ABSTRACT

Title of Dissertation: ALGORITHMIC FAIRNESS AND ONLINE DECISION MAKING IN COMBINATORIAL OPTIMIZATION AND UNSUPERVISED LEARNING

Sharmila Duppala
Doctor of Philosophy, 2026

Dissertation Directed by: John P. Dickerson and Aravind Srinivasan
Department of Computer Science

Classical combinatorial and unsupervised learning problems such as graph matching, set packing, and clustering are commonly used to model large-scale real-world applications. Examples of such applications include facility location problems to provide services in emergency, online advertising markets (matching ads to customers), organ exchanges which involves patient-donor pairs to exchange organs, recommending applicants to jobs, assigning jobs to servers, clustering of points, among others.

These problems are extensively studied and are reasonably well-understood. Recently, there has been a growing emphasis on integrating fairness considerations into these classical problems spurred due to the bias in existing algorithms. This has given rise to a growing field known as *algorithmic fairness*. In this thesis we focus on designing algorithms with provable guarantees for fair and/or online variants of the aforementioned problems.

In the first part of the thesis we study fair variants of classical combinatorial optimization problems like graph matching, online bipartite matching and set packing. Particularly, we study these prob-

lems under well-studied notions of group fairness called proportionality fairness and rawlsian fairness. The primary challenge in designing fair algorithms for these problems is that checking the feasibility[1] is often NP-hard. In this work, we introduce probabilistic relaxations for deterministic group-fairness constraints. Further, we provide randomized algorithms that offer approximate probabilistic fairness guarantees, while maintaining a good (constant factor) approximation on the objective.

We begin with the fair online bipartite matching problem. While existing works primarily focus on one-sided fairness, we generalize the framework to provide simultaneous guarantees for both sides of the market. We incorporate both group and individual Rawlsian fairness criteria which seeks to maximize the utility of the worst case groups and individuals. To address this, we present randomized algorithms that satisfy these constraints in expectation. Furthermore, our algorithms also allow for tunable parameters designed to handle the trade-offs between the utilities of the two sides of market and the platform operator (e.g., in ride-sharing, the two sides are the drivers and riders, where the platform (Uber/Lyft) refers to the operator).

Next we present set packing and bipartite matching problems under proportional fairness constraints. Proportional fairness is a well-studied notion of group fairness where the desired lower bound and upper bound proportions set by stakeholder for each group must be satisfied. A notable special case is when when these bounds are equal for all groups. In this thesis we begin by introducing strong probabilistic relaxations of group-fairness notions and we refer to the solutions that satisfy such definitions as *probably almost fair* and *probably approximately fair*. The high-level idea is that we show strong guarantees like tail bounds for the probability that the solution returned by our randomized algorithm is almost-fair and approximately-fair respectively. We conclude the first part by introducing a variant of online minimum bipartite matching problem under uniform metric and provide algorithms

[1] determining the existence of a solution that satisfies the fairness constraints

along with matching lower bounds.

The second part of this thesis study the problems related to *fair clustering*, a well-studied topic in the fairness of unsupervised learning. Particularly, we consider the practical downstream challenges of fair clustering. In the first problem, we study the setting where cluster centers are assigned *qualitative* labels based on their inherent features. Because these labels are often shared across multiple clusters, traditional fair clustering constraints no longer capture this settings. Therefore, we propose *fair labeled clustering* that ensures proportional demographic representation across labels rather than individual clusters. We show that unlike their NP-hard counterparts in traditional fair clustering, these problems admit computationally efficient solutions. We further study the generalized model where the decision-maker has the flexibility to assign labels to the centers.

Next, we study the problem of noisy group memberships by proposing a robust fair clustering framework. Specifically, we formulate a noise model to capture the imperfect group information, requiring only a small number of noise parameters. We formally define the robust fair clustering problem for the k-center objective under proportional fairness constraints. We provide algorithm with provable guarantees for both robustness and clustering quality. We further characterize the inherent trade-off between these two metrics. Finally, empirical evaluations on real-world datasets validate the effectiveness and practical utility of our proposed methods.

$\breve{}$Finally, we conclude by discussing open problems and outlining future research directions. $\breve{}$

ALGORITHMIC FAIRNESS AND ONLINE DECISION MAKING IN COMBINATORIAL OPTIMIZATION AND UNSUPERVISED LEARNING

by

Sharmila Duppala

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2026

Advisory Committee:
Professor Aravind Srinivasan, Chair/Advisor
Dr. John P. Dickerson, Co-Advisor
Professor Joseph F. Jaja, Dean's Representative
Assistant Professor Laxman Dhulipala
Professor David Mount
Professor Ashok Agrawala

Dedication

To my family.

Acknowledgments

There is a saying that it takes a village to raise a child. I believe it also took a village for me to earn this PhD - many of the names may not appear in this thesis, though I will do my best to acknowledge as many as possible.

This thesis would not have been possible without the immeasurable support of my advisors, Aravind Srinivasan and John P. Dickerson. I am deeply grateful to Aravind for the opportunity to learn from his profound knowledge of randomized algorithms, concentration inequalities, among others. Whenever I found myself stuck, Aravind offered insights that I carefully noted down, often fully appreciated their depth only after hours of reflection and discussions with other students and collaborators. Beyond technical inputs, he taught me how to be a dedicated student, a generous collaborator, and a kind human being by setting a strong example. Aravind fostered a highly collaborative, inclusive, healthy and supportive environment within the group, which helped me grow not only as a researcher but also as a person. His feedback was consistently filled with empathy and understanding. Through Aravind, I learned that collaboration is a powerful way to learn and to help others, and that I can contribute meaningfully regardless of whether I am the smartest person in the room. I was never judged for this, which a perspective that was entirely new and profoundly impactful for me. Finally, I owe my love for probability, randomized algorithms, and any understanding I have developed in these areas entirely to Aravind, and I look forward to a lifetime of continued learning in this field.

I began my grad school at UMD just before the COVID-19 pandemic, and I am especially grateful to John, whose unwavering support was a primary reason I was able to persevere through my PhD, without which I would have quit my PhD long ago. His passion for research and enthusiasm were infectious and they consistently encouraged me to pursue independent ideas with confidence and conviction. I have been deeply inspired by John's broad knowledge across diverse applications, which

offered me new perspectives and helped me formulate meaningful research problems. Above all, I admire the enthusiasm he brings to every discussion - a quality I strive to cultivate in myself. I am forever grateful for the humble example he sets.

I am also extremely grateful to my committee members: Ashok Agrawala, Laxman Dhulipala, Joseph Jaja, and David Mount for their time, insightful feedback, and prompt responses while serving on my committee. Their support made the process smooth and pleasant for me. I also thank Prof. Hal Daume for serving on my Thesis Proposal committee.

Special thanks to professors Alexander Barg, Dave Mount, Aravind Srinivasan, Xiaodi Wu for teaching my favorite courses at UMD. I extend my humble thanks to all my collaborators without which this thesis would not have been possible - Brian Brubach, Davidson Cheng, Seyed Esmaeili, Nathaniel Grammel, Juan Luque, Calum MacRury, Vedant Nanda, Karthik Abinav Sankararaman, and Pan Xu.

I sincerely thank the funding from NSF Award CCF-1918749 and NSF CAREER Award IIS-1846237 for supporting the research in this thesis. I sincerely thank the CS department staff - Tom, Sharron, Migo and others for their extensive support throughout my time at UMD.

I am deeply grateful to my Master's advisor, Rezaul Chowdhury, who first introduced me to the world of research, from reading papers to writing one. His kindness, constant support, and genuine care for my well-being and career shaped my path in meaningful ways. His thoughtfulness extended beyond academia; he came to see me at the hospital immediately after my surgery with a box of chocolates, a gesture that I will never forget. He showed me even greater support each time I failed, teaching me that unconditional help means standing by someone more strongly in moments of failure. I owe most of where I am today to his unconditional kindness, support and grace and am infinitely indebted to him for everything he has ever done for me.

I am extremely grateful (not in any particular order) to my close friends and collaborators: Nate (for our innumerable discussions and problem-solving sessions on math, life, the PhD, countries, culture, boundaries, and beyond), Kishen (for the shared love of Hyderabadi biryani, algorithms, and math; meeting you in my third year and having you as my desk neighbor was truly a blessing that enriched my PhD experience manyfold. I learned immensely from you - your problem-solving speed is quite impressive, and I loved working through problems together (even if we never wrote a paper!).), Juan (from whom I learned immensely, both technically and beyond, including language, culture, and more. Your feedback about everything was deeply helpful in making me a better person/collaborator), Renata and Nitya (I wish I had more time to do research and other things with you; both of you are so brilliant yet so humble, and cheerful – qualities I would like to emulate), Calum (working with you has been one of the highlights of my PhD - I learned immensely from you and am deeply grateful for the opportunity to collaborate with such a kind, helpful, and knowledgeable researcher), Seyed (from whom I learned a great deal about writing papers and formulating well-motivated research problems), and Brian and Pan (for mentoring me and helping me begin my journey in matching problems). Special thanks to fellow theory students Sijing, Arushi, Jeff, Harper, Leo, Aviva, Auguste for their enthusiasm and frequent participation in CATS.

Special thanks to Tom, Migo, Sharron and other administrative staff for making my life easier. I am quite spoiled by such helpful staff who have instant solutions to all my logistics mishaps. Will miss you all.

I truly enjoyed my time in IRB-2104 and feel grateful to have made such an amazing set of friends and lab mates: Keon (thanks for everything! You're one of the coolest people I know), Pankhy (for kind things you've done for me over the years, despite our frequent arguments, most of which I gladly take the blame for.), Jie (our walks and countless discussions about life as a graduate student;

you have been a friend from day one until the end), Kishen (you really are an amazing desk neighbor!), and Nate, the unsolicited visitor but crowd favorite (thanks for all the entertainment during our discussions and for patiently tolerating my unreasonable enthusiasm for work). Special mention to Kwesi for constantly invading IRB 2104, overhearing our conversations from the adjacent office, and always keeping an eye out for us.

I also made many wonderful friends in IRB-5108 during my final year; thanks to Shreyas, Manan, Laura, Finn, Alperen, and Justine for your company over holidays, weekends, and late night lab-owl moments. Sharing a lab with such enthusiastic and lively people made everything easier. Above all, everyone I have mentioned are incredibly smart and talented, yet quite simple and humble – a quality I deeply admire and strive to emulate. Graduate school has been the hardest yet the best time of my life and has offered me new perspectives on work, research, culture, friendships, and education.

Lastly, I am grateful to my UMD friends who took the time to stay in touch and support me throughout my PhD: Susmija (I wish I had met you earlier, I instantly felt comfortable being unfiltered and vulnerable with you! grateful for your care and for inspiring me to be a better friend), Gihan (you constantly make me think deeply about important things I previously only had superficial opinions about), Yuelin (your genuine care for others and willingness to help anyone, anytime, is truly inspiring), Srilekha (for being part of this long journey and holding me up during all the lows), Sandesh (a tremendous source of inspiration and a role model for me as a PhD student), and Arghya (for sharing cool math, Bengali poems, and insights about Rabindranath Tagore; your enthusiasm is infectious!), last but not the least: Sreedhar, Raghu, Harsha for being good friends and caring for me during my low times.

I also want to thank the friends I made at conferences and workshops, who are full of fun, enthusiasm and some of whom became close friends and supported me during various phases of my

PhD: Rucha, Vaishali, Eklavya, Arpit, Pramthamesh, Aniket, Ananya, Nidhi, and others.

Many thanks to my close group of friends: Kishen, Vinu, Souradip, Peeyush, Pankhy, and Gihan, for all the fun memories and the immense support over the years. Many thanks to Anu and Arushi for treating me like family. Anu has gone beyond and above to help me give a good job talk and wise career advice, Arushi for all the love and care during several sick days. I am also grateful to the other friends in College Park with whom I shared many entertaining and pleasant conversations: Ethan, Noor, Divya, Abhilasha, Sahil, Vasu, Pulkit, Pooja, Vaishnavi, and others whom I may not be remembering at the moment. My school, college, and Master's friends have also played a huge role in shaping who I am today: special thanks to my oldest (ranging 8-20 years) friends Varun, Teja, Nandan, Sreeja, Harshita, Sravanthi, Alekhya, Bharu, Sowji, Sammu, Nikki, Sandeep (we miss you, ra), Ch, Padmasree, Mbhu, Kanishka, Akshay, Praneeth, Kishan, Zafar, Yimin, Gandhi, Haindavi, Amol, Sanju, Mahdi, Sergey, Guru, Anu, Naresh, Srujana, and many others for their friendship and love.

A huge thanks to my extended family for helping my parents and supporting them whenever my brother and I were away: especially, Pinni, Chinnanna, Sathwik, Rohit, Dhilli Mavayya, Sudha Mavayya, Ramu Mavayya, all the Atthas, and other relatives. Special thanks to my cousins, especially Sathwik and Rohit, whose unconditional love and unwavering support kept me grounded. I am also deeply grateful to my in-laws for embracing me as their own daughter, supporting me in every possible way, and always wishing the best for me. Finally, I thank my grandparents who gave me everything that is possible for them and their affection is the strength I carry with myself forever.

Lastly, my husband, Raju, has been an anchor in my life since end of 2020. None of this would have possible without his constant support, encouragement and selflessness. I am deeply grateful to him for always believing in me, understanding me, and encouraging me beyond what I could have imagined. His insights about life have had a huge influence in how I conduct myself, as a friend,

researcher, student, daughter, collaborator, and a human being. I can never thank him enough for all that he has done and continues to do for me.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Algorithmic fairness has gained significant interest in the recent years due to the growing applications of algorithmic decision-making in socially critical applications like recidivism prediction, loan application approvals, renal-failure risk assessment, etc [1]. There have been numerous studies showing bias of these algorithms from criminal justice to healthcare – e.g., a widely popularized study [2] conducted by ProPublica found that the COMPAS, an algorithm used for risk prediction was biased against African-American defendants. In healthcare, a study conducted by [3] found that the highest-risk black patients (the threshold at which the patients are automatically enrolled in the treatment program), have significantly more illnesses compared to white patients with the same risk score. Due to the direct impact of such algorithms on individuals, fairness has become a critical necessity in these algorithms. The aforementioned studies provide the motivation for the research presented in the following sections.

The central theme of this thesis is: (i) to incorporate rigorous notions of fairness into classical problems in combinatorial optimization and unsupervised learning, and (ii) to develop algorithms with provable guarantees for desired fairness metrics and the underlying optimization objectives. While fairness has been one of the foundational topic in economics and game theory for a long time, it has only recently emerged as a critical requirement for problems combinatorial optimization and machine learning with widespread deployment of such algorithms in real-world. For example, in two-sided

marketplaces, algorithms are deployed to automate decisions such as matching residents to hospitals or kidney donors with recipients.

In this research we study the fair generalizations of classical problems like graph matching, online matching, set packing, and clustering of points under varying notions of group and individual fairness. Although the source of *unfair* outcomes is often due to historical biases in the training data, it can also arise from the algorithm logic/design, objective functions, or constraints of the algorithm itself. This thesis specifically addresses the latter - algorithmic fairness which focuses on mitigating the systematic bias induced during the optimization process.

1.1 Motivation for Fairness in Combinatorial Optimization

Combinatorial optimization arise in various applications like job recommendations (recommending jobs to applicants), ad-matching (matching ads to online customers/viewers), organ-exchanges (matching incompatible patient-donor pairs to exchange kidneys), cloud scheduling (mapping jobs to resources), ride-sharing (matching riders to drivers), clustering of points (used in several downstream applications in Machine Learning), among others. Several classical variants of these problems are optimized solely based on utilitarian objectives. For example, maximizing the revenue in ad-matching or maximizing the number of matched pairs in organ-exchanges. However, these marketplaces typically have users with diverse preferences based on demographics such as race, gender, and age. Therefore, it is important to incorporate fairness considerations into these problems to prevent systemic discrimination by these algorithms.

1.1.1 Group and Individual Fairness

The two main paradigms that emerged in fairness literature are group fairness and individual fairness. Group fairness aims to provide fairness guarantees at a group-level such as ensuring fair treatment to different groups of people. This can help to prevent systematic discrimination against any group. A significant body of research in group fairness explored various notions of fairness. Among these, we mainly consider Rawlsian (maxmin) fairness, aiming to maximize the utility for the worst-off group [4, 5, 6], and proportional fairness, as explored in [7, 8, 9], ensures that the fractional representation of individuals from each group lies between the desired lower and upper bounds. Most of these studies involve assigning group memberships to individuals/entities in the underlying problem. Next, we rigorously define both of these fairness notions. Consider an optimization problem defined by set of individuals $\mathcal{P}$ partitioned into ℓ disjoint groups forming a partition $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_\ell$. Let $\mathcal{F}$ be the discrete of feasible solutions to the optimization problem.

1.1.1.1 Rawlsian or Maxmin Fairness

Rawlsian fairness, also referred to as the maxmin group fairness has origins from Criminal Justice theory by John Rawls. Maxmin fairness ensures that no individual/group is marginalized to achieve a higher average utility as studied in previous works [e.g., 4, 5, 6, 10, 11]. Here we maximize the minimum utility received by any group/individual to improves the utility for the worst-off groups.

Definition 1.1.1 (Rawlsian or Maxmin Group Fairness)**.** Let a utility function $u_e : \mathcal{P} \to \mathbb{R}$ representing

the utility of individual e. A solution $S^* \in \mathcal{F}$ is said to be group-level maxmin fair if:

$$S^* \in \arg\max_{S \in \mathcal{F}} \left(\min_{g \in \{1,2,\ldots,\ell\}} \sum_{e \in \mathcal{P}_g} u_e(S) \right) \tag{1.1}$$

Similarly, we say that $S^* \in \mathcal{F}$ is individual maxmin fair if:

$$S^* \in \arg\max_{S \in \mathcal{F}} \left(\min_{e \in \mathcal{P}} u_e(S) \right) \tag{1.2}$$

The maxmin fairness is generally applied to maximization problems; however, for minimization problems, we adapt this definition to minmax fairness objective. In this paradigm, the goal is to minimize the maximum cost incurred by any group, thus protecting the worst-off group.

1.1.1.2 Proportional Fairness

Proportional fairness is a group-level constraint that bounds the representation of different demographic groups within a solution. Specifically, any feasible solution must satisfy the proportional group representation predetermined by stakeholders, as established in various fair clustering and matching works [e.g., 7, 8, 12, 13, 14, 15, 16]. By enforcing lower bound (α) for minority protection and an upper bound (β) for restricted dominance, we ensure that the fraction of individuals from any group in the solution is at least α and at most β respectively. There are various notions of proportional fairness, depending on the chosen *lower* and *upper* bound parameters. We define a small subset of these notions below.

