not a power of 2 we prove the claim by induction on the level. The base case is level 1. Indeed q is not a power of 2 which implies that $q > 2$ and as level 1 has 2 nodes, both nodes are reachable.

Assume the statement is true for all levels up to k . We now show the statement is true for $k + 1$. By the induction hypothesis let $r = 2^k \bmod 2q$ be the number of reachable nodes in level k . if $r < q$ we know that k is not an output level. In this case each reachable node $a_0 \dots a_{k-1}$ becomes two reachable nodes in level $k + 1$, since each node has two children: $a_0 \dots a_{k-1}0$ and $a_0 \dots a_{k-1}1$. From this we get that the number of reachable nodes in level $k + 1$ is $2r$, and the statement holds since:

$$2^{k+1} \equiv 2 \cdot 2^k \equiv 2r \pmod{2q}.$$

If $q < r < 2q$ then k is an output level. Write $r = q + s$ for some $1 \leq s < q$, and observe that q nodes will be output nodes in that level. This means that the number of reachable nodes in level $k + 1$ will be $2s$ and indeed:

$$2^{k+1} \equiv 2 \cdot 2^k \equiv 2r \equiv 2(q + s) \equiv 2q + 2s \equiv 2s \pmod{2q}.$$

This concludes part 1 of Lemma.

For Part 2 notice that at each output level the output nodes are the q leftmost reachable nodes, and we know there are exactly $2^n \bmod 2q$ reachable nodes from Part 1 the binary sequences of $\{\lfloor \frac{2^n}{2q} \rfloor \cdot 2q, \lfloor \frac{2^n}{2q} \rfloor \cdot 2q + 1, \dots, 2^n - 1\}$ correspond to the reachable nodes. Part 2 now follows by taking $\bmod 2q$ on the sequence above.

Finally, by Part 1 the number of reachable nodes in each level is between 0 and $2q - 1$, and in addition, if there are at least q reachable nodes, then q nodes produce an output and therefore $2^n - t_n < q$. ◀

11:22 Resistance to Timing Attacks for Sampling and Privacy Preserving Schemes

Proof of Observation 2.14. Recall that $S_k = \sum_{i=1}^k \ell_i(\mathcal{A})$ is the probability that the algorithm produced an output up until level and including k . By Part 2 of Lemma 2.6 we get:

$$S_k = \sum_{i=0}^k \ell_i(\mathcal{A}) = \sum_{i=0}^k 2^{-i} m_i = \frac{1}{2^k} \sum_{i=0}^k 2^{k-i} m_i = \frac{t_k}{2^k}. \quad \blacktriangleleft$$

Proof of Lemma 2.17. For the lower bound: the first output level k satisfies $2^k \geq q$ and $\mathcal{T}(\mathcal{A}) \geq k$ since it is the *first* output level. Taking expectation it follows that $\mathbb{E}[\mathcal{T}(\mathcal{A})] \geq \lceil \log_2 q \rceil$.

For the upper bound: Recall that by construction of Algorithm 1 the algorithm outputs on nodes from left to right, and at each level there are less than q nodes which are reachable but not output nodes. From this we get that after the first time the algorithm reads a 0, it will produce an output after at most $\log q$ more steps. Observe that the probability to read the first 0 in the i^{th} level is a *Geo* $(\frac{1}{2})$ R.V. and therefore we get:

$$\mathbb{E}[\mathcal{T}(\mathcal{A})] \leq \sum_{i=1}^{\infty} \frac{1}{2^i} (i + \log q) = \log q \sum_{i=1}^{\infty} \frac{1}{2^i} + \sum_{i=1}^{\infty} \frac{i}{2^i} = \log q + 2. \quad \blacktriangleleft$$

Proof of Theorem 2.15. Let \mathcal{D} be a distribution and \mathcal{A} be Algorithm 1 for \mathcal{D} . Assume towards a contradiction that \mathcal{A} is not optimal. This means that there exists a time oblivious algorithm \mathcal{A}' which generates \mathcal{D} and a k such that $S_k(\mathcal{A}') > S_k(\mathcal{A})$. By Observation 2.14 $S_k(\mathcal{A}') = \frac{t'_k}{2^k}$ and $S_k(\mathcal{A}) = \frac{t_k}{2^k}$. From the assumption about \mathcal{A}' we get that $t'_k > t_k$. From Lemma 2.6 we know that t'_k and t_k are both multiples of q which means $t'_k \geq t_k + q$. From Lemma 2.8 we know that $2^k - t_k < q$ which means that $t_k > 2^k - q$. Taking both inequalities into account we get that $t'_k > 2^k$. Recall that t'_k is the number of unreachable or output nodes of algorithm \mathcal{A}' at level k , and therefore cannot be bigger than 2^k . Thus we get a contradiction and conclude \mathcal{A} is optimal. \blacktriangleleft

Proof of Claim 2.20. We assume that $\text{supp}(\mathcal{D})$ is $[n]$ and define $p_i = \mathbb{P}[X = i]$. Let k be a parameter of the number of coins that the sampler uses before returning an answer, and assume $k > H + c$ for large enough c that will be picked later. Define $p_i^+ = \frac{\lceil p_i 2^k \rceil}{2^k}$ and $p_i^- = \frac{\lfloor p_i 2^k \rfloor}{2^k}$.