Definition 1.1.2 ((α, β)-Balanced Fairness)**.** A solution $S \in \mathcal{F}$ is (α, β)-balanced if for every group

$g \in \{1, \ldots, \ell\}$, the number of individuals/elements selected from that group g satisfies:

$$\alpha \leq \frac{|S \cap \mathcal{P}_g|}{|S|} \leq \beta$$

where $\mathcal{P}_g$ refers to the elements from group g and $0 \leq \alpha \leq \beta \leq 1$ are fixed constants.

This definition can be further generalized to allow for different values of lower and upper bound requirements (α and β) for different groups.

Definition 1.1.3 (Generalized (α, β)-Balanced Fairness)**.** A solution $S \in \mathcal{F}$ is generalized (α, β)-balanced if for every group $g \in \{1, \ldots, \ell\}$, the number of individuals/elements selected from that group g satisfies:

$$\alpha_g \leq \frac{|S \cap \mathcal{P}_g|}{|S|} \leq \beta_g$$

where $\mathcal{P}_g$ refers to the elements from group g and $\vec{\alpha}, \vec{\beta} \in [0, 1]^\ell$ are ℓ-dimensional vectors where $0 \leq \alpha_g \leq \beta_g \leq 1$ are fixed constants for any group g.

Thus, allowing us to generalize this framework. And a notable special case of this notion of fairness is when $\alpha_g = \beta_g$ for any group g in which case, we ask for solution that exactly satisfies the proportional requirement.

Definition 1.1.4 (α-Balanced Fairness)**.** A solution $S \in \mathcal{F}$ is generalized α-balanced if for every group $g \in \{1, \ldots, \ell\}$, the number of individuals/elements selected from that group g satisfies:

$$\frac{|S \cap \mathcal{P}_g|}{|S|} = \alpha_g$$

where $\mathcal{P}_g$ refers to the elements from group g and $0 \leq \alpha_g \leq 1$ are fixed constants for any group g.

1.2 Overview of the thesis

This thesis is divided into two main parts. In the first part, we present the bipartite matching and set packing under various notions of fairness and stochastic settings. We address these problems by formulating fair and online variants of aforementioned problems and providing randomized algorithms with provable guarantees on both fairness (e.g., Rawlsian and proportional fairness) and underlying objective – e.g., weight of the overall matching (in bipartite matching) or packing (in set packing).

The second part of the thesis shifts the focus to downstream challenges in fair clustering. We first investigate fair labeled clustering, a new variant addressing scenarios where cluster centers exhibit qualitative differences. For example, in healthcare facility allocation, centers may vary based on available services. In such cases the assignment of centers must ensure group representation not only within clusters but also across each facility/center type. We introduce two variants of fair labeled cluster: (i) where center types are fixed a priori, and (ii) where the algorithm is allowed to assign labels to each center. We provide algorithms and hardness results for the two variants. Next, we study fair k-center problem under group membership uncertainty. Specifically, we introduce a novel noise model to capture inaccuracies in group labels and provide algorithms with provable guarantees for both clustering cost and robustness with respect to fairness.

1.2.1 Fairness in Online Bipartite Matching

Online bipartite matching has been used to model many important applications such as crowdsourcing [17, 18, 19], ridesharing [20, 21, 22], and online ad allocation [23, 24]. Online bipartite matching algorithms are often designed to optimize a performance measure—usually, maximizing overall profit for the platform operator or a proxy of that objective. However, fairness considerations

were largely ignored. This is troubling especially given that recent reports have indicated that different demographic groups may not receive similar treatment. This has given rise to a line of research focused on designing fair algorithms for online bipartite matching. [5] studied online bipartite matching with fair considerations for only one side of the matching. However, several applications of online bipartite matching are two-sided market places (e.g., ridesharing) with three distinct entities: the two sides of the matching (e.g., riders and drivers) and the platform operator (e.g., Uber/Lyft). In Chapter 2, we generalize the one-sided fairness model by [5] in two key ways (i) we extend the model to ensure fairness for both sides of the market and, (ii) we additionally formulate individual fairness constraints for participants within each side of the market. Especially, we consider both group and individual-level rawlsian fairness criteria describe in Definition 1.1.1 for both sides of the market. Further, we provide fair algorithms with provable guarantees for the platform operator's profit as well as fairness guarantees for the two sides of the market.

This work is largely based on our published paper [25].

1.2.2 Proportional Fairness in Graph Matching

Graph matching, in particular the special case of bipartite matching, is a classical computational problem with many applications such as ad allocation [26, 27], crowdsourcing [18, 19, 28, 29], job hiring [30], organ exchange [31, 32, 33, 34], and ride sharing [29, 35, 36]. In many applications, algorithms are employed to make decisions that could significantly impact the lives of individuals. In such settings, we ought to ensure fairness and equity in the decision-making process. However, classical algorithms typically do not consider such socially motivated objectives and constraints such as fairness or diversity.

There are many ways to formalize and incorporate some notion of *fairness* into the computed solutions of matching problems (see, e.g., [16, 25]). Following the work of Bandyapadhyay et al. [16], we consider a notion of *proportional fairness* described in Definition 1.1.2. Unfortunately, although this notion of fairness seems quite powerful, it may in fact be too strong in the basic formulation: Bandyapadhyay et al. [16] demonstrated that (α, β)-BALANCEDMATCHING is NP-hard to approximate, even when G is an *unweighted path graph*. To address this fundamental hardness, we consider a slightly relaxed probabilistic notion of proportional fairness. Roughly speaking, we allow for small violations of the α and β bounds while still guaranteeing a constant factor approximation on the overall objective, i.e., the weight of the matching. Particularly, we ask the following question: *does there exists an approximation algorithm with constant factor violations in fairness while guaranteeing a good approximation on the objective?* In Chapter 3 answer this question affirmatively by providing approximate randomized algorithm that satisfy the aforementioned probabilistic guarantees for fairness.

This work is largely based on our published paper [37].

1.2.3 Fair Set Packing

Fielded kidney exchange programs (KEPs) often employ a utilitarian objective, where the overall utility of successful transplants is maximized. This approach, however, can penalize certain classes of hard-to-match patients (e.g., highly-sensitized, older, O-type blood, and others), as highlighted by many studies from the AI/ML [38, 39, 40, 41], economics [42, 43], and medical communities [44].

Prioritizing the welfare of highly-sensitized (hard-to-match) patients has been studied in several previous works [40, 42, 45] as a natural *fairness* criterion. We formulate the KEP problem as k-set packing (inspired from the works of [46, 47]) under the notion of α-Balanced fairness described in

Definition 1.1.4. However, this notion is too strong and therefore we relax this to probabilistic group fairness notion of proportionality fairness. In Chapter 4 we begin by rigorous formulating the problem under the relaxed notions of probabilistic fairness and provide randomized algorithms that return a probabilistically fair solution with a good approximation guarantees on the overall objective. Further, we also discuss the hardness results for several variants of *fair k-set packing*.

This work is largely based on our published paper [48]

1.2.4 Online Minimum Bipartite Matching

Applications such as assignment of ride-sharing (like Uber, Lyft), assignment of client requests to servers, traditionally called the k-server problem, real-time crowdsourcing where workers are to be assigned to jobs etc, can be modelled as Online Minimum Bipartite Matching problem. Formally, the OMBM problem can be defined as finding a matching that maximizes the size of the matching with an objective of minimizing the overall cost (total distance for metric spaces) of the matching where the requests arrive dynamically and are to be matched immediately to an unmatched (unassigned) service.

The online metric matching problem is similar to the wellstudied k-server problem introduced by Manasse et.al in [49] as it can be viewed as a restricted case of the k-server problem where the servers do not move. The methods used to give lower and upper bounds for the online matching problem, particularly on the uniform metric, are closely related to the results for k-server. We recall that uniform metric consists of set of points where distance between any two points is the same, without loss of generality, we can assume that this distance is one unit. In Chapter 5 we give optimal deterministic and randomized algorithms for the OMBM (Online Bipartite Matching Problem) under uniform metric and ROA (random order arrival) model.

This work is largely based on our published paper [50].

1.2.5 Fair Labeled Clustering

The widespread use of machine learning algorithms in settings that directly affect human lives has instigated significant interest in designing variants of these algorithms that are provably fair. Recent work in this direction has produced numerous algorithms for the fundamental problem of clustering under many different notions of fairness. Perhaps the most common family of notions currently studied is group fairness, in which proportional group representation is ensured in every cluster.

We extend this direction by considering the downstream application of clustering and how group fairness should be ensured for such a setting. Specifically, we consider a common setting in which a decision-maker runs a clustering algorithm, inspects the center of each cluster, and decides an appropriate outcome (label) for its corresponding cluster. In hiring for example, there could be two outcomes, positive (hire) or negative (reject), and each cluster would be assigned one of these two outcomes. To ensure group fairness in such a setting, we would desire proportional group representation in every label but not necessarily in every cluster as is done in group fair clustering.

In Chapter 6 we formulate the problem and provide algorithms for such problems and show that in contrast to standard fair clustering problem, these new variant permits efficient solutions. We also consider a well-motivated alternative setting where the decision-maker is free to assign labels.

This work is largely based on our published paper [51]

1.2.6 Robust Fair Clustering under Noisy Group Memberships

In fair clustering under *proportional* fairness, each point belongs to a demographic group and therefore each demographic group has some percentage representation in the entire dataset. One can see a significant advantage behind ensuring group fairness. Despite the attractive properties of group fairness, it requires complete knowledge of each point's membership in a group. In practice—say, in an advertising setting where membership is estimated via a machine learning model, or in a lending scenario where membership may be illegal to estimate at train time—knowledge of group membership may range from noisy, to adversarially corrupted, to completely unknown. This salient problem has received significant attention in fair *classification* [see, e.g., 52, 53, 54, 55, 56, 57]. However, this important consideration has not received significant attention in fair *clustering* with the exception of the theoretical work of Esmaeili et al. [58] that introduced uncertain group membership and the empirical work of Chhabra et al. [59], who provide a data-driven approach to achieve robustness against adversarial perturbations on fair clustering systems.

Chapter 7 addresses the practical modeling shortcomings of both Esmaeili et al. [58] and Chhabra et al. [59] and gives new theoretical worst-case guarantees.

This chapter is largely based on our published paper [60]

Finally we conclude with Chapter 8 where we outline important directions for future work.

Chapter 2: Rawlsian Fairness in Online Bipartite Matching: Two-Sided, Group, and Individual

This chapter is largely based on our paper [25].

2.1 Introduction

Online bipartite matching has been used to model many important applications such as crowdsourcing [17, 18, 19], rideshare [20, 21, 22], and online ad allocation [23, 24]. In the most general version of the problem, there are three interacting entities: two sides of the market to be matched and a platform operator which assigns the matches. For example, in rideshare, riders on one side of the market submit requests, drivers on the other side of the market can take requests, and a platform operator such as Uber or Lyft matches the riders' requests to one or more available drivers. In the case of crowdsourcing, organizations offer tasks, workers look for tasks to complete, and a platform operator such as Amazon Mechanical Turk (MTurk) or Upwork matches tasks to workers.

Online bipartite matching algorithms are often designed to optimize a performance measure—usually, maximizing overall profit for the platform operator or a proxy of that objective. However, fairness considerations were largely ignored. This is troubling especially given that recent reports have indicated that different demographic groups may not receive similar treatment. For example, in rideshare platforms once the platform assigns a driver to a rider's request, both the rider and the driver

have the option of rejecting the assignment and it has been observed that membership in a demographic group may cause adverse treatment in the form of higher rejection. Indeed, [61, 62, 63] report that drivers could reject riders based on attributes such as gender, race, or disability. Conversely, [64] reports that drivers are likely to receive less favorable ratings if they belong to certain demographic groups. A similar phenomenon exists in crowdsourcing [65]. Moreover, even in the absence of such evidence of discrimination, as algorithms become more prevalent in making decisions that directly affect the welfare of individuals [66, 67], it becomes important to guarantee a standard of fairness. Also, while much of our discussion focuses on the for-profit setting for concreteness, similar fairness issues hold in not-for-profit scenarios such as the fair matching of individuals with health-care facilities, e.g., in the time of a pandemic.

In response, a recent line of research has been concerned with the issue of designing fair algorithms for online bipartite matching. [68, 69, 70] present algorithms which give a minimum utility guarantee for the drivers at a bounded drop to the operator's profit. Conversely, [5] give guarantees for both the platform operator and the riders instead. Finally, [71] shows empirical methods that achieve fairness for both the riders and drivers simultaneously but lacks theoretical guarantees and ignores the operator's profit.

Nevertheless, the existing work has a major drawback in terms of optimality guarantees. Specifically, the two sides being matched along with the platform operator constitute the three main interacting entities in online matching and despite the significant progress in fair online matching none of the previous work considers all three sides simultaneously. In this paper, we derive algorithms with theoretical guarantees for the platform operator's profit as well as fairness guarantees for the two sides of the market. Unlike the previous work we not only consider the size of the matching but also its quality. Further, we consider two online arrival settings: the KNOWNIID and the richer KNOWNAD setting

(see Section 2.2 for definitions). We consider both group and individual notions of Rawlsian fairness and interestingly show a reduction from individual fairness to group fairness in the KNOWNAD setting. Moreover, we show upper bounds on the optimality guarantees of any algorithm and derive impossibility results that show a conflict between group and individual notions of fairness. Finally, we empirically test our algorithms on a real-world dataset.

2.2 Online Model & Optimization Objectives

Our model follows that of [24, 72, 73, 74] and others. We have a bipartite graph $G = (U, V, E)$ where U represents the set of static (offline) vertices (workers) and V represents the set of online vertex types (job types) which arrive one-by-one in an online manner.

The online matching is done over T rounds. In a given round t, a vertex of type v is sampled from V with probability $p_{v,t}$ with $\sum_{v \in V} p_{v,t} = 1, \forall t \in [T]$ the probability $p_{v,t}$ is known beforehand for each type v and each round t. This arrival setting is referred to as the known adversarial distribution KNOWNAD setting [21, 74].

When the distribution is stationary, i.e. $p_{v,t} = p_v, \forall t \in [T]$, we have the arrival setting of the known independent identical distribution (KNOWNIID). Accordingly, the expected number of arrivals of type v in T rounds is $n_v = \sum_{t \in [T]} p_{v,t}$, which reduces to $n_v = Tp_v$ in the KNOWNIID setting. We assume that $n_v \in \mathbb{Z}^+$ for KNOWNIID [73].

Every vertex u (v) has a group membership[1], where $\mathcal{G}$ being the set of all group memberships; for any vertex $u \in U$, we denote its group memberships by $g(u) \in \mathcal{G}$ (similarly, we have $g(v)$ for $v \in V$). Conversely, for a group g, $U(g)$ ($V(g)$) denotes the subset of U (V) with group membership

[1]For a clearer representation we assume each vertex belongs to one group although our algorithms apply to the case where a vertex can belong to multiple groups.

g. A vertex u (v) has a set of incident edges E_u (E_v) which connect it to vertices in the opposite side of the graph. In a given round, once a vertex (job) v arrives, an irrevocable decision has to be made on whether to reject v or assign it to a neighbouring vertex u (where $(u, v) \in E_v$) which has not been matched before. Suppose, that v is assigned to u, then the assignment is not necessarily successful rather it succeeds with probability $p_e = p_{(u,v)} \in [0, 1]$. This models the fact that an assignment could fail for some reason such as the worker refusing the assigned job. Furthermore, each vertex u has patience parameter $\Delta_u \in \mathbb{Z}^+$ which indicates the number of failed assignments it can tolerate before leaving the system, i.e. if u receives Δ_u failed assignments then it is deleted from the graph. Similarly, a vertex v has patience $\Delta_v \in \mathbb{Z}^+$, if a vertex v arrives in a given round, then it would tolerate at most Δ_v many failed assignments in that round before leaving the system.

For a given edge $e = (u, v) \in E$, we let each entity assign its own utility to that edge. In particular, the platform operator assigns a utility of w_e^O, whereas the offline vertex u assigns a utility of w_e^U, and the online vertex v assigns a utility of w_e^V. This captures entities' heterogeneous wants. For example, in ridesharing, drivers may desire long trips from nearby riders, whereas the platform operator would not be concerned with the driver's proximity to the rider, although this maybe the only consideration the rider has. Similar motivations exist in crowdsourcing as well. We finally note that most of the details of our model such the KNOWNIID and KNOWNAD arrival settings as well as the vertex patience follow well-established and pratically motivated model choices in online matching.

Letting $\mathcal{M}$ denote the set of successful matchings made in the T rounds, then we consider the following optimization objectives:

Operator's Utility (Profit): The operator's expected profit is simply the expected sums of the profits across the matched edges, this leads to $\mathbb{E}[\sum_{e \in \mathcal{M}} w_e^O]$.

Rawlsian Group Fairness:

1. **Offline Side:** Denote by $\mathcal{M}_u$ the subset of edges in the matching that are incident on u. Then our fairness criterion is equal to

$$\min_{g \in \mathcal{G}} \frac{\mathbb{E}[\sum_{u \in U(g)} (\sum_{e \in \mathcal{M}_u} w_e^U)]}{|U(g)|}. \tag{2.1}$$

   this value equals the minimum average expected utility received by a group in the offline side U.

2. **Online Side:** Similarly, we denote by $\mathcal{M}_v$ the subset of edges in the matching that are incident on vertex v, and define the fairness criterion to be

$$\min_{g \in \mathcal{G}} \frac{\mathbb{E}[\sum_{v \in V(g)} (\sum_{e \in \mathcal{M}_v} w_e^V)]}{\sum_{v \in V(g)} n_v}.$$

   this value equals the minimum average expected utility received throughout the matching by any group in the online side V.

Rawlsian Individual Fairness:

1. **Offline Side:** The definition here follows from the group fairness definition for the offline side by simply considering that each vertex u belongs to its own distinct group. Therefore, the objective is $\min_{u \in U} \mathbb{E}[\sum_{e \in \mathcal{M}_u} w_e^U]$.

2. **Online Side:** Unlike the offline side, the definition does not follow as straightforwardly. Here we cannot obtain a valid definition by simply assigning each vertex type its own group. Rather, we note that a given individual is actually a given arriving vertex at a given round $t \in [T]$,

accordingly our fairness criterion is the minimum expected utility an individual receives in a given round, i.e. $\min_{t\in[T]} \mathbb{E}[\sum_{e\in\mathcal{M}_{v_t}} w_e^V)]$, where v_t is the vertex that arrived in round t.

2.2.1 Arrival Model:

The modeling choices we have made follow standard settings in online matching [24, 74]. To elaborate further, the initial seminal paper on online matching [75] does not assume any prior knowledge on the arrival of the online vertices of V and follows adversarial analysis to establish theoretical guarantees on the competitive ratio. In addition to overly pessimistic theoretical results, the lack of prior knowledge is often an unrealistic assumption. Most decision makers in online matching settings are able to gain knowledge on the arrival rates of the online vertices and this knowledge can be used to build more realistic probabilistic knowledge of the arrival.

Specifically, the Known Independent and Identically Distributed KNOWNIID model is an established model in online matching [19, 24, 72, 76, 77]. In this model, the collection of arriving vertices on the online side belong to a finite set of known types where the type of a vertex v decides the edge connections E_v it has to the vertices of U along with the weights $w_e, \forall e \in E_v$ of those edges. Further, a given vertex of type v arrives with the same probability p_v in every round. These arrival probabilities can be estimated easily from historical data based on previous matchings.

While the KNOWNIID model utilizes prior knowledge which is frequently available in practical applications, it is still restrictive since it assumes that the probabilities do not vary through time. The Known Adversarial Arrival KNOWNAD model (also known as prophet inequality matching) on the other hand, takes into account the dynamic variation in the probabilities. Therefore, the probability a vertex of type v arrives in round t is $p_{v,t}$ instead of being constant for every round t. This model

is also well-established in the matching literature and has been used in a collection of papers such as [21, 78, 79, 80]. Despite the fact that the KNOWNAD model is well-motivated and richer than the KNOWNIID model it was not used in the one-sided online fair matching papers of [5, 70].

2.2.2 Patience Parameters

The patience parameter of a vertex Δ_u (or Δ_v) for an offline vertex u (or an online vertex v) models its tolerance for unsuccessful probes (match attempts) before leaving the system. We note that this is an important detail in the online matching model since it is frequently the case that the vertices in the online matching applications (such as advertising, crowdsourcing, and ridesharing) represent human participants who would only tolerate a fixed number of failed matching attempts before leaving the system. Like the KNOWNIID and KNOWNAD arrival models, the patience parameter is also well-established in online matching, see for example [24, 73, 81]. Despite the importance of this parameter, the previous work in fair online matching did not consider the patience issue for both sides simultaneously [5, 70], handling both parameters at the same time is more challenging and leads to more tedious analysis.

We further elaborate on the meaning of the patience for both the online and offline sides, we note again that this is following the research literature on online matching:

Offline Patience: Consider a vertex u with patience Δ_u, then vertex u will remain on the offline side U unless it is successfully matched or it receives Δ_u many failed matching attempts. As a concrete example, consider a vertex u_1 with patience $\Delta_{u_1} = 2$. Clearly, in the first round ($t = 1$) u_1 will be in the offline side U, suppose an unsuccessful matching attempt (unsuccessful probe) is made in this round, then in the next round u_1 will still be there. Suppose that the next round when u_1 is probed is

in the fifth round ($t = 5$), then if the probe is successful then u_1 is matched and will be removed from the offline side in the next rounds ($t > 5$), but also if the match is unsuccessful then u_1 will not be matched but will still be removed for all of the next rounds ($t > 5$) since it has a patience $\Delta_{u_1} = 2$ and therefore can only take two failed matching attempts before leaving.

Online Patience: Unlike the offline side, an online vertex v would arrive in a round t and must be matched or rejected in that given round. While in a round t we can at most match one online vertex (which is the arriving vertex v) to some offline vertex u, we can make multiple match attempts (probes) from v to the vertices it is connected to in U in that round t. The patience Δ_v of v decides the upper limit on the number of failed attempts we can make in round t before v leaves the system and can no longer be matched even if a possible match was still not attempted. As a concrete example, suppose vertex of type v_1 with $\Delta_{v_1} = 3$ arrives in round $t = 7$ and that v_1 is connected to a total of four vertices $\{u_1, u_2, u_3, u_4\}$ in U all of which are still available (i.e. unmatched and still have not ran out of patience), suppose we make match attempts (probes) to u_1 then u_2 then u_3, it follows since $\Delta_{v_1} = 3$ that v_1 has left the system and we can no longer even attempt to match it to u_4 despite that fact that its available. Further, if at any probe attempt v_1 was matched then no further probe attempts are made to v_1, e.g. if the first probe (v_1 to u_1) in the above discussion was successful, then v_1 and u_1 are matched to each other and we cannot attempt to match v_1 to $u_2, u_3,$ or u_4.

2.3 Main Results

Performance Criterion: We note that we are in the online setting, therefore our performance criterion is the competitive ratio. Denote by $\mathcal{I}$ the distribution for the instances of matching problems, then $\mathrm{OPT}(\mathcal{I}) = \mathbb{E}_{I\sim\mathcal{I}}[\mathrm{OPT}(I)]$ where $\mathrm{OPT}(I)$ is the optimal value of the sampled instance I. Similarly,

for a given algorithm ALG, we define the value of its objective over the distribution $\mathcal{I}$ by $\mathrm{ALG}(\mathcal{I}) = \mathbb{E}_{\mathcal{D}}[\mathrm{ALG}(I)]$ where the expectation $\mathbb{E}_{\mathcal{D}}[.]$ is over the randomness of the instance and the algorithm. The competitive ratio is then defined as $\min_{\mathcal{I}} \frac{\mathrm{ALG}(\mathcal{I})}{\mathrm{OPT}(\mathcal{I})}$.

In our work, we address optimality guarantees for each of the three sides of the matching market by providing algorithms with competitive ratio guarantees for the operator's profit and the fairness objectives of the static and online side of the market simultaneously. Specifically, for the KNOWNIID arrival setting we have:

Theorem 2.3.1. *For the* KNOWNIID *setting, algorithm* $\mathrm{TSGF}_{\textsc{KnownIID}}(\alpha, \beta, \gamma)$ *achieves a competitive ratio of* $(\frac{\alpha}{2e}, \frac{\beta}{2e}, \frac{\gamma}{2e})$[2] *simultaneously over the operator's profit, the group fairness objective for the offline side, and the group fairness objective for the online side, where* $\alpha, \beta, \gamma > 0$ *and* $\alpha + \beta + \gamma \leq 1$.

The following two theorems hold under the condition that $p_e = 1, \forall e \in E$. Specifically for the KNOWNAD setting we have:

Theorem 2.3.2. *For the* KNOWNAD *setting, algorithm* $\mathrm{TSGF}_{\textbf{\textit{KnownAD}}}(\alpha, \beta, \gamma)$ *achieves a competitive ratio of* $(\frac{\alpha}{2}, \frac{\beta}{2}, \frac{\gamma}{2})$ *simultaneously over the operator's profit, the group fairness objective for the offline side, and the group fairness objective for the online side, where* $\alpha, \beta, \gamma > 0$ *and* $\alpha + \beta + \gamma \leq 1$.

Moreover, for the case of individual fairness whether in the KNOWNIID or KNOWNAD arrival setting we have:

Theorem 2.3.3. *For the* KNOWNIID *or* KNOWNAD *setting, we can achieve a competitive ratio of* $(\frac{\alpha}{2}, \frac{\beta}{2}, \frac{\gamma}{2})$ *simultaneously over the operator's profit, the individual fairness objective for the offline side, and the individual fairness objective for the online side, where* $\alpha, \beta, \gamma > 0$ *and* $\alpha + \beta + \gamma \leq 1$.

[2]Here, e denotes the Euler number, not an edge in the graph.

We also give the following hardness results. In particular, for a given arrival (KNOWNIID or KNOWNAD) setting and fairness criterion (group or individual), the competitive ratios for all sides cannot exceed 1 simultaneously:

Theorem 2.3.4. *For all arrival models, given the three objectives: operator's profit, offline side group (individual) fairness, and online side group (individual) fairness. No algorithm can achieve a competitive ratio of* (α, β, γ) *over the three objectives simultaneously such that* $\alpha + \beta + \gamma > 1$.

It is natural to wonder if we can combine individual and group fairness. Though it is possible to extend our algorithms to this setting. The follow theorem shows that they can conflict with one another:

Theorem 2.3.5. *Ignoring the operator's profit and focusing either on the offline side alone or the online side alone. With* α_G *and* α_I *denoting the group and individual fairness competitive ratios, respectively. No algorithm can achieve competitive ratios* (α_G, α_I) *over the group and individual fairness objectives of one side simultaneously such that* $\alpha_G + \alpha_I > 1$.

Finally, we carry experiments on real-world datasets in Section 2.5.

2.4 Algorithms and Theoretical Guarantees

Our algorithms use linear programming (LP) based techniques [5, 70, 73, 82] where first a *benchmark* LP is set up to upper bound the optimal value of the problem, then an LP solution is sampled from to produce an algorithm with guarantees.

2.4.1 Group Fairness for the KIID Setting:

Before we discuss the details of the algorithm, we note that for a given vertex type $v \in V$, the expected arrival rate n_v could be greater than one. However, it is not difficult to modify the instance by "fragmenting" each type with $n_v > 1$ such that in the new instance $n_v = 1$ for each type. This can be done with the operator's profit, offline group fairness, and online group fairness having the same values. Therefore, in what remains for the KNOWNIID setting $n_v = 1, \forall v \in V$ and therefore for any round t, each vertex v arrives with probability $\frac{1}{T}$. It also follows that for a given group g, $\sum_{v \in V(g)} n_v = \sum_{v \in V(g)} 1 = |V(g)|$.

For each edge $e = (u, v) \in E$ we use x_e to denote the expected number of probes (i.e, assignments from u to type v not necessarily successful) made to edge e in the LP benchmark. We have a total of three LPs each having the same set of constraints of (2.5), but differing by the objective. For compactness we do not repeat these constraints and instead write them once. Specifically, LP objective (2.2) along with the constraints of (2.5) give the optimal benchmark value of the operator's profit. Similarly, with the same set of constraints (2.5) LP objective (2.3) and LP objective (2.4) give the optimal group max-min fair assignment for the offline and online sides, respectively. Note that the expected max-min objectives of (2.3) and (2.4), can be written in the form of a linear objective. For example, the max-min objective of (2.3) can be replaced with an LP with objective $\max \eta$ subject to the additional constraints that $\forall g \in \mathcal{G}$, $\eta \leq \frac{\sum_{u \in U(g)} \sum_{e \in E_u} w_e^U x_e p_e}{|U(g)|}$. Having introduced the LPs, we will use LP(2.2), LP(2.3), and LP(2.4) to refer to the platform's profit LP, the offline side group fairness LP, and the online side group fairness LP, respectively.

$$\max \sum_{e \in E} w_e^O x_e p_e \tag{2.2}$$

$$\max \min_{g \in \mathcal{G}} \frac{\sum_{u \in U(g)} \sum_{e \in E_u} w_e^U x_e p_e}{|U(g)|} \tag{2.3}$$

$$\max \min_{g \in \mathcal{G}} \frac{\sum_{v \in V(g)} \sum_{e \in E_v} w_e^V x_e p_e}{|V(g)|} \tag{2.4}$$

$$\text{s.t} \quad \forall e \in E : 0 \leq x_e \leq 1 \tag{2.5a}$$

$$\forall u \in U : \textstyle\sum_{e \in E_u} x_e p_e \leq 1 \tag{2.5b}$$

$$\forall u \in U : \textstyle\sum_{e \in E_u} x_e \leq \Delta_u \tag{2.5c}$$

$$\forall v \in V : \textstyle\sum_{e \in E_v} x_e p_e \leq 1 \tag{2.5d}$$

$$\forall v \in V : \textstyle\sum_{e \in E_v} x_e \leq \Delta_v \tag{2.5e}$$

Now we prove that LP(2.2), LP(2.3) and LP(2.4) indeed provide valid upper bounds (benchmarks) for the optimal solution for the operator's profit and expected max-min fairness for the offline and online sides of the matching.

Lemma 2.4.1. *For the* KNOWNIID *setting, the optimal solutions of LP (2.2), LP (2.3), and LP (2.4) are upper bounds on the expected optimal that can be achieved by any algorithm for the operator's profit, the offline side group fairness objective, and the online side group fairness objective, respectively.*

Proof. We follow a similar proof to that used in [73]. We shall focus on the operator's profit objective since the other objectives follow by very similar arguments. First, we note that LP(2.2) uses the

expected values of the problem parameters, i.e. if we consider a specific graph realization G, then let N_v^G be the number of arrival for vertex type v, then it follows that LP(2.2) uses the expected values since $\mathbb{E}_{\mathcal{I}}[N_v^G] = 1, \forall v \in V$ where $\mathbb{E}_{\mathcal{I}}[.]$ is an expectation over the randomness of the instance. We shall therefore refer to the value of LP(2.2) as $LP(\mathbb{E}_{\mathcal{I}}[G])$.

To prove that $LP(\mathbb{E}_{\mathcal{I}}(G))$ is a valid upper bound it suffices to show that $LP(\mathbb{E}_{\mathcal{I}}[G[) \geq \mathbb{E}_{\mathcal{I}}[LP(G)]$ where $LP(G)$ is the optimal LP value of a realized instance G and $\mathbb{E}_{\mathcal{I}}[LP(G)]$ is the expected value of that optimal LP value. Let us then consider a specific realization G', its corresponding LP would be the following:

$$\max \textstyle\sum_{e' \in E'} w_{e'}^{O} p_{e'} x_{e'} \tag{2.6}$$

$$\text{s.t} \quad \forall e' \in E' : 0 \leq x_{e'} \leq 1 \tag{2.7a}$$

$$\forall u \in U : \textstyle\sum_{e' \in E'_u} x_{e'} p_{e'} \leq 1 \tag{2.7b}$$

$$\forall u \in U : \textstyle\sum_{e' \in E'_u} x_{e'} \leq \Delta_u \tag{2.7c}$$

$$\forall v' \in V' : \textstyle\sum_{e' \in E'_{v'}} x_{e'} p_{e'} \leq 1 \tag{2.7d}$$

$$\forall v' \in V' : \textstyle\sum_{e' \in E'_{v'}} x_{e'} \leq \Delta_{v'} \tag{2.7e}$$

where V' is the realization of the online side. It is clear that for a given realization $G' = (U, V', E')$ the above LP(2.6) is an upper bound on the operator's objective value for that realization.

Now we prove that $LP(\mathbb{E}_{\mathcal{I}}[G]) \geq \mathbb{E}_{\mathcal{I}}[LP(G)]$. The dual of the LP for the realization G' is the

following:

$$\min \sum_{u \in U} (\alpha_u + \Delta_u \beta_u) + \sum_{v' \in V'} (\alpha_{v'} + \Delta_{v'} \beta_{v'}) + \sum_{(u,v')} \gamma_{u,v'} \tag{2.8}$$

$$\text{s.t.} \quad \forall u \in U, \forall v' \in V' :$$

$$\beta_u + \beta_{v'} + p_{(u,v')}(\alpha_u + \alpha_{v'}) + \gamma_{(u,v')} \geq w^O_{(u,v')} p_{(u,v')} \tag{2.9a}$$

$$\alpha_u, \alpha_{v'}, \beta_u, \beta_{v'}, \gamma_{(u,v')} \geq 0 \tag{2.9b}$$

Consider the graph with the expected number of arrival $\mathbb{E}_{\mathcal{I}}(G)$ it would have a dual of the above form, let $\vec{\alpha}^*, \vec{\beta}^*, \vec{\gamma}^*$ be the optimal solution of its corresponding dual. Then it follows by the strong duality of LPs that solution $\vec{\alpha}^*, \vec{\beta}^*, \vec{\gamma}^*$ would have a value of $LP(\mathbb{E}_{\mathcal{I}}[G])$. Now for the instance G', we shall use the following dual solution $\vec{\hat{\alpha}}, \vec{\hat{\beta}}, \vec{\hat{\gamma}}$ which is set as follows:

- $\forall u \in U : \hat{\alpha}_u = \alpha^*_u, \hat{\beta}_u = \alpha^*_u$.
- $\forall v' \in V'$ of type v: $\hat{\alpha}_{v'} = \alpha^*_v, \hat{\beta}_{v'} = \beta^*_v$.
- $\forall u \in U, \forall v' \in V'$ of type v: $\hat{\gamma}_{(u,v')} = \gamma^*_{(u,v)}$.

Note that the new solution $\vec{\hat{\alpha}}, \vec{\hat{\beta}}, \vec{\hat{\gamma}}$ is a feasible dual solution since it satisfies constraints 2.9a and 2.9b. By weak duality the value of the solution $\vec{\hat{\alpha}}, \vec{\hat{\beta}}, \vec{\hat{\gamma}}$ upper bounds $LP(G')$. Now if we were to denote the number of vertices of type v that arrived in instance G' by $n^{G'}_v$, then the value of the solution $\vec{\hat{\alpha}}, \vec{\hat{\beta}}, \vec{\hat{\gamma}}$

satisfies:

$$\sum_{u\in U}(\hat{\alpha}_u + \Delta_u\hat{\beta}_u) + \sum_{v'\in V'}(\hat{\alpha}_{v'} + \Delta_{v'}\hat{\beta}_{v'}) + \sum_{(u,v')}\hat{\gamma}_{u,v'}$$
$$= \sum_{u\in U}(\alpha^*_u + \Delta_u\beta^*_u) + \sum_{v\in V} n_v^{G'}(\alpha^*_v + \Delta_v\beta^*_v) + \sum_{(u,v)} n_v^{G'}\gamma^*_{u,v}$$
$$\geq LP(G')$$

Now taking the expectation, we get:

$$\mathbb{E}_{\mathcal{I}}[LP(G')]$$
$$\leq \mathbb{E}_{\mathcal{I}}\Big[\sum_{u\in U}(\hat{\alpha}_u + \Delta_u\hat{\beta}_u) + \sum_{v'\in V'}(\hat{\alpha}_{v'} + \Delta_{v'}\hat{\beta}_{v'}) + \sum_{(u,v')}\hat{\gamma}_{u,v'}\Big]$$
$$= \mathbb{E}_{\mathcal{I}}\Big[\sum_{u\in U}(\alpha^*_u + \Delta_u\beta^*_u) + \sum_{v\in V} n_v^{G'}(\alpha^*_v + \Delta_v\beta^*_v) + \sum_{(u,v)} n_v^{G'}\gamma^*_{u,v}\Big]$$
$$= \sum_{u\in U}(\alpha^*_u + \Delta_u\beta^*_u) + \sum_{v\in V}\mathbb{E}_{\mathcal{I}}[n_v^{G'}](\alpha^*_v + \Delta_v\beta^*_v) + \sum_{(u,v)}\mathbb{E}_{\mathcal{I}}[n_v^{G'}]\gamma^*_{u,v}$$
$$= \sum_{u\in U}(\alpha^*_u + \Delta_u\beta^*_u) + \sum_{v\in V}(\alpha^*_v + \Delta_v\beta^*_v) + \sum_{(u,v)}\gamma^*_{u,v}$$
$$= LP(\mathbb{E}_{\mathcal{I}}[G])$$

For the offline and online group fairness objectives, we use the same steps. The difference would be in the constraints of the dual program, however following a similar assignment as done from $\vec{\alpha}^*, \vec{\beta}^*, \vec{\gamma}^*$ to $\vec{\hat{\alpha}}, \vec{\hat{\beta}}, \vec{\hat{\gamma}}$ is sufficient to prove the lemma for both fairness objectives. □

Our algorithm makes use of the dependent rounding subroutine [83]. We mention the main properties of dependent rounding. In particular, given a fractional vector $\vec{x} = (x_1, x_2, \ldots, x_t)$ where

each $x_i \in [0,1]$, let $k = \sum_{i\in[t]} x_i$, dependent rounding rounds x_i (possibly fractional) to $X_i \in \{0,1\}$ for each $i \in [t]$ such that the resulting vector $\vec{X} = (X_1, X_2, X_3, \ldots, X_t)$ has the following properties: (1) **Marginal Distribution**: The probability that $X_i = 1$ is equal to x_i, i.e. $Pr[X_i = 1] = x_i$ for each $i \in [t]$. (2) **Degree Preservation**: Sum of X_i's should be equal to either $\lfloor k \rfloor$ or $\lceil k \rceil$ with probability one, i.e. $Pr[\sum_{i\in[t]} X_i \in \{\lfloor k \rfloor, \lceil k \rceil\}] = 1$. (3) **Negative Correlation**: For any $S \subseteq [t]$, (1) $Pr[\wedge_{i\in S} X_i = 0] \leq \Pi_{i\in S} Pr[X_i = 0]$ (2) $Pr[\wedge_{i\in S} X_i = 1] \leq \Pi_{i\in S} Pr[X_i = 1]$. It follows that for any $x_i, x_j \in \vec{x}$, $\mathbb{E}[X_i = 1 | X_j = 1] \leq x_i$.