The sampler \mathcal{A} will have $\mathbb{P}[\mathcal{O}(\mathcal{A}) = i]$ is either p_i^+ or p_i^- and will always output after reading all k bits. Therefore it will satisfy that $\mathbb{P}[\mathcal{T}(\mathcal{A}) \in \{k\}] = 1$ which will make the conditioning on the time in Definition 2.19 redundant. It is possible to always “sign” p_i^\pm (i.e. pick between p_i^+ or p_i^-) to obtain a distribution (the sum of the probabilities is 1). We will show that for any such signing algorithm \mathcal{A} is an approximate time oblivious sampler of \mathcal{D} . Notice that since $k > H + c$ we have that $2^k \cdot p_i = r_i + t_i$ for some integer $r_i > 2$ and $0 \leq t_i < 1$:

$$\frac{p_i^+}{p_i^-} = \frac{\lceil p_i 2^k \rceil}{\lfloor p_i 2^k \rfloor} \leq \frac{r_i + 1}{r_i - 1} \leq 1 + \frac{2}{r_i - 1}.$$

If $\frac{2}{r_i - 1} < \varepsilon$, then we could conclude with the inequality $1 + x \leq e^x$ that $\frac{p_i^+}{p_i^-} \leq e^\varepsilon$, and therefore $\frac{p_i^+}{p_i^-} < e^\varepsilon$ as well as $\frac{p_i}{p_i^-} \leq e^\varepsilon$ since we know $p_i^+ \geq p_i \geq p_i^-$.

Therefore it is enough to find k for which $\frac{2}{r_i - 1} < \varepsilon$. Notice that by the definition of H we have for $k = H + c$ that $2^k \cdot p_i \geq 2^c$. Therefore $c = \log \frac{1}{\varepsilon} + 4$ suffices. \blacktriangleleft

Proof of Claim 2.21. We use the notation $p_i := \mathbb{P}[X = i]$ for $X \sim \mathcal{D}$. We also continue with the notation defined in Claim 2.20, p_i^+, p_i^- with $k = \log n + \log\left(\frac{1}{\delta}\right)$. Clearly $p_i^+ - p_i^- = \frac{1}{2^k} \leq \frac{\delta}{n}$. Let \mathcal{D}' be a *distribution* that correspond to some signing p_i^\pm and we will again take \mathcal{A} that outputs according to \mathcal{D} after $k = \log n + \log\frac{1}{\delta}$ coin tosses. For $S \subseteq [n]$:

$$\begin{aligned} \mathbb{P}[X \in S] - \delta \cdot \frac{|S|}{n} &= \sum_{i \in S} \left(p_i - \frac{\delta}{n} \right) \leq \sum_{i \in S} p_i^- \\ &\leq \Pr[\mathcal{O}(\mathcal{A}) \in S] \\ &\leq \sum_{i \in S} p_i^+ \leq \sum_{i \in S} \left(p_i + \frac{\delta}{n} \right) = \mathbb{P}[X \in S] + \delta \cdot \frac{|S|}{n}. \end{aligned}$$

Therefore since $|S| \leq n$

$$\Pr[\mathcal{O}(\mathcal{A}) \in S] - \delta \leq \mathbb{P}[X \in S] \leq \Pr[\mathcal{O}(\mathcal{A}) \in S] + \delta. \quad \blacktriangleleft$$

B.2 Missing Proofs in Section 3

Proof of Claim 3.4. Let D be some database and let k be the index of the first output level of $\mathcal{M}(D)$. By Lemma 3.2 we know that there are $r_1, \dots, r_\ell \in \mathcal{R}$ that appear in the output level of all possible databases. Since \mathbf{U} is sofic and there are infinitely many possible databases we can find a database D' such that $\mathbf{U}(D', r_1) = 0$ and therefore $G_{\mathbf{U}}(D) < 1$ and thus the guaranteed utility of \mathcal{M} satisfies $\mathbf{U}(\mathcal{M}) < 1$. \blacktriangleleft

Proof of Proposition 3.5. Assume that $G_{\mathcal{T}}(\mathcal{M}) < m$. Let D be any database and $\tau > 0$ and let r_1, \dots, r_ℓ be the elements of \mathcal{R} that D outputs in levels $1, \dots, \tau m$. We know that there is a database D' that satisfies $\mathbf{U}(D', r_i) = 0$ but by Markov's inequality $\mathbb{P}[\mathcal{T}(\mathcal{M}(D')) > \tau m] < \frac{1}{\tau}$ and therefore we get that $G_{\mathbf{U}}(\mathcal{M}) \leq \mathbf{U}(D') \leq \frac{1}{\tau}$. Since $\tau > 0$ is arbitrary we conclude that $G_{\mathbf{U}}(\mathcal{M}) = 0$. \blacktriangleleft