Going back to the LPs (2.2,2.3,2.4), we denote the optimal solutions to LP (2.2), LP (2.3), and LP (2.4) by $\vec{x}^*$,$\vec{y}^*$ and $\vec{z}^*$ respectively. Further, we introduce the parameters $\alpha, \beta, \gamma \in [0,1]$ where $\alpha+\beta+\gamma \leq 1$ and each of these parameters decide the "weight" the algorithm places on each objective (the operator's profit, the offline group fairness, and the online group fairness objectives). We note that our algorithm makes use of the subroutine **PPDR** (Probe with Permuted Dependent Rounding) shown in Algorithm 1.

Algorithm 1 PPDR($\vec{x}_v$)

1: Apply dependent rounding to the fractional solution $\vec{x}_v$ to get a binary vector $\vec{X}_v$.
2: Choose a random permutation π over the set E_v.
3: **for** $i = 1$ to $|E_v|$ **do**
4: Probe vertex $\pi(i)$ if it is available and $\vec{X}_v(\pi(i)) = 1$
5: **if** Probe is successful (i.e., a match) **then**
6: break
7: **end if**
8: **end for**

The procedure of our parameterized sampling algorithm $\text{TSGF}_{\text{KNOWNIID}}$ is shown in Algorithm 2. Specifically, when a vertex of type v arrives at any time step we run **PPDR**$(\vec{x}^*_v)$, **PPDR**$(\vec{y}^*_v)$, or **PPDR**$(\vec{z}^*_v)$ with probabilities α, β, and γ, respectively. We do not run any of the **PPDR** subroutines and instead reject the vertex with probability $1-(\alpha+\beta+\gamma)$. The LP constraint (2.5e) guarantees

that $\forall v \in V : \sum_{e \in E_r} s^*_e \leq \Delta_v$ where $\vec{s}^*$ could be $\vec{x}^*, \vec{y}^*$, or $\vec{z}^*$. Therefore, when **PPDR** is invoked by the **degree preservation** property of dependent rounding the number of edges probed will not exceed Δ_v, i.e. it would be within the patience limit.

Algorithm 2 $\text{TSGF}_{\text{KnownIID}}(\alpha, \beta, \gamma)$

1: Let v be the vertex type arriving at time t.
2: With probability α run the subroutine, **PPDR**$\left(\vec{x}^*_v\right)$.
3: With probability β run the subroutine, **PPDR**$\left(\vec{y}^*_v\right)$.
4: With probability γ run the subroutine, **PPDR**$\left(\vec{z}^*_v\right)$.
5: Reject the arriving vertex with probability $1 - (\alpha + \beta + \gamma)$.

Now we analyze $\text{TSGF}_{\text{KnownIID}}$ to prove Theorem 2.3.1. It would suffice to prove that for each edge e the expected number of successful probes is at least $\alpha\frac{x^*_e}{2e}$, $\beta\frac{y^*_e}{2e}$ and $\gamma\frac{z^*_e}{2e}$. And finally from the linearity of expectation we show that the worst case competitive ratio of the proposed online algorithm with parameters α, β and γ is at least $(\frac{\alpha}{2e}, \frac{\beta}{2e}, \frac{\gamma}{2e})$ for the operator's profit and group fairness objectives on the offline and online sides of the matching, respectively.

A critical step is to lower bound the probability that a vertex u is available (safe) at the beginning of round $t \in [T]$. Let us denote the indicator random variable for that event by $SF_{u,t}$. The following lemma enables us to lower bound for the probability of $SF_{u,t}$.

Before we prove Lemma 2.4.4 for the lower bound on the probability of $SF_{u,t}$. We have to first introduce the following two lemmas. Specifically, let $A_{u,t}$ be the number of successful assignments that u received and accepted before round t. Then the following lemma holds.

Lemma 2.4.2. *For any given vertex u at time $t \in [T]$, $P[A_{u,t} = 0] \geq \left(1 - \frac{1}{T}\right)^{t-1}$.*

Proof. Let $X_{e,k}$ be the indicator random variable for u receiving an arrival request of type v where $e \in E_u$ and $k < t$. Let $Y_{e,k}$ be the indicator random variable that the edge e gets sampled by the $\text{TSGF}_{\text{KnownIID}}(\alpha, \beta, \gamma)$ algorithm at time $k < t$. Let $Z_{e,k}$ be the indicator random variable that as-

signment $e = (u, v)$ is successful (a match) at time $k < t$. Then $A_{u,t} = \sum_{k<t} \sum_{e \in E_u} X_{e,k} Y_{e,k} Z_{e,k}$.

$$
\begin{aligned}
Pr[A_{u,t} = 0] &= \Pi_{k<t} Pr\Big[\sum_{e=(u,v) \in E_u} X_{e,k} Y_{e,k} Z_{e,k} = 0 \Big] \\
&= \Pi_{k<t} \Big(1 - Pr\Big[\sum_{e \in E_u} X_{e,k} Y_{e,k} Z_{e,k} \geq 1 \Big] \Big) \\
&\geq \Pi_{k<t} \Big(1 - \sum_{e \in E_u} \frac{1}{T} \cdot \big(\alpha x_e^* + \beta \frac{y_e^*}{q_v} + \gamma \frac{z_e^*}{q_v} \big) \cdot p_e \Big) \\
&= \Pi_{k<t} \Big(1 - \frac{1}{T} \cdot \big(\alpha \sum_{e \in E_u} x_e^* p_e + \beta \sum_{e \in E_u} y_e^* p_e + \gamma \sum_{e \in E_u} z_e^* p_e \big) \Big) \\
&\geq \Pi_{k<t} \Big(1 - \frac{1}{T} \cdot \big(\alpha \cdot 1 + \beta \cdot 1 + \gamma \cdot 1 \big) \Big) \\
&\geq \Pi_{k<t} \Big(1 - \frac{1}{T} \Big) = \Big(1 - \frac{1}{T} \Big)^{t-1}
\end{aligned}
$$

□

Now we lower bound the probability that u was probed less than Δ_u times prior to t. Denote the number of probes received by u before t by $B_{u,t}$, then the following lemma holds:

Lemma 2.4.3. $Pr[B_{u,t} < \Delta_u] \geq 1 - \frac{t-1}{T}$.

Proof. First it is clear that $B_{u,t} = \sum_{k<t} \sum_{e \in E_u} X_{e,k} Y_{e,k}$.

$$
\begin{aligned}
\mathbb{E}[B_{u,t}] &= \sum_{k<t} \sum_{e \in E_u} \mathbb{E}[X_{e,k} Y_{e,k}] \\
&\leq \sum_{k<t} \sum_{e \in E_u} \frac{1}{T} \Big(\alpha x_e^* + \beta y_e^* + \gamma z_e^* \Big) \\
&\leq \sum_{k<t} \frac{1}{T} \Big(\alpha \sum_{e \in E_d} x_e^* + \beta \sum_{e \in E_u} y_e^* + \gamma \sum_{e \in E_u} z_e^* \Big) \\
&\leq \sum_{k<t} \frac{\Delta_u}{T} (\alpha + \beta + \gamma) \leq \frac{(t-1)\Delta_u}{T}
\end{aligned}
$$

The inequality before the last follows from $(\alpha + \beta + \gamma) \leq 1$. Now using Markov's inequality: $Pr[B_{u,t} < \Delta_u] \geq 1 - \frac{\mathbb{E}[B_{u,t}]}{\Delta_u}$, we get $\implies Pr[B_{u,t} < \Delta_u] \geq 1 - \frac{t-1}{T}$. □

Lemma 2.4.4. $Pr[SF_{u,t}] \geq \Big(1 - \frac{t-1}{T}\Big)\Big(1 - \frac{1}{T}\Big)^{t-1}$.

Proof. Consider a given edge $e \in E_u$ where $k < t$

$$
\begin{aligned}
&\mathbb{E}[X_{e,k} Y_{e,k} \mid A_{u,t} = 0] = \mathbb{E}[X_{e,k} Y_{e,k} \mid A_{u,k} = 0] \\
&= \frac{Pr[X_{e,k} = 1, Y_{e,k} = 1, Z_{e,k} = 0]}{Pr[A_{u,k} = 0]} \\
&\leq \frac{\frac{1}{T} \cdot \big(\alpha x_e^* + \beta y_e^* + \gamma z_e^*\big) \cdot (1 - p_e)}{1 - \sum_{e \in E_d} \frac{1}{T} \cdot \big(\alpha x_e^* + \beta y_e^* + \gamma z_e^*\big) \cdot p_e} \\
&= \frac{\frac{1}{T} \cdot \big(\alpha x_e^* + \beta y_e^* + \gamma z_e^*\big) \cdot (1 - p_e)}{1 - p_e + p_e \Big(1 - \sum_{e \in E_d} \frac{1}{T} \cdot \big(\alpha x_e^* + \beta y_e^* + \gamma z_e^*\big)\Big)} \\
&\leq \frac{1}{T} \cdot \big(\alpha x_e^* + \beta y_e^* + \gamma z_e^*\big) \cdot
\end{aligned}
$$

The above inequality is due to the fact that $\sum_{e \in E_u} \frac{1}{T}\big(\alpha x_e^* + \beta y_e^* + \gamma z_e^*\big) \leq \frac{\Delta_u}{T} < 1$.

$$\begin{aligned}
\mathbb{E}[B_{u,t}|A_{u,t} = 0] &= \sum_{k<t} \sum_{e \in E_u} \mathbb{E}[X_{e,k} Y_{e,k} | A_{u,k} = 0] \\
&\leq \sum_{k<t} \sum_{e \in E_u} \frac{1}{T}\Big(\alpha x_e^* + \beta y_e^* + \gamma z_e^*\Big) \\
&\leq \sum_{k<t} \frac{1}{T}\Big(\alpha \sum_{e \in E_u} x_e^* + \beta \sum_{e \in E_u} y_e^* + \gamma \sum_{e \in E_u} z_e^*\Big) \\
&\leq \sum_{k<t} \frac{1}{T}\Big(\alpha \cdot \Delta_u + \beta \cdot \Delta_d + \gamma \cdot \Delta_u\Big) \\
&= \sum_{k<t} \frac{\Delta_u}{T}(\alpha + \beta + \gamma) \leq \frac{(t-1)\Delta_u}{T}
\end{aligned}$$

Therefore the expected number of assignments (probes) to vertex u until time t is at most $\frac{(t-1)\Delta_u}{T}$. Therefore, we have:

$$\begin{aligned}
Pr[B_{u,t} < \Delta_u | A_{u,t} = 0] &\geq 1 - \frac{\mathbb{E}[B_{u,t}|A_{u,t} = 0]}{\Delta_d} \\
&\geq 1 - \frac{(t-1)\Delta_u}{T\Delta_u} \geq 1 - \frac{t-1}{T}
\end{aligned}$$

It is to be noted that $B_{u,t}$ is the total number of probes u received before round t. Thus, we have that the events $(B_{u,t} < \Delta_u)$ and $(A_{u,t} = 0)$ are positively correlated. Therefore,

$$\begin{aligned}
Pr[SF_{u,t}] &\geq Pr[(B_{u,t} < \Delta_u) \wedge (A_{u,t} = 0)] \\
&\geq Pr[B_{u,t} < \Delta_d | A_{u,t} = 0] Pr[A_{u,t} = 0] \\
Pr[SF_{u,t}] &\geq \Big(1 - \frac{t-1}{T}\Big)\Big(1 - \frac{1}{T}\Big)^{t-1}
\end{aligned}$$

□

. Now that we have established a lower bound on $Pr[SF_{u,t}]$, we lower bound the probability that an edge e is probed by one of the **PPDR** subroutines conditioned on the fact that u is available (Lemma 2.4.5). Let $1_{e,t}$ be the indicator that $e = (u, v)$ is probed by the $\text{TSGF}_{\text{KNOWNIID}}$ Algorithm at time t. Note that event $1_{e,t}$ occurs when (1) a vertex of type v arrives at time t and (2) e is sampled by **PPDR**($\vec{x_v}$), **PPDR**($\vec{y_v}$), or **PPDR**($\vec{z_v}$).

Lemma 2.4.5. $Pr[1_{e,t} \mid SF_{u,t}] \geq \alpha\frac{x^*_e}{2T}$,$Pr[1_{e,t} \mid SF_{u,t}] \geq \beta\frac{y^*_e}{2T}$, $Pr[1_{e,t} \mid SF_{u,t}] \geq \gamma\frac{z^*_e}{2T}$

Proof. In this part we prove that the probability that edge e is probed at time t is at least $\alpha\frac{x^*_e}{2T}$. Note that the probability that a vertex of type v arrives at time t and that Algorithm 2 calls the subroutine **PPDR**($\vec{x_r}$) is $\alpha\frac{1}{T}$. Let $E_{v,\bar{e}}$ be the set of edges in E_v excluding $e = (u, v)$. For each edge $e' \in E_{v,\bar{e}}$ let $Y_{e'}$ be the indicator for e' being before e in the random order of π (in algorithm 1) and let $Z_{e'}$ be the probability that the assignment is successful for e'. It is clear that $\mathbb{E}[Y_{e'}] = 1/2$ and that $\mathbb{E}[Z_{e'}] = p_{e'}$. Now we have:

$$Pr[1_{e,t} \mid SF_{u,t}] \tag{2.10}$$

$$\geq \alpha\frac{1}{T}Pr[X_e = 1]Pr\big[\sum_{e'\in E_{r,\bar{e}}} X_{e'}Y_{e'}Z_{e'} \mid X_e = 1\big] \tag{2.11}$$

$$= \alpha\frac{Pr[X_e = 1]}{T}\big(1 - Pr\big[\sum_{e'\in E_{v,\bar{e}}} X_{e'}Y_{e'}Z_{e'} \geq 1 \mid X_e = 1\big]\big) \tag{2.12}$$

$$\geq \alpha\frac{Pr[X_e = 1]}{T}\big(1 - \mathbb{E}\big[\sum_{e'\in E_{v,\bar{e}}} X_{e'}Y_{e'}Z_{e'} \geq 1 \mid X_e = 1\big]\big) \tag{2.13}$$

$$\geq \alpha\frac{Pr[X_e = 1]}{T}\big(1 - \sum_{e'\in E_{v,\bar{e}}} \mathbb{E}\big[X_{e'}Y_{e'}Z_{e'} \geq 1 \mid X_e = 1\big]\big) \tag{2.14}$$

$$\geq \alpha\frac{x^*_e}{T}\big(1 - \sum_{e'\in E_{v,\bar{e}}} x^*_{e'}\frac{1}{2}p_{e'}\big) \tag{2.15}$$

$$\geq \alpha\frac{x^*_e}{T}\big(1 - \frac{1}{2}\big) = \alpha\frac{x^*_e}{2T} \tag{2.16}$$

Applying Markov inequality we get the inequality (2.13). By linearity of expectation we get inequality (2.14). Since X_e and $X_{e'}$ are negatively correlated to each other from the Negative Correlation property of Dependent Rounding we have $\mathbb{E}[X_{e'} \mid X_e = 1] \leq x_e^*$ and we get (2.15). The last inequality (2.16) is due the fact that for any feasible solution $\{x_e^*\}$ the constraints imply that $\sum_{e \in E_v} x_e^* p_e \leq 1$ for all $v \in V$. Using similar analysis we can also prove that $Pr[1_{e,t} \mid SF_{u,t}] \geq \beta \frac{y_e^*}{2T}$ and $Pr[1_{e,t} \mid SF_{u,t}] \geq \gamma \frac{z_e^*}{2T}$. □

Given the above lemmas Theorem 2.3.1 can be proved. We restate the theorem below.

Theorem 2.3.1. *For the* KNOWNIID *setting, algorithm* $\text{TSGF}_{\text{KNOWNIID}}(\alpha, \beta, \gamma)$ *achieves a competitive ratio of* $(\frac{\alpha}{2e}, \frac{\beta}{2e}, \frac{\gamma}{2e})$[2] *simultaneously over the operator's profit, the group fairness objective for the offline side, and the group fairness objective for the online side, where* $\alpha, \beta, \gamma > 0$ *and* $\alpha + \beta + \gamma \leq 1$.

Proof. Denote the expected number of probes on each edge $e \in E$ resulting from $\textbf{PPDR}(\vec{x}_v^*)$ by n_e^x. It follows that:

$$n_e^x \geq \sum_{t=1}^{T} Pr[1_{e,t}] = \sum_{t=1}^{T} Pr[1_{e,t} \mid SF_{u,t}] Pr[SF_{u,t}]$$
$$\geq \sum_{t=1}^{T} \Big(1 - \frac{1}{T}\Big)^{t-1} \Big(1 - \frac{t-1}{T}\Big) \Big(\alpha \frac{x_e^*}{2T}\Big) \xrightarrow{T \to \infty} \frac{\alpha x_e^*}{2e}$$

Denote the optimal solution for the operator's profit LP by OPT_O. Let ALG_O be operator's profit obtained by our online algorithm. Using the linearity of expectation we get:

$$ALG_O = \mathbb{E}\Big[\sum_{e \in E} w_e^O n_e^x p_e\Big] \geq \sum_{e \in E} w_e^O p_e \frac{\alpha x_e^*}{2e} \geq \sum_{e \in E} w_e^O p_e \Big(\frac{1}{e}\Big) \frac{\alpha x_e^*}{2} \geq \frac{\alpha}{2e}(OPT_O).$$

Similarly, we can obtain $\frac{\beta}{2e}$ and $\frac{\gamma}{2e}$ competitive ratios for the expected max-min group fairness guarantees on the offline and online sides, respectively. □

[2]Here, e denotes the Euler number, not an edge in the graph.

2.4.2 Group Fairness for the KAD Setting:

For the KNOWNAD setting, the distribution over V is time dependent and hence the probability of sampling a type v in round t is $p_{v,t} \in [0,1]$ with $\sum_{v \in V} p_{v,t} = 1$. Further, we assume for the KNOWNAD setting that for every edge $e \in E$ we have $p_e = 1$. This means that whenever an incoming vertex v is assigned to a safe-to-add vertex u the assignment is successful. This also means that any non-trivial values for the patience parameters Δ_u and Δ_v become meaningless and hence we can WLOG assume that $\forall u \in U, \forall v \in V, \Delta_u = \Delta_v = 1$. From the above discussion, we have the following LP benchmarks for the operator's profit, the group fairness for the offline side and the group fairness for the online side:

$$\max \sum_{t \in [T]} \sum_{e \in E} w_e^O x_{e,t} \tag{2.17}$$

$$\max \min_{g \in \mathcal{G}} \frac{\sum_{t \in [T]} \sum_{u \in U(g)} \sum_{e \in E_u} w_e^U x_{e,t}}{|U(g)|} \tag{2.18}$$

$$\max \min_{g \in \mathcal{G}} \frac{\sum_{t \in [T]} \sum_{v \in V(g)} \sum_{e \in E_v} w_e^V x_{e,t}}{\sum_{v \in V(g)} n_v} \tag{2.19}$$

$$\text{s.t} \quad \forall e \in E, \forall t \in [T] : 0 \leq x_{e,t} \leq 1 \tag{2.20a}$$

$$\forall u \in U : \sum_{t \in [T]} \sum_{e \in E_u} x_{e,t} \leq 1 \tag{2.20b}$$

$$\forall v \in V, \forall t \in [T] : \textstyle\sum_{e \in E_v} x_{e,t} \leq p_{v,t} \tag{2.20c}$$

Lemma 2.4.6. *For the* KNOWNAD *setting, the optimal solutions of LP (2.17), LP (2.18) and LP (2.19) are upper bounds on the expected optimal that can be achieved by any algorithm for the operator's profit, the offline side group fairness objective, and the online side group fairness objective, respectively.*

Proof. We shall consider only the operator's profit objective as the other objectives follow through an identical argument. Let $1_{v,t}$ be the indicator random variable for the arrival for vertex type v in round t. Then we can obtain a realization and solve the corresponding LP and then take the expected value of LP as an upper bound on the operator's profit objective, i.e. the value $\mathbb{E}_{\mathcal{I}}[LP(G)]$ where $\mathbb{E}_{\mathcal{I}}$ is an expectation with respect to the randomness of the problem. This means replacing $1_{v,t}$ by its realization in the LP below:

$$\max \sum_{t\in[T]} \sum_{e\in E} w_e^O x_{e,t} \tag{2.21}$$

$$\text{s.t} \quad \forall e \in E, \forall t \in [T] : 0 \leq x_{e,t} \leq 1 \tag{2.22a}$$

$$\forall u \in U : \sum_{t\in[T]} \sum_{e\in E_u} x_{e,t} \leq 1 \tag{2.22b}$$

$$\forall v \in V, \forall t \in [T] : \textstyle\sum_{e\in E_v} x_{e,t} \leq 1_{v,t} \tag{2.22c}$$

If we were to replace the random variables $1_{v,t}$ by their expected value, then we would retrieve LP(2.17) where $\mathbb{E}_{\mathcal{I}}[1_{v,t}] = p_{v,t}$. It suffices to show that the value of LP(2.17) which is the LP value over the "expected" graph (the parameters replaced by their expected value) which we now denote by $LP(\mathbb{E}_{\mathcal{I}}[G])$ is an upper bound to $\mathbb{E}_{\mathcal{I}}[LP(G)]$, i.e. $LP(\mathbb{E}_{\mathcal{I}}[G]) \geq \mathbb{E}_{\mathcal{I}}[LP(G)]$. Let $x_{e,t}^{*,G}$ be the optimal

solution for a given realization G and $1^G_{v,t}$ be the realization of the random variables over the instance, then we have that $\sum_{e\in E_v} x^{*,G}_{e,t} \leq 1^G_{v,t}$. It follows that $\mathbb{E}_{\mathcal{I}}[x^{*,G}_{e,t}]$ is a feasible solution for LP(2.17), since $\mathbb{E}_{\mathcal{I}}[\sum_{e\in E_v} x^{*,G}_{e,t}] \leq \mathbb{E}_{\mathcal{I}}[1^G_{v,t}] = p_{v,t}$ and the rest of the constraints are satisfied as well since they are the same in every realization. However, we have that $\mathbb{E}_{\mathcal{I}}[LP(G)] = \mathbb{E}_{\mathcal{I}}[\sum\limits_{t\in[T]}\sum\limits_{e\in E} w^O_e x^{*,G}_{e,t}] = \sum\limits_{t\in[T]}\sum\limits_{e\in E} w^O_e \mathbb{E}_{\mathcal{I}}[x^{*,G}_{e,t}] \leq \sum\limits_{t\in[T]}\sum\limits_{e\in E} w^O_e x^*_{e,t} = LP(\mathbb{E}_{\mathcal{I}}[G])$ where $x^*_{e,t}$ is the optimal solution for LP(2.17) over the "expected" graph. The inequality followed since a feasible solution to a problem cannot exceed its optimal solution. □

Note that in the above LP we have $x_{e,t}$ as the probability for successfully assigning an edge in round t (with an explicit dependence on t), unlike in the KNOWNIID setting where we had x_e instead to denote the expected number of times edge e is probed through all rounds. Similar to our solution for the KNOWNIID setting, we denote by $x^*_{e,t}$, $y^*_{e,t}$, and $z^*_{e,t}$ the optimal solutions of the LP benchmarks for the operator's profit, offline side group fairness, and online side group fairness, respectively.

Having the optimal solutions to the LPs, we use algorithm $\text{TSGF}_{\textbf{KnownAD}}$ shown in Algorithm 3. In $\text{TSGF}_{\textbf{KnownAD}}$ new parameters are introduced, specifically λ and $\rho_{e,t}$ where $\rho_{e,t}$ is the probability that edge $e = (u, v)$ is safe to add in round t, i.e. the probability that u is unmatched at the beginning of round t. For now we assume that we have the precise values of $\rho_{e,t}$ for all rounds and discuss how to obtain these values at the end of this subsection. Now conditioned on v arriving at round t and $e = (u, v)$ being safe to add, it follows that e is sampled with probability $\alpha\frac{x^*_{e,t}}{p_{v,t}}\frac{\lambda}{\rho_{e,t}} + \beta\frac{y^*_{e,t}}{p_{v,t}}\frac{\lambda}{\rho_{e,t}} + \gamma\frac{z^*_{e,t}}{p_{v,t}}\frac{\lambda}{\rho_{e,t}}$ which would be a valid probability (positive and not exceeding 1) if $\rho_{e,t} \geq \lambda$. This follows from the fact that $\alpha, \beta, \gamma \in [0, 1]$ and $\alpha+\beta+\gamma \leq 1$ and also by constraint (2.20c) which leads to $\frac{\sum_{e\in E_v} x_{e,t}}{p_{v,t}} \leq 1$. Further, if $\rho_{e,t} \geq \lambda$ then by constraint (2.20c) we have $\sum_{e\in E_v}\left(\alpha\frac{x^*_{e,t}}{p_{v,t}}\frac{\lambda}{\rho_{e,t}} + \beta\frac{y^*_{e,t}}{p_{v,t}}\frac{\lambda}{\rho_{e,t}} + \gamma\frac{z^*_{e,t}}{p_{v,t}}\frac{\lambda}{\rho_{e,t}}\right) \leq 1$ and therefore the distribution is valid. Clearly, the value of λ is important for the validity of the algorithm,

the following lemma shows that $\lambda = \frac{1}{2}$ leads to a valid algorithm.

Lemma 2.4.7. *Algorithm* $\text{TSGF}_{\textbf{\textit{KnownAD}}}$ *is valid for* $\lambda = \frac{1}{2}$.

Proof. We prove the validity of the algorithm for $\lambda = \frac{1}{2}$ by induction. For the base case, it is clear that $\forall e \in E, \rho_{e,t} = 1$, hence $\rho_{e,t} \geq \lambda = \frac{1}{2}$. Assume for $t' < t$, that $\rho_{e,t'} \geq \lambda = \frac{1}{2}$, then at round t we have:

$$
\begin{aligned}
1 - \rho_{e,t} &= \Pr[e \text{ is not available at } t] \\
&= \Pr[e \text{ is matched in } [T-1]] \\
&\leq \sum_{t'<t} \Pr[e \text{ is matched in } t'] \\
&= \sum_{t'<t} \Pr[(e \text{ is chosen by the algorithm}) \wedge (u \text{ is unmatched at } t) \wedge (v \text{ arrives at } t)] \\
&= \sum_{t'<t} p_{v,t}\rho_{e,t}(\alpha \frac{x^*_{e,t}}{p_{v,t}} \frac{\lambda}{\rho_{e,t}} + \beta \frac{y^*_{e,t}}{p_{v,t}} \frac{\lambda}{\rho_{e,t}} + \gamma \frac{z^*_{e,t}}{p_{v,t}} \frac{\lambda}{\rho_{e,t}}) \\
&= \sum_{t'<t} \lambda(\alpha x^*_{e,t'} + \beta y^*_{e,t'} + \gamma z^*_{e,t'}) \\
&\leq \lambda \sum_{t'<t} (\alpha x^*_{e,t'} + \beta y^*_{e,t'} + \gamma z^*_{e,t'}) \\
&\leq \lambda(\alpha + \beta + \gamma) \leq \lambda \leq \frac{1}{2}
\end{aligned}
$$

where we used the fact that $x^*_{e,t'}, y^*_{e,t'}, z^*_{e,t'} \leq 1$ from constraint (2.20a) and the fact that $\alpha + \beta + \gamma \leq 1$. From the above, it follows that $\rho_{e,t} \geq \frac{1}{2} \geq \lambda$. □

We now return to the issue of how to obtain the values of $\rho_{e,t}$ for all rounds. This can be done by using the simulation technique as done in [21, 81]. To elaborate, we note that we first solve the LPs (2.17,2.18,2.19) and hence have the values of $x^*_{e,t}$, $y^*_{e,t}$, and $z^*_{e,t}$. Now, for the first round $t = 1$, clearly $\rho_{e,t} = 1, \forall e \in E$. To obtain $\rho_{e,t}$ for $t = 2$, we simulate the arrivals and algorithm a collection

Algorithm 3 $\text{TSGF}_{\textbf{KnownAD}}(\alpha, \beta, \gamma)$

Let v be the vertex type arriving at time t.
if $E_{v,t} = \phi$ **then**
  Reject v
else
  With probability α probe e with probability $\frac{x^*_{e,t}}{p_{v,t}} \frac{\lambda}{\rho_{e,t}}$.
  With probability β probe e with probability $\frac{y^*_{e,t}}{p_{v,t}} \frac{\lambda}{\rho_{e,t}}$.
  With probability γ probe e with probability $\frac{z^*_{e,t}}{p_{v,t}} \frac{\lambda}{\rho_{e,t}}$.
  With probability $[1 - (\alpha + \beta + \gamma)]$ reject v .
end if

of times, and use the empirically estimated probability. More precisely, for $t = 1$ we sample the arrival of vertex v from $p_{v,t}$ with $t = 1$ ($p_{v,t}$ values are given as part of the model), then we run our algorithm for the values of α, β, γ that we have chosen. Accordingly, at $t = 2$ some vertex in U might be matched. We do this simulation a number of times and then we take $\rho_{e,t}$ for $t = 2$ to be the average of all runs. Now having the values of $\rho_{e,t}$ for $t = 1$ and $t = 2$, we further simulate the arrivals and the algorithm to obtain $\rho_{e,t}$ for $t = 3$ and so on until we get $\rho_{e,t}$ for the last round T. We note that using the Chernoff bound [84] we can rigorously characterize the error in this estimation, however by doing this simulation a number of times that is polynomial in the problem size, the error in the estimation would only affect the lower order terms in the competitive ration analysis [21] and hence for simplicity it is ignored. Now, with Lemma 2.4.7 Theorem 2.3.2 can be proved as shown below.

Theorem 2.3.2. *For the* KNOWNAD *setting, algorithm* $\text{TSGF}_{\textbf{KnownAD}}(\alpha, \beta, \gamma)$ *achieves a competitive ratio of* $(\frac{\alpha}{2}, \frac{\beta}{2}, \frac{\gamma}{2})$ *simultaneously over the operator's profit, the group fairness objective for the offline side, and the group fairness objective for the online side, where* $\alpha, \beta, \gamma > 0$ *and* $\alpha + \beta + \gamma \leq 1$.

Proof. For an edge e the probability that it is matched (successfully probed) is the following:

$$\begin{aligned}
&\Pr[e \text{ is successfully probed in round } t] \\
&= \Pr[(e \text{ is chosen by the algorithm}) \\
&\wedge (u \text{ is unmatched at the beginning of } t) \wedge (v \text{ arrives at } t)] \\
&= p_{v,t}\rho_{e,t}(\alpha\frac{x^*_{e,t}}{p_{v,t}}\frac{\lambda}{\rho_{e,t}} + \beta\frac{y^*_{e,t}}{p_{v,t}}\frac{\lambda}{\rho_{e,t}} + \gamma\frac{z^*_{e,t}}{p_{v,t}}\frac{\lambda}{\rho_{e,t}}) = \\
&= \alpha\lambda x^*_{e,t} + \beta\lambda y^*_{e,t} + \gamma\lambda z^*_{e,t}
\end{aligned}$$

Setting $\lambda = \frac{1}{2}$, it follows from the above that e is successfully matched with probability at least $\frac{1}{2}\alpha x^*_{e,t}$, at least $\frac{1}{2}\beta y^*_{e,t}$, and at least $\frac{1}{2}\gamma z^*_{e,t}$. Hence, the guarantees on the competitive ratios follow by linearity of the expectation. □

2.4.3 Individual Fairness KIID and KAD Settings:

For the case of Rawlsian (max-min) individual fairness, we consider each vertex of the offline side to belong to its own distinct group and the definition of group max-min fairness would lead to individual max-min fairness. On the other hand, for the online side a similar trick would not yield a meaningful criterion, we instead define the individual max-min fairness for the online side to equal $\min_{t\in[T]} \mathbb{E}[\text{util}(v_t)] = \min_{t\in[T]} \mathbb{E}[\sum_{e\in\mathcal{M}_{v_t}} w^V_e)]$ where util(v_t) is the utility received by the vertex arriving in round t. If we were to denote by $x_{e,t}$ the probability that the algorithm would successfully match e in round t, then it follows straightforwardly that $\mathbb{E}[\text{util}(v_t)] = \sum_{e\in E_{v_t}} w^V_e x_{e,t}$. We consider this definition to be the valid extension of max-min fairness for the online side as we are now concerned with the

minimum utility across the online individuals (arriving vertices) which are T many. The following lemma shows that we can solve two-sided individual max-min fairness by a reduction to two-sided group max-min fairness in the KNOWNAD arrival setting:

Lemma 2.4.8. *Whether in the* KNOWNIID *or* KNOWNAD *setting, a given instance of two-sided individual max-min fairness can be converted to an instance of two-sided group max-min fairness in the* KNOWNAD *setting.*

Proof. Given an instance with individual fairness, define $\mathcal{G} = \{g_1, \ldots, g_T\} \cup \{g'_1, \ldots, g'_{|U|}\}$ as the set of all groups, thus $|\mathcal{G}| = T + |U|$, i.e. one group for each time round and one group for each offline vertex. Further given the online side types V, create a new online side V' where $|V'| = T|V|$ and $V' = V'_1 \cup V'_2 \cdots \cup V'_t \cdots \cup V'_T$ where V'_t consists of the same types as V. Moreover, $\forall v' \in V'_t, p_{v',t} = p_{v,t}$ and $p_{v',\bar{t}} = 0, \forall \bar{t} \in [T] - \{t\}$, finally $\forall v' \in V'_t, g(v') = g_t$. For the offline side U, we let each vertex have its own distinct group membership, i.e. for vertex $u_i \in U$, $g(u_i) = g'_i$.

Based on the above, it is not difficult to see that both problems have the same operator profit, and that the individual max-min fairness objectives of the original instance equal the group max-min fairness objectives of the new instance. □

From the above Lemma, applying algorithm $\text{TSGF}_{\textbf{KnownAD}}$ to the reduced instance leads to the following corollary:

Corollary 2.4.9. *Given an instance of two-sided individual max-min fairness, applying* $\text{TSGF}_{\textbf{KnownAD}}$ (α, β, γ) *to the reduction from Theorem 2.4.8 leads to a competitive ratio of* $(\frac{\alpha}{2}, \frac{\beta}{2}, \frac{\gamma}{2})$ *simultaneously over the operator's profit, the individual fairness objective for the offline side, and the individual fairness objective for the online side, where* $\alpha, \beta, \gamma > 0$ *and* $\alpha + \beta + \gamma \leq 1$.

The proof of Theorem 2.3.3 is immediate from the above corollary.

2.4.4 Hardness Results

In this section we restate and prove the hardness result of Theorem 2.3.4:

Theorem 2.3.4. *For all arrival models, given the three objectives: operator's profit, offline side group (individual) fairness, and online side group (individual) fairness. No algorithm can achieve a competitive ratio of* (α, β, γ) *over the three objectives simultaneously such that* $\alpha + \beta + \gamma > 1$.

Proof. We prove it for group fairness in the KNOWNIID setting, since the KNOWNIID setting is a special case of the KNOWNAD setting, then this also proves the upper bound for the KNOWNAD setting.

Consider the graph $G = (U, V, E)$ which consists of three offline vertices and three online vertex types, i.e. $|U| = |V| = 3$. Each vertex in U (V) belongs to its own distinct group. The time horizon T is set to an arbitrarily large value. The arrival rate for each $v \in V$ is uniform and independent of time, i.e. KNOWNIID with $p_v = \frac{1}{3}$. Further, the bipartite graph is complete, i.e. each vertex of U is connected to all of the vertices of V with $p_e = 1$ for all $e \in E$. We also let $\Delta_u = 1$ for each $u \in U$, $n_v = \frac{T}{3}$ and $\Delta_v = 1$ for each $v \in V$. We represent the utilities on the edges of E with matrices where the (i, j) element gives the utility of the edge connecting vertex $u_i \in U$ and vertex $v_j \in V$. The utility matrices for the platform operator, offline, and online sides are following, respectively:

$$M_O = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, M_U = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, M_V = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

It can be seen that the utility assignments in the above example conflict between the three entities.

Let $\mathrm{OPT}_O, \mathrm{OPT}_U$, and OPT_V be the optimal values for the operator's profit, offline group fairness, and online group fairness, respectively. It is not difficult to see that $\mathrm{OPT}_O = 3$, $\mathrm{OPT}_U = 1$, and $\mathrm{OPT}_V = 1$. Now, denote by A, B, and C the edges with values of 1 for M_O, M_U, and M_V in the graph, respectively. Further, for a given online algorithm, let a_j, b_k, and c_ℓ be the expected number of probes received by edges $j \in A, k \in B$, and $\ell \in C$, respectively. Moreover, denote the algorithm's expected value over the operator's profit, expected fairness for offline vertices, and expected fairness for online vertices by $\mathrm{ALG}_O, \mathrm{ALG}_U$, and ALG_V, respectively. We can upper bound the sum of the competitive ratios as follows:

$$\begin{aligned}
\frac{\mathrm{ALG}_O}{\mathrm{OPT}_O} + \frac{\mathrm{ALG}_U}{\mathrm{OPT}_U} + \frac{ALG_V}{\mathrm{OPT}_V} &\leq \frac{\sum_{j \in A} a_j}{3} + \frac{\min_{k \in B} b_j}{1} + \frac{\min_{\ell \in C} c_j}{1} \\
&\leq \frac{\sum_{j \in A} a_i}{3} + \frac{\left(\sum_{k \in B} b_i\right)/3}{1} + \frac{\left(\sum_{\ell \in C} c_i\right)/3}{1} \\
&\leq \frac{\sum_{j \in A} a_i + \sum_{k \in B} b_i + \sum_{\ell \in C} c_i}{3} \leq \frac{3}{3} = 1.
\end{aligned}$$

The second inequality follows since the minimum value is upper bounded by the average. The last inequality follows since $\Delta_u = 1$ and therefore the expected number of probes any offline vertex receives cannot exceed 1 and we have $|U| = 3$ many vertices.

To prove the same result for individual fairness we use the same graph. We note that the arrival of vertices in V is KNOWNAD instead with the i^{th} vertex v_i having $p_{v_i,i} = 1$ and $p_{v_i,t} = 0, \forall t \neq i$. Then we follow an argument similar to the above. □

The following proves Theorem 2.3.5 therefore showing that there is indeed a conflict between achieving group and individual fairness even if we were to consider only one side of the graph.

Theorem 2.3.5. *Ignoring the operator's profit and focusing either on the offline side alone or the*

online side alone. With α_G and α_I denoting the group and individual fairness competitive ratios, respectively. No algorithm can achieve competitive ratios (α_G, α_I) over the group and individual fairness objectives of one side simultaneously such that $\alpha_G + \alpha_I > 1$.

Proof. Let us focus on the offline side, i.e. we consider α_G and α_I that are the competitive ratios for the group and individual fairness of the offline side.

Consider a graph which consists of two offline vertices and one online vertex, i.e. $|U| = 2$ and $|V| = 1$. Further, there is only one group. Let $p_e = 1, \forall e \in E$ and $\forall u \in U, \forall v \in V : \Delta_u = \Delta_v = 1$. U has two vertices u_1 and u_2 both connected to the same vertex $v \in V$. For edge (u_1, v), we let $w^U_{(u_1,v)} = 1$ and for edge (u_2, v), we let $w^U_{(u_2,v)} = L$ where L is an arbitrarily large number. Note that both of these weights are for the utility of the offline side. Finally, we only have one round so $T = 1$.

Let θ_1 and θ_2 be the expected number of probes edges (u_1, v) and (u_2, v) receive, respectively. Note that $\theta_1 = 1 - \theta_2$. It follows that the optimal offline group fairness objective is $\mathrm{OPT}^U_G = \max_{\theta_1,\theta_2}(\theta_1 + L\theta_2) = \max_{\theta_2}((1 - \theta_2) + L\theta_2) = L$. Further, the optimal offline individual fairness objective is $\mathrm{OPT}^U_I = \min\{\theta_1, L\theta_2\}$, it is not difficult to show that $\mathrm{OPT}^U_I = \frac{L}{L+1}$. Now consider the sum of competitive ratios, we have:

$$\begin{aligned}
\frac{\mathrm{ALG}^U_G}{\mathrm{OPT}^U_G} + \frac{\mathrm{ALG}^U_I}{\mathrm{OPT}^U_I} &= \frac{\theta_1 + L\theta_2}{L} + \frac{\min\{\theta_1, L\theta_2\}}{\frac{L}{L+1}} \\
&\leq \frac{\theta_1 + L\theta_2}{L} + \frac{\theta_1(L+1)}{L} \\
&= \frac{(L+2)\theta_1 + L\theta_2}{L} \\
&= (\theta_1 + \theta_2) + \frac{2\theta_1}{L} \\
&\leq 1 + \frac{2\theta_1}{L} \xrightarrow{L\to\infty} 1
\end{aligned}$$

this proves the result for the offline side of the graph.

To prove the result for the online side, we reverse the graph construction, i.e. having one vertex in U and two vertex types in V which arrive with equal probability. It now holds that $\mathrm{OPT}_I^V = \min\{\theta_1, L\theta_2\}$ and by setting T to an arbitrarily large value $\mathrm{OPT}_G^V = L$. Then we follow an identical argument to the above. □

2.5 Experiments

In this section, we verify the performance of our algorithm and our theoretical lower bounds for the KNOWNIID and group fairness setting using algorithm $\mathrm{TSGF}_{\text{KNOWNIID}}$ (Section 2.4.1). We note that none of the previous work consider our three-sided setting. We use rideshare as an application example of online bipartite matching [see also, e.g., 5, 21, 70, 85]. We expect similar results and performance to hold in other matching applications such as crowdsourcing.

Experimental Setup: As done in previous work, the drivers' side is the offline (static) side whereas the riders' side is the online side. We run our experiments over the widely used New York City (NYC) yellow cabs dataset [5, 70, 86, 87] which contains records of taxi trips in the NYC area from 2013. Each record contains a unique (anonymized) ID of the driver, the coordinates of start and end locations of the trip, distance of the trip, and additional metadata.

Similar to [5, 21], we bin the starting and ending latitudes and longitudes by dividing the latitudes from $40.4°$ to $40.95°$ and longitudes from $-73°$ to $-75°$ into equally spaced grids of step size 0.005. This enables us to define each driver and request type based on its starting and ending bins. We pick out the trips between 7pm and 8pm on January 31, 2013, which is a rush hour with 10,814 drivers and 35,109 trips. We set driver patience Δ_u to 3. Following [70], we uniformly sample rider patience Δ_v

from $\{1, 2\}$.

Since the dataset does not include demographic information, for each vertex we randomly sample the group membership [5]. Specifically, we randomly assign 70% of the riders and drivers to be advantaged and the rest to be disadvantaged. The value of p_e for $e = (u, v)$ depends on whether the vertices belong to the advantaged or disadvantaged group. Specifically, $p_e = 0.6$ if both vertices are advantaged, $p_e = 0.3$ if both are disadvantaged, and $p_e = 0.1$ for other cases.

In addition to this, a key component of our work is the use of driver and rider specific utilities. We follow the work of [71] to set the utilities. We adopt the Manhattan distance metric rather than the Euclidean distance metric since the former is a better proxy for length of taxi trips in New York City. We set the operator's utility to the rider's trip length $w_e^O = \text{tripLength}(v)$—a rough proxy for profit. In addition, the rider's utility over an edge $e = (u, v)$ is set to $w_e^V = -\text{dist}(u, v)$ where $\text{dist}(u, v)$ is the distance between the rider and the driver. The driver's utility is set to $w_e^U = \text{tripLength}(v) - \text{dist}(u, v)$. Whereas the trip length $\text{tripLength}(v)$ is available in the dataset, the distance between the rider and the driver $\text{dist}(u, v)$ is not. We therefore simulate the distance, by creating an equally spaced grid with step size 0.005 around the starting coordinates of the trip. This results in 81 possible coordinates in the vicinity of the starting coordinates of the trip. We then randomly choose one of these 81 coordinates to be the location of the driver when the trip was requested. Then $dist(u, v)$ is the distance between this coordinate to the start coordinate of the trip. This is a valid approximation since the platform would not assign drivers unreasonably far away to pickup a rider. Lastly, we scale the utilities by a constant to prevent them from being negative.

We run $\text{TSGF}_{\text{KnownIID}}$ at the scale of $|U| = 49$, $|V| = 172$ for 100 trials. During each trial, we randomly sample 49 drivers and 172 requests between 7 and 8pm, and run $\text{TSGF}_{\text{KnownIID}}$ 100 times to measure the expected competitive ratios of this trial. We then averaged the competitive ratios over

all trials, and the results are reported in figure 2.1. Code to reproduce our experiments is available in the blinded format[2]; we will release that code in deblinded form upon acceptance.

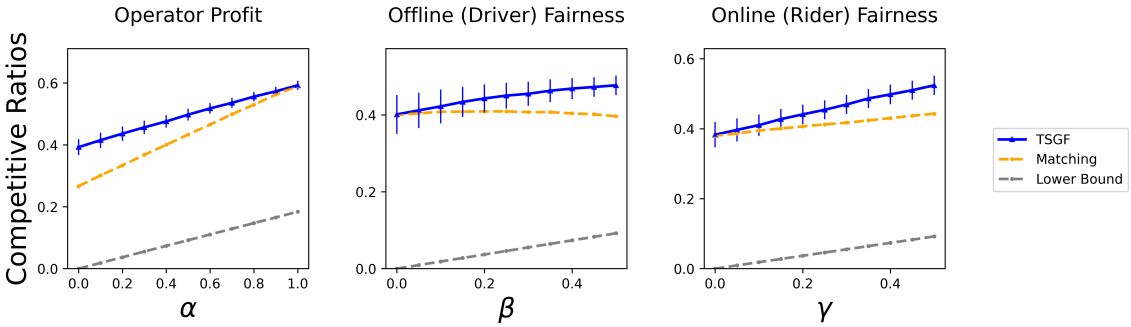


Figure 2.1: Competitive ratios for $\text{TSGF}_{\text{KNOWNIID}}$ over the operator's profit, offline (driver) fairness objective, and online (rider) fairness objective with different values of α, β, γ.

Performance of $\text{TSGF}_{\text{KNOWNIID}}$ with Varied Parameters: Figure 2.1 shows the performance of our algorithm over the three objectives: operator's profit, offline (driver) group fairness, and online (rider) group fairness. We note that "Matching" refers to the case where driver and rider utilities are set to 1 across all edges. The experiment is run with $\alpha = \{0, 0.1, 0.2, ..., 1\}$, and $\beta = \gamma = \frac{1-\alpha}{2}$. Higher competitive ratio indicates better performance. It is clear that the algorithm behaves as expected with all objectives being steadily above their theoretical lower bound. More importantly, we see that increasing the weight for an objective leads to better performance for that objective. I.e., a higher weight for β leads to better performance for the offline side fairness and similar observations follow in the case of α for the operator's objective and in the case of γ for the online-fairness. This also indicates the limitation in previous work which only considered fairness for one-side since their algorithms would not be able to improve the fairness for the other ignored side.

Furthermore, previous work [e.g., 5, 69, 70] only considered the matching size when optimizing the fairness objective for the offline (drivers) or online (riders) side. This is in contrast to our setting

[2]https://github.com/anonymousUser634534/TSGF

where we consider the matching quality. To see the effect of ignoring the matching quality and only considering the size, we run the same experiments with $w_e^U = w_e^V = 1, \forall e \in E$, i.e. the quality is ignored. The results are shown shown in the graph labelled "Matching" in figure 2.1, it is clear that ignoring the match quality leads to noticeably worse results.

Comparison to Heuristics: We also compare the performance of $\text{TSGF}_{\text{KNOWNIID}}$ against three other heuristics. In particular, we consider Greedy-O which is a greedy algorithm that upon the arrival of an online vertex (rider) v picks the edge $e \in E_v$ with maximum value of $p_e w_e^O$ until it either results in a match or the patience quota is reached. We also consider Greedy-R which is identical to Greedy-O except that it greedily picks the edge with maximum value of $p_e w_e^V$ instead, therefore maximizing the rider's utility in a greedy fashion. Moreover, we consider Greedy-D which is a greedy algorithm that upon the arrival of an online vertex v, first finds the group on the offline side with the lowest average utility so far, then it greedily picks an offline vertex (driver) $u \in E_v$ from this group (if possible) which has the maximum utility until it either results in a match or the patience limit is reached. We carried out 100 trials to compare the performance of $\text{TSGF}_{\text{KNOWNIID}}$ with the greedy algorithms, where each trial contains 49 randomly sampled drivers and 172 requests and is repeated 100 times. The aggregated results are displayed in table 2.1. We see that $\text{TSGF}_{\text{KNOWNIID}}$ outperforms the heuristics with the exception of a small under-performance in comparison to Greedy-D. However, using Greedy-D we cannot tune the weights (α, β, and γ) to balance the objectives as we can in the case of $\text{TSGF}_{\text{KNOWNIID}}$.

As mentioned before one of the major contributions of our work is that we consider the operator's profit and fairness for both sides *simultaneously* instead of fairness for only one side. To further see the effects of ignoring one side, we run $\text{TSGF}_{\text{KNOWNIID}}$ with one side ignored (see table 2.2). Table 2.2 shows the results of running $\text{TSGF}_{\text{KNOWNIID}}$ on the NYC dataset with the fairness on one side ignored,

	Profit	Driver Fairness	Rider Fairness
Greedy-O	0.431	0.549	0.503
$\text{TSGF}_{\textbf{KnownIID}}$ ($\alpha = 1$)	0.595	0.398	0.384
Greedy-D	0.371	0.609	0.563
$\text{TSGF}_{\textbf{KnownIID}}$ ($\beta = 1$)	0.517	0.571	0.44
Greedy-R	0.316	0.504	0.513
$\text{TSGF}_{\textbf{KnownIID}}$ ($\gamma = 1$)	0.252	0.353	0.574

Table 2.1: Competitive ratios of $\text{TSGF}_{\text{KnownIID}}$ with Greedy heuristics on the NYC dataset at $|U| = 49$, $|V| = 172$. Higher competitive ratio indicates better performance.

i.e. its optimization weight set to 0: (**Top Row**) Offline (Driver) fairness ignored ($\alpha = \gamma = 0.5, \beta = 0$) and (**Bottom Row**) Online (Rider) fairness ignored ($\alpha = 0.5, \beta = 0.5, \gamma = 0$). It is clear that the fairness objective for the ignored side is indeed lower in comparison to what can be achieved in figure 2.1. More precisely, we can see that the Offline (Driver) and Online (Rider) fairness can be simultaneously improved to around 0.5 by setting $\alpha = 0.5, \beta = \gamma = 0.25$ (figure 2.1) whereas their values when their optimization weight is set to zero is 0.387 and 0.41 , respectively (see table 2.2).

	Profit	Driver Fairness	Rider Fairness
$\alpha = \gamma = 0.5, \beta = 0$	0.43	0.387	0.509
$\alpha = \beta = 0.5, \gamma = 0$	0.564	0.498	0.41

Table 2.2: Results of running $\text{TSGF}_{\text{KnownIID}}$ with one-sided fairness.

Chapter 3: Proportionally Fair Matching

This chapter is largely based on our paper [37].

3.1 Introduction

Graph matching, in particular the special case of bipartite matching, is a classical computational problem with many applications such as ad allocation [26, 27], crowdsourcing [18, 19, 28, 29], job hiring [30], organ exchange [31, 32, 33, 34], and ride sharing [29, 35, 36]. Matching is also a fundamental subroutine in several domains including computer vision [88], text similarity estimation [89] in natural language processing, machine learning algorithms [90, 91, 92, 93] and computational biology [94] among others.

In many applications, algorithms are employed to make decisions that could significantly impact the lives of individuals. In such settings, we ought to ensure fairness and equity in the decision-making process. However, classical algorithms typically do not consider such socially motivated objectives and constraints such as fairness or diversity. For instance, a ride-sharing platform may provide better quality assignments (e.g., with shorter wait times, newer vehicles, or better pricing) to riders from certain demographic groups while providing lower quality assignments to others as studied by [25]. Similarly, cognitive biases from workers can negatively impact the result of crowdsourced data, which may potentially be mitigated by assigning a diverse set of workers to a wide range of different *task*

types.

There are many ways to formalize and incorporate some notion of *fairness* into the computed solutions of matching problems (see, e.g., Bandyapadhyay et al. [16], Esmaeili et al. [25]). Following the work of Bandyapadhyay et al. [16], we consider a notion of *proportional fairness*. This notion is closely related to the notion of *disparate impact* [95] and has been explored in various fundamental problems, including matroid optimization [9], clustering [8], online matching [96], set packing [48], spectral clustering [97], and others. In the present work, we consider a formulation of proportional fairness known as (α, β)-BALANCEDMATCHING: given an edge-colored graph (where edge colors may denote membership in specific groups), the objective is to find a maximum weight matching ensuring that the proportion of edges of any color among all matched edges lies between α and β where $0 \leq \alpha \leq \beta \leq 1$. We highlight that (α, β)-BALANCEDMATCHING adheres to two key criteria: *restricted dominance*, which limits the fraction of edges selected from any given group to at most β, and *minority protection*, which ensures that the fraction of edges from any given group is at least α.

This notion of fairness can be applied to the applications discussed above, such as in ride-sharing, job hiring, and crowd sourcing. By assigning colors to edges (rather than only to vertices), the *color* is able to capture information about both sides of the match. For instance, in ridesharing, the edge color may encode information about both the driver (e.g., vehicle type, rating) and rider (e.g., race or gender), the distance between the driver and rider, and even pricing information (e.g., whether dynamic or surge pricing is applied). As a concrete example, we may choose to encode information about a rider's race or economic status as well as whether an assignment incurs increased fares due to dynamic pricing; then, an (α, β)-BALANCEDMATCHING should ensure that no racial or economic group receives increased fares disproportionately often compared to other riders. In crowdsourcing, we may use edge color to encode both the task type and demographic information about the worker to ensure that a diverse set of

workers are assigned to each type of task (for cognitive or human intelligence tasks, this may increase the diversity of perspectives among responses, and thus the overall quality of the final aggregate data).

As a final example application, consider online advertising: an edge's color may encode information about the type of advertisement (or, in political advertising, the political party sponsoring the ad) and the user's personal information. Here, an (α, β)-BALANCEDMATCHING would limit the extent of *targeted* advertising based on protected demographic traits, which may violate (or appear to violate) a user's expectations and rights with regards to their privacy and use of their data; the social consequences of such indiscriminate highly targeted advertising were seen in the real world with the Cambridge Analytica data scandal around the 2016 US presidential election, which saw Facebook's CEO Mark Zuckerberg testify before congress [98, 99].

Unfortunately, although this notion of fairness seems quite powerful, it may in fact be too strong in the basic formulation: Bandyapadhyay et al. [16] demonstrated that (α, β)-BALANCEDMATCHING is NP-hard to approximate, even when G is an *unweighted path graph*.

To address this fundamental hardness, we consider a slightly relaxed probabilistic notion of proportional fairness. Roughly speaking, we allow for small violations of the α and β bounds while still guaranteeing a constant factor approximation on the overall objective, i.e., the weight of the matching. Particularly, we ask the following question: *does there exists an approximation algorithm with constant factor violations in fairness while guaranteeing a good approximation on the objective?*

We answer this question by designing a $\frac{1}{2}$-approximate algorithm for bipartite graphs and whose matching is *probably*[1] *almost fair*. That is, there exists a small constant $\delta > 0$ for which the matching is $(\alpha(1-\delta), \beta(1+\delta))$-balanced with high probability. A formal definition of this notion is given in Section 3.3. Thus, while the (α, β)-BALANCEDMATCHING formulation of Bandyapadhyay et al.

[1] i.e., with probability close to 1

[16] remains hard on bipartite graphs, we can achieve our notion of *probably almost fair*. We note that for the special case when $\alpha = 0$ and $\beta < 1$, we can attain a constant approximation ratio without violating the fairness constraint at all. Combined with the hardness result of Bandyapadhyay et al. [16], this implies that the one-sided fairness case is strictly easier than the two-sided case.

3.2 Related Work

A significant body of research in fair matching explored various notions of fairness. Among these, several notions of *group fairness*, such as socially-fair matching [16], focus on Rawlsian (maxmin) fairness, aiming to maximize the utility for the worst-off group [25]. On the other hand, proportional fairness, as explored in Chierichetti et al. [9], Bandyapadhyay et al. [16], ensures a proportional representation of edges from each group, while leximin fairness [100] is also considered. Most of these studies involve assigning group memberships to vertices in the graph. However, works such as [9, 16] investigate fairness in edge-colored graphs, which aligns with our area of study. We note that [9, 16] focus only on the unweighted setting where the objective is the cardinality/size of the output. That being said, our works are incomparable. While ours applies to edge weighted bipartite graphs, [9] consider when the constraint system is described by the intersection of two-matroids (this generalizes unweighted bipartite matching). They focus on the case of two colors, and present a polynomial-time algorithm achieving a $3/2$-approximation. Bandyapadhyay et al. [16] consider when input is a general graph and the number of colors may be greater than two. They show that approximating the problem for an arbitrary number of colors is NP-hard and moreover that approximating the optimal solution requires time exponential in ℓ unless the Exponential Time Hypothesis (ETH) [101] is false. Their main algorithm therefore runs in time exponential in ℓ, with an approximation guarantee of $1/(4\ell)$.

Note that even for the easier case when $\alpha = 0$, their algorithm may still violate the fairness constraint imposed by β by a factor up to $1+1/\ell$. Their hardness result motivates our study of the slightly relaxed probabilistic fairness. This relaxation allows us to improve on the prior work by achieving, in polynomial time, a *constant factor* (i.e., one half) approximation in the matching weight while ensuring, with high probability, only a small violation in the fairness constraints.

3.3 Preliminaries and Problem Formulation

We denote $[k] = \{1, \ldots, k\}$ for any positive integer k. Consider an undirected bipartite graph $G = (U, V, E)$ with the set of edges E forming a partition $\dot{\cup}_{c\in[\ell]} E_c$, where $\dot{\cup}$ denotes a disjoint union over the sets E_c. Each color $c \in [\ell]$ is associated with the set of edges E_c, representing edges of color c. For any color class c, we define $\psi_c(e)$ such that $\psi_c(e) = 1$ if $e \in E_c$, and $\psi_c(e) = 0$ otherwise. A *matching* $M \subseteq E$ in G is defined as a subset of edges where no two edges in M share a common vertex. For a vertex v of G, let $N(v)$ and $\delta(v)$ denote the neighbors and incident edges of v, i.e., $N(v) := \{u : (u, v) \in E\}$ and $\delta(v) := \{(u, v) : (u, v) \in E\}$. For any $\alpha, \beta \in [0, 1]$ and $\alpha \leq \beta$, we define a matching $M \subseteq E$ as (α, β)-BALANCED if, for each color $c \in [\ell]$, the proportion of edges in M belonging to color c lies between α and β. In other words, M is (α, β)-BALANCED if it contains at least α and at most β fraction of edges from every color. The goal of the *proportional fair matching problem* (PFM) is to find a (α, β)-BALANCEDMATCHING of G with maximum weight, and we denote the weight of such a matching by OPT. Since [16] proved that it is NP-Hard to verify the existence of a (α, β)-BALANCEDMATCHING in the PFM problem even on unweighted path graphs, we focus on deriving approximation ratios which hold against OPT.

3.3.1 Our Contributions and Techniques

We design Algorithm 4, an efficient randomized algorithm for weighted matching on bipartite graphs with fairness constraints specified by $0 < \alpha \leq \beta < 1$. By allowing the α (respectively, β) fairness constraint to be violated up to a multiplicative factor of $1 - \delta$ (respectively, $1 + \delta$), we show that Algorithm 4 attains an asymptotic approximation ratio against OPT. (We write the exact asymptotic guarantee of Algorithm 4 in Theorem 3.5.1). We argue that if the input is well-behaved, then we can take $\delta \approx 0$, and so we barely violate the fairness constraints, while still attaining a constant approximation ratio. We next analyze Algorithm 4 when $\alpha = 0$ and $\beta < 1$. By allowing for a $\frac{1-\varepsilon}{2}$-approximation ratio, where $0 < \varepsilon < 1$ is a parameter of Algorithm 4 which can be chosen arbitrarily small, we show how to ensure that the β fairness constraints are satisfied exactly. This allows us to get a true approximation ratio which is arbitrarily close to $1/2$. We additionally note that our algorithms can be extended to the slightly more general setting where we have different proportionality requirements for each color; e.g., where we have values α_c, β_c (with $0 \leq \alpha \leq \beta \leq 1$) for each color $c \in [\ell]$, and we insist that $\alpha_c|\mathcal{M}| \leq |\mathcal{M}_c| \leq \beta_c|\mathcal{M}|$ for each c. This allows us to capture settings where we wish to represent different groups in different proportions, for example so that each group's representation in the matching is roughly proportional to its representation in the original graph.

We take a linear programming (LP) based approach to designing Algorithm 4. Below we state LP-FAIR, which extends the standard matching polytope to include the proportionality constraints

described by $0 < \alpha \leq \beta < 1$:

$$\max \sum_{e \in E} w_e x_e \tag{LP-FAIR}$$

$$\text{s.t.} \sum_{e \in \delta(v)} x_e \leq 1, \, \forall v \in U \cup V \tag{3.1a}$$

$$\alpha \sum_{e \in E} x_e \leq \sum_{e \in E_c} x_e \leq \beta \sum_{e \in E} x_e, \qquad \forall c \in [\ell] \tag{3.1b}$$

$$x_e \geq 0, \, \forall e \in E \tag{3.1c}$$

Lemma 3.3.1. LP-FAIR *relaxes* OPT. *That is,* $\text{OPT} \leq \sum_{e \in E} w_e x_e$, *where* $\boldsymbol{x} = (x_e)_{e \in E}$ *is an optimal solution to* LP-FAIR.

Proof. We show this by □

As a first attempt to using LP-FAIR, observe that since G is bipartite, if $\boldsymbol{x} = (x_e)_{e \in E}$ is an optimal solution to the LP, then we can write $\boldsymbol{x}$ as a convex combination of integral matchings. By randomly sampling such a matching according to the coefficients of the convex combination, this yields a randomized algorithm with expected value $\sum_{e \in E} w_e x_e \geq \text{OPT}$. Unfortunately, while this preserves the fairness constraints in expectation due to (3.1b), there is no guarantee that any *particular* matching we output will satisfy these constraints. In order for an algorithm to succeed with constant probability, an ideal approach would be to match each edge e with probability x_e while ensuring that the size of a matching of a color class c is concentrated about $\sum_{e \in E_c} x_e$. For instance, if we had *negative correlation* amongst the matching statuses of the edges of E_c, then this would be sufficient (see [102]). The GKPS dependent rounding scheme of [103] provides such a guarantee for *certain*

edge subsets of G. However, since the edges of E_c are adversarially chosen, it is easy to construct an example where the GKPS rounding scheme induces positive correlation amongst the matched statuses.

Due to the limitations of these approaches, we need to round our LP in a different way. Let us suppose that we round $\boldsymbol{x}$ into a random matching $\mathcal{M}$, where we denote $\mathcal{M}_c := \mathcal{M} \cap E_c$ for a fixed color class $c \in [\ell]$. Roughly speaking, our goal is to ensure that the following properties simultaneously hold, where $\delta > 0$ is a fixed parameter to be specified in Theorem 3.5.3:

1. For each $e \in E$, $\Pr[e \in \mathcal{M}] = x_e/2$.

2. With constant probability,

$$\frac{|\mathcal{M}_c|}{|\mathcal{M}|} \in \left[(1-\delta)\frac{\mathbb{E}[|\mathcal{M}_c|]}{\mathbb{E}[|\mathcal{M}|]}, (1+\delta)\frac{\mathbb{E}[|\mathcal{M}_c|]}{\mathbb{E}[|\mathcal{M}|]}\right].$$

Property 1 and Lemma 3.3.1 then imply that

$$\mathbb{E}[\sum_{e\in\mathcal{M}} w_e] = \sum_{e\in E} w_e x_e/2 \geq \mathrm{OPT}\,/2.$$

Moreover, since $\mathbb{E}[|\mathcal{M}_c|] = \sum_{e\in E_c} x_e/2$ and $\mathbb{E}[|\mathcal{M}|] = \sum_{e\in E} x_e/2$, we can combine Property 2. with (3.1b) to conclude that $(1-\delta)\alpha|\mathcal{M}| \leq |\mathcal{M}_c| \leq (1+\delta)\beta|\mathcal{M}|$.

Our approach uses a randomized rounding tool known as a *contention resolution scheme (CRS)*. CRS's were originally introduced by [104] to solve certain constrained sub-modular optimization problems. Motivated by the applications to prophet inequalities, they were later adapted to the online setting by [105], where they are referred to as *online* contention resolution schemes (OCRS's). Since then, they have found many other applications, and have become a fundamental tool in stochastic optimiza-

tion. We continue this line of research and show that this tool can be useful even for our problem which is offline and has no inherit stochasticity.

Our algorithm first has every $v \in V$ independently draw a random vertex $F_v \in N(v)$, where

$$\Pr[F_v = u] = x_{u,v} \tag{3.2}$$

for each $u \in N(v)$. (For convenience, we define F_v to be a *null element* $\perp$ if no draw is made, where $\mathbb{P}[F_v = \perp] = 1 - \sum_{u \in N(v)} x_{u,v}$). If $F_v = u$, we say that v *proposes* to u and refer to F_v as a *proposal* for u. At this point, we know that each vertex $v \in V$ has made at most one proposal. However, a fixed vertex $u \in U$ may have received multiple proposals, and so we must resolve which proposal each vertex u should take. This is precisely the purpose of the OCRS, and we use the explicit scheme of [106]. For a fixed $u \in U$, the input required of the OCRS is the edge values $(x_{u,v})_{v \in N(u)}$, together with the proposals $(F_v)_{v \in N(u)}$. Moreover, in the contention resolution framework, $(F_v)_{v \in N(u)}$ must be independent. Under these conditions, the OCRS outputs at most one edge (u, v) with $F_v = u$, while guaranteeing that for *all* $v \in N(u)$

$$\Pr[(u, v) \text{ is output} \mid F_v = u] = 1/2. \tag{3.3}$$

Due to guarantee of (3.3), the OCRS is said to be $1/2$-*selectable* in the literature. By concurrently running a separate execution of this OCRS for each $u \in U$, the matching we output satisfies Property 1.

We still need the explain why our algorithm satisfies Property 2. It turns out our algorithm has a very simple description. First, we order the vertices of V arbitrarily, say $v_1, \ldots, v_n$. Then, when v_t

is processed and proposes to u (i.e., $F_{v_t} = u$), we draw an *independent* random bit A_{u,v_t} We match (u, v_t) provided u was not previously matched and $A_{u,v_t} = 1$. We note that in regards to verifying Property 2, the specific Bernouli parameter of A_{u,v_t} is not important. We simply require that $(A_e)_{e\in E}$ are drawn independently.

If we let $\mathcal{M}$ be the output of the algorithm, and $\mathcal{M}_c = E_c \cap \mathcal{M}$, then we can analyze the Doob martingale of $|\mathcal{M}_c|$. Here the Doob martingale is defined by revealing the partial matching after the vertices $v_1, \ldots, v_t$ are processed. Due to the simplicity of our algorithm, we can bound the one-step changes in the Doob martingale in a very precise way that is related to $\sum_{e\in E_c} x_e$. Note that the standard Azuma–Hoeffding martingale concentration inequality does not allow for good bounds due to constant worst-case one-step changes (as we explain in Section 3.6.1 and an example in Example 3.9.1) leading to $\Theta(n)$ over the entire process from $t=1$ to $t = n$. We show that by controlling the *variance* of this martingale, we can get stronger bounds via Freedman's inequality. More precisely, we can tighten the concentration on the random variable $\mathcal{M}_c$ by using one-step variance which can be much smaller than the worse-case one-step changes since the events involving the bad cases occur with small probability. The total variance can be precisely computed and is upper bounded by $10\sum_{e\in E_c} x_e$ which is much smaller than $\Theta(n)$. The concentration helps us achieve *almost* fairness, i.e., $|\mathcal{M}_c|$ violates the fairness constraints on each side by at most a δ-multiplicative factor for some small $\delta > 0$. Finally, when we only have β-sided fairness constraints i.e., $\alpha = 0, \beta < 1$, we apply an LP *perturbation trick* to recover a matching that satisfies the constraints *exactly*, meaning that without any violations with only a $0 < \epsilon < 1$ multiplicative loss in the objective. However, it is important to note that our algorithm still has a probabilistic guarantee: with a certain probability, the matching returned will *not* satisfy the fairness constraints. At the expense of additional runtime, we can improve the success probability to be arbitrarily close to 1 by repeatedly running our algorithm and taking the best solution which is

feasible.

3.4 Our Algorithm

We begin by formally describing our algorithm to find a matching for a bipartite graph $G = (U, V, E)$ with color classes $\cup_{c\in[\ell]} E_c$ and $\alpha \leq \beta \leq 1$. Note that our algorithm takes in a parameter $0 < \varepsilon < 1$ which is only needed for the special case when $\alpha = 0$.

For $\delta > 0$, a random matching $M \subseteq E$ is said to be δ-δ-PROBABLYALMOSTFAIR with respect to $0 \leq \alpha \leq \beta \leq 1$ if for any color class c, [2]

$$\Pr\left[(1-\delta)\alpha \leq \frac{|M_c|}{|M|} \leq (1+\delta)\beta\right] \geq 1 - f_c(\delta, G). \tag{3.4}$$

where $f_c(\delta, G)$ is a term that depends on the tunable parameter δ and the input instance. This definition allows us to violate the constraints by a small δ-multiplicative factor which can be adjusted to attain a reasonable probability of success.

Algorithm 4 OCRS Rounding Algorithm

Input: $G = (U, V, E)$ with color classes $(E_c)_{c\in[\ell]}$, $0 \leq \alpha \leq \beta \leq 1$, and $0 < \varepsilon < 1$.
Output: subset of edges forming a matching $\mathcal{M}$

$\mathcal{M} \leftarrow \varnothing$
If $\alpha > 0$, compute an optimal solution $\boldsymbol{x} = (x_e)_{e\in E}$ to LP-FAIR with $0 < \alpha \leq \beta \leq 1$. Otherwise, compute an optimal solution $\boldsymbol{x}$ to LP-FAIR with β in (3.1b) replaced by $\tilde{\beta} = (1-\varepsilon)\beta$.
Order the vertices of V as $v_1, \ldots, v_n$ arbitrarily.
Draw $(F_{v_t})_{t=1}^n$ as described in (3.2)
for $t = 1, \ldots, n$ with $F_{v_t} \neq \perp$ **do**
    Set $u := F_{v_t}$, and $a_{u,v_t} := \frac{1/2}{1-(1/2)\sum_{i<t} x_{u,v_i}}$.
    Draw $A_{u,v_t} \sim \mathrm{Ber}(a_{u,v_t})$ independently.
    If $A_{u,v_t} = 1$ and u is currently unmatched, add (u, v_t) to $\mathcal{M}$
end for

return $\mathcal{M}$.

[2] Taking $\delta = 1$ satisfies the Equation (3.4) trivially

3.5 Main Results

Our main results are as follows:

Theorem 3.5.1. *Algorithm 4 returns a matching $\mathcal{M}$ that has an expected weight of at least $\frac{1}{2}\,\mathrm{OPT}$ and is δ-δ-*PROBABLYALMOSTFAIR *with $f_c(\delta, G) = 4\exp\left(\frac{-\delta^2 \sum_{e\in E_c} x_e}{28}\right)$.*

Remark 3.5.2. The violations dependent on δ in the fairness constraints are inversely exponential to the failure probability $f_c(\delta, G)$. This implies that achieving arbitrarily small violations $\delta \approx 0$ with a high success probability requires stronger assumptions on the term $\sum_{e\in E_c} x_e$ i.e., for input instances with $\sum_{e\in E_c} x_e = \omega(1)$ we can ensure high success probability with $\delta \approx 0$.

We next state our improved guarantee for the special case when $\alpha = 0$. Recall that in this setting, we can ensure that we do *not* violate the β-fairness constraint.

Theorem 3.5.3. *Given any $0 < \varepsilon < 1$, Algorithm 4 returns a matching that satisfies the β-fairness constraints with probability at least $1 - f(\epsilon, G)$ where $f(\epsilon, G) = 2\exp\left(-\epsilon^2\beta\frac{\sum_{e\in E} x_e}{28}\right)$, and has an expected weight of at least $\frac{1}{2}(1-\epsilon)\,\mathrm{OPT}$.*

In Theorem 3.5.3, the values $(x_e)_{e\in E}$ represent the optimal fractional solution to LP-FAIR with $\tilde{\beta} = \beta(1-\epsilon)$. Additionally, $f(\epsilon, G)$ denotes the probability that the algorithm fails to produce a matching that satisfies the β-fairness constraints *exactly* i.e., with no violations.

Remark 3.5.4. The probability that our algorithm fails decreases exponentially as a function of ϵ and $\beta\sum_{e\in E} x_e$. Therefore, to attain a failure probability $f(\epsilon, G) \approx 0$ and the approximation factor close to $1/2$, we require that $\beta\sum_{e\in E} x_e$ to be sufficiently large, say $\beta\sum_{e\in E} x_e \geq C$ for some constant C, say $C \geq 100$.

Remark 3.5.5. For the instances where $\beta \sum_{e \in E} x_e \leq C$, we address these cases separately using a brute force algorithm as described in Section 3.7.1. This involves enumerating all feasible matchings that satisfy the fairness with respect to β and having size at most $\frac{U^2}{L^2} \frac{C}{\beta(1-\epsilon)}$ where $U = \max_{e \in E} w_e$ and $L = \min_{e \in E} w_e$.

For all non-degenerate instances, $\frac{U}{L}$ is bounded by a constant. Hence, our algorithm runs in polynomial time if the term $\frac{U^2}{L^2} \frac{C}{(1-\epsilon)\beta}$ remains constant, given that $\frac{U}{L}$is bounded. However, in the worst-case, where $\beta = \frac{1}{\ell}$, the running time will be exponential in ℓ.

3.6 Algorithm Analysis

Suppose that we are given an instance of proportional fair matching. We let $\boldsymbol{x} = (x_e)_{e \in E}$ denote an optimal solution to LP-FAIR used by Algorithm 4. Note that if $\alpha = 0$, then $\boldsymbol{x}$ is computed by replacing β in (3.1b) with $\tilde{\beta} = (1 - \varepsilon)\beta$.

Recall that the vertices of V are processed in order $v_1, \ldots, v_n$. Let us say that $u \in U$ is *safe* at time t, provided u is *not* matched to any of $v_1, \ldots, v_{t-1}$. Observe that (u, v_t) is matched if and only if u is safe at time t, $F_{v_t} = u$ and $A_{u,v_t} = 1$. Let Z_{u,v_t} be an indicator random variable for this event. For convenience, we define $\tilde{F}_{v_t} := u$ provided $F_{v_t} = u$ and $A_{u,v_t} = 1$. Otherwise, $\tilde{F}_{v_t} = \perp$. Observe then that

$$\Pr[Z_{u,v_t} = 1] = \Pr[\tilde{F}_{v_t} = u, u \text{ is safe at } t] \tag{3.5}$$

Let $Z(v_t)$ be an indicator random variable that an edge of E_c is matched to vertex v_t where $Z(v_t) =$

$\sum_{u \in N(v_t)} \psi_c(u, v_t) Z_{u,v_t}$. Recall that $\mathcal{M}$ is the matching returned by our algorithm, therefore,

$$|\mathcal{M}_c| = \sum_{t \geq 1} \sum_{(u,v_t) \in \delta(v_t)} \psi_c(u, v_t) Z_{u,v_t}$$

Now, the instantiations of $(\tilde{F}_{v_i})_{i=1}^{t}$ are sufficient to determine which edges (if any) are matched to $v_1, \ldots, v_t$. Thus, we shall work with $\mathcal{H}_t$, the sigma-algebra generated by $(\tilde{F}_{v_i})_{i=1}^{t}$, which intuitively corresponds to the history of the algorithm's execution after $v_1, \ldots, v_t$ are processed. We define $\mathcal{H}_0$ to be the trivial sigma-algebra, where we do not condition on anything.

Fix a color class c, and let $M_t := \mathbb{E}[|\mathcal{M}_c| \mid \mathcal{H}_t]$ denote the expected number of edges from color class c in our matching, conditional on $\mathcal{H}_t$. Observe that $M_0 = \mathbb{E}[|\mathcal{M}_c|]$, and M_n is the actual number of edges chosen from the color class c at the end of our algorithm. Since we expose strictly more information in each round, i.e., formally, $\mathcal{H}_{t-1} \subseteq \mathcal{H}_t$ for each $t \geq 1$, the sequence $(M_t)_{t=0}^{n}$ is a martingale with respect to $(\mathcal{H}_t)_{t=0}^{n}$. (This is often referred to as a Doob martingale).

We aim to understand the size of the matching produced by our algorithm as each vertex v_t is processed. More specifically, when v_t is processed, our algorithm adds the edge (u, v_t) to the current matching if u is safe and $\tilde{F}_{v_t} = u$. Observe that M_t is a function of the current matching between U and $\{v_1, v_2, \ldots, v_t\}$. Moreover, for any $1 \leq t \leq n$,

$$M_t = \mathbb{E}\Big[\sum_{k \geq 1} Z(v_k) \mid \mathcal{H}_t\Big]. \tag{3.6}$$

Our goal is to control the one step changes of our martingale, which will imply that $M_n = |\mathcal{M}_c|$ is concentrated about $M_0 = \mathbb{E}[|\mathcal{M}_c|]$.

3.6.1 Warmup: Azuma-Hoeffding inequality

To apply concentration bounds like the Azuma–Hoeffding inequality, the martingale must be well-behaved i.e., at any step $1 \leq t \leq n$, the martingale cannot change dramatically. Thus, we need to determine the quantities c_t such that if $\Delta M_t := M_t - M_{t-1}$ is the one-step change at step t, then $|\Delta M_t| \leq c_t$.

Lemma 3.6.1. $|\Delta M_t| = |M_t - M_{t-1}| \leq 2$ *for all* $1 \leq t \leq n$.

Proof. By Lemma 3.6.6 we know that the one-step change $|\Delta M_t|$ is given by

$$\begin{aligned} |M_t - M_{t-1}| &= \max_{q \in N(v_t) \cup \{\perp\}} |\mathbb{E}[M_n \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = q] - \mathbb{E}[M_n \mid \mathcal{H}_{t-1}]| \\ &= \max \Big\{ \max_{u \in N(v_t)} |\mathbb{E}[M_n \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = u] - \mathbb{E}[M_n \mid \mathcal{H}_{t-1}]|, \\ &\qquad\qquad |\mathbb{E}[M_n \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = \perp] - \mathbb{E}[M_n \mid \mathcal{H}_{t-1}]| \Big\} \\ &\leq 2 \end{aligned}$$

The last inequality follows from Lemma 3.8.1 and Lemma 3.8.2. □

Let consider the one-step change in ΔM_t when $\tilde{F}_{v_t}$ is revealed. Since we can match at most one edge at each time step, we may miss the opportunity to match at most *two* edges of a given color. Therefore, in the worst case, we have that $\sum_t |\Delta M_t| = \Theta(n)$. For example, consider the star graph, where a central vertex is connected to all other vertices. In each time step, if we reveal $\tilde{F}_{v_t}$, then the change in the maximum matching will be at most 1, as we can only match or unmatch one edge (we defer the full details of the example to Example 3.9.1). To remedy this situation, we use a "second order" inequality for analyzing a martingale called Freedman's inequality:

Theorem 3.6.2 ([107]). *Suppose* $(M_t)_{t\in[n]}$ *is a martingale such that* $|\Delta M_t| \leq \Lambda$ *for any* $1 \leq t \leq n$, *and* $\sum_{t\in[n]} \mathrm{Var}[\Delta M_t \mid \mathcal{H}_{t-1}] \leq \nu$. *Then, for all* $\lambda > 0$,

$$\Pr\left[|M_n - M_0| \leq \lambda\right] \leq 2\exp\left(\frac{-\lambda^2}{2(\nu + \lambda\Lambda)}\right) \tag{3.7}$$

Remark 3.6.3. The variance term $\mathrm{Var}(\Delta M_t \mid \mathcal{H}_{t-1})$ reduces to $\mathrm{Var}[\Delta M_t \mid \mathcal{H}_{t-1}] = \mathbb{E}[|\Delta M_t|^2 \mid \mathcal{H}_{t-1}]$ since $\mathrm{Var}[\Delta M_t \mid \mathcal{H}_{t-1}] := \mathbb{E}[|\Delta M_t|^2 \mid \mathcal{H}_{t-1}] - \mathbb{E}[\Delta M_t \mid \mathcal{H}_{t-1}]^2$ and $\mathbb{E}[\Delta M_t \mid \mathcal{H}_{t-1}] = 0$ due to the martingale property.

Freedman's inequality still depends on worst case one-step change Λ i.e., an upper bound on $|\Delta M_t|$. However, its dependence on this parameter is less severe compared to the Azuma–Hoeffding inequality. The key difference is that Freedman's inequality depends on $\mathbb{E}[|\Delta M_t|^2 \mid \mathcal{H}_{t-1}]$, and so we get to *average* over the randomness in $\tilde{F}_{v_t}$. This is much smaller than than the pessimistic bound of Lemma 3.6.1, as even conditional on $\mathcal{H}_{t-1}$, the probability that a worst-case change in $|\Delta M_t|$ occurs is small.

3.6.2 Concentration via Freedman's inequality

Freedman's inequality allows us to use the expected one-step changes instead of the worst case changes as shown in the Observation 3.6.1 below.

Observation 3.6.1. $\mathrm{Var}(\Delta M_t \mid \mathcal{H}_{t-1}) \leq 2\mathbb{E}[|\Delta M_t| \mid \mathcal{H}_{t-1}]$

Proof. The variance of the one-step changes $\Delta M_t := M_t - M_{t-1}$, given $\mathcal{H}_{t-1}$, simplifies to $\mathrm{Var}(\Delta M_t \mid \mathcal{H}_{t-1}) = \mathbb{E}\left[|\Delta M_t|^2 \mid \mathcal{H}_{t-1}\right]$, because $\mathbb{E}[\Delta M_t \mid \mathcal{H}_{t-1}] = 0$ due to the martingale property. Recall that

$M_n = |\mathcal{M}_c|$. By the definition of M_t and given that $\mathcal{H}_t = (\mathcal{F}_{v_i})_{i=q}^{t}$, we have:

$$\begin{aligned}
\mathrm{Var}(\Delta M_t \mid \mathcal{H}_{t-1}) &= \sum_{u \in N(v_t) \cup \{\perp\}} \Pr[\tilde{F}_{v_t} = q] \left| \mathbb{E}\left[M_n \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = q\right] - \mathbb{E}\left[M_n \mid \mathcal{H}_{t-1}\right] \right|^2 \\
&\leq \sum_{u \in N(v_t) \cup \{\perp\}} 2 \Pr[\tilde{F}_{v_t} = q] \left| \mathbb{E}\left[M_n \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = q\right] - \mathbb{E}\left[M_n \mid \mathcal{H}_{t-1}\right] \right| \\
&= 2\mathbb{E}\left[|\Delta M_t| \mid \mathcal{H}_{t-1}\right]
\end{aligned}$$

The inequality above holds because the worst case change in the expected matching belonging to color class c is at most 2 as given by Lemma 3.6.1. □

We will prove two important facts (Lemma 3.6.4 and Lemma 3.6.5) about the one-step changes that will be used in the rest of the analysis.

Lemma 3.6.4. *For any step $k > t$, $w \in N(v_t)$ and $w \neq u$, if $\tilde{F}_{v_t} = u$, i.e., v_t proposes to u and $A_{u,v_t} = 1$, then*

$$\Pr[w \textit{ is safe at } k \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = u] - \Pr[w \textit{ is safe at } k \mid \mathcal{H}_{t-1}] \leq x_{w,v_t}.$$

Proof. Any vertex $w \in U$ is safe at step k if $\tilde{F}_{v_r} \neq w$ for any $1 \leq r \leq k$. The probability that

$w \in N(v_t)$ is safe conditional on $\mathcal{H}_{t-1}$ can be either 0 or 1. Therefore, we can conclude that

$$\begin{aligned}
&\Pr\left[w \text{ is safe at } k \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = u\right] - \Pr\left[w \text{ is safe at } k \mid \mathcal{H}_{t-1}\right] \\
&\qquad \leq \Pr\left[\cap_{t \leq r \leq k} \tilde{F}_{v_r} \neq w \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = u\right] - \Pr\left[\cap_{t \leq r \leq k} \tilde{F}_{v_r} \neq w \mid \mathcal{H}_{t-1}\right] \\
&\qquad = \prod_{t < r \leq k} \Pr\left[\tilde{F}_{v_r} \neq w \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = u\right] - \prod_{t \leq r \leq k} \left[\tilde{F}_{v_r} \neq w \mid \mathcal{H}_{t-1}\right] \\
&\qquad \leq \prod_{t < r \leq k} \Pr\left[\tilde{F}_{v_r} \neq w \mid \mathcal{H}_{t-1}\right](1 - 1 + x_{w,v_t}) \\
&\qquad \leq x_{w,v_t}
\end{aligned}$$

The second last inequality is due to the fact for any $r \geq t$ that $\Pr[\tilde{F}_{v_r} \neq w \mid \mathcal{H}_{t-1}] \geq (1 - x_{w,v_r})$. □

Lemma 3.6.5. *For any step $k > t$, and $w \in N(v_t)$, if $\tilde{F}_{v_t} = \perp$ i.e., v_t is not matched to any vertex incident on v_t then*

$$\Pr[w \textit{ is safe at } k \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = \perp] - \Pr[w \textit{ is safe at } k \mid \mathcal{H}_{t-1}] \leq x_{w,v_t}$$

Proof. For each vertex $w \in N(v_t)$, if $\tilde{F}_{v_t} = \perp$, a proof similar to that in Lemma 3.6.4 can be used. This is because, in both cases where $\tilde{F}_{v_t} = u$ or $\tilde{F}_{v_t} = \perp$, we have:

$$\Pr[\tilde{F}_{v_t} \neq w \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = u] = \Pr[\tilde{F}_{v_t} \neq w \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = \perp].$$

□

We now prove Lemma 3.6.6 which establishes a bound on the one-step changes in the random variable $Z(v_k)$ at any step t where $k \geq t$.

Lemma 3.6.6. *For any step $1 \leq t \leq n$ and $u \in N(v_t)$,*

- $\Big|\mathbb{E}\big[Z(v_t)\,|\,\mathcal{H}_{t-1}, \tilde{F}_{v_t}=u\big]-\mathbb{E}\big[Z(v_t)\,|\,\mathcal{H}_{t-1}\big]\Big| \leq \psi_c(u,v_t)+\sum_{w\in N(v_t)}\psi_c(w,v_t)x_{w,v_t}$
- $\Big|\mathbb{E}\big[Z(v_k)\mid \mathcal{H}_{t-1}, \tilde{F}_{v_t}=u\big]-\mathbb{E}\big[Z(v_k)\mid \mathcal{H}_{t-1}\big]\Big| \leq \psi_c(u,v_k)x_{u,v_k}+\sum_{w\in N_{\tilde{u}}(v_t)}\psi_c(w,v_k)x_{w,v_k}x_{w,v_t}$ *if* $k > t$.

Proof. We begin by proving the first part of the lemma. Recall that Z_{w,v_t} is the indicator random variable that the edge (w,v_t) is selected in the matching $\mathcal{M}$. From the definitions of $Z(v_t)$ and $\tilde{F}_{v_t}$, the probability $\Pr[\tilde{F}_{v_t}=w]=\Pr[A_{w,v_t}=1,F_{v_t}=w]$. This implies that $\Pr[\tilde{F}_{v_t}=w\mid\mathcal{H}_{t-1}]\leq x_{w,v_t}$. Therefore we have

$$\begin{aligned}&\mathbb{E}\big[Z(v_t)\mid\mathcal{H}_{t-1},\tilde{F}_{v_t}=u\big]-\mathbb{E}\big[Z(v_t)\mid\mathcal{H}_{t-1}\big]\\&=\sum_{w\in N(v_t)}\psi_c(w,v_t)\Big(\mathbb{E}\big[Z_{w,v_t}\mid\mathcal{H}_{t-1},\tilde{F}_{v_t}=u\big]-\mathbb{E}\big[Z_{w,v_t}\mid\mathcal{H}_{t-1}\big]\Big)\end{aligned}$$

We apply the bounds established in Lemma 3.8.1, specifically Equation (3.22) and Equation (3.23), and substitute them into the equation above. That implies

$$\mathbb{E}\big[Z(v_t)\mid\mathcal{H}_{t-1},\tilde{F}_{v_t}=u\big]-\mathbb{E}\big[Z(v_t)\mid\mathcal{H}_{t-1}\big]\leq\psi_c(u,v_t)-\psi_c(u,v_t)\frac{x_{u,v_t}}{2}-\sum_{w\in N_{\tilde{u}}(v_t)}\psi_c(w,v_t)x_{w,v_t}$$

Therefore, by applying triangle inequality we get the desired bound for the first part of the lemma as follows:

$$\Big|\big[Z(v_t)\mid\mathcal{H}_{t-1},\tilde{F}_{v_t}=u\big]-\mathbb{E}\big[Z(v_t)\mid\mathcal{H}_{t-1}\big]\Big|\leq\sum_{w\in N(v_t)}\psi_c(w,v_t)x_{w,v_t}+\psi_c(u,v_t).$$

For the second part, where $k > t$, we consider the bounds from Lemma 3.8.1 in Equation (3.24) and Equation (3.26) to obtain the following,

$$\left|\mathbb{E}[Z(v_k) \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = u] - \mathbb{E}[Z(v_k) \mid \mathcal{H}_{t-1}]\right| \leq \psi_c(u, v_k)x_{u,v_k} + \sum_{w \in N_{\bar{u}}(v_t)} \psi_c(w, v_k)x_{w,v_k}x_{w,v_t}$$

as desired. This concludes the proof of the lemma. □

Following this lemma, we can say that the one-step change in the size of the matching restricted to color class c at any step t, conditioned on the event $\tilde{F}_{v_t} = u$, is given by:

$$\mathbb{E}[M_n \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = u] - \mathbb{E}[M_n \mid \mathcal{H}_{t-1}] = \sum_{k \geq t} \mathbb{E}[Z(v_k) \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = u] - \mathbb{E}[Z(v_k) \mid \mathcal{H}_{t-1}].$$

Therefore, by using the Lemma 3.6.6 we can derive the following result about the expected one-step change in the size of the matching from color class c when $\tilde{F}_{v_t} = u$. Notice that these one step differences are in the worst case; therefore they can be as large as 2 as discussed in the previous section. However, when we compute the one-step variances, we get to take an expectation over the randomness of $\tilde{F}_{v_t}$ as shown in Lemma 3.6.7 and Lemma 3.6.8. This allows us to bypass the worst-case bound of 2.

Lemma 3.6.7. *For any step $1 \leq t \leq n$ and $u \in N(v_t)$,*

$$\sum_{t \in [n]} \sum_{u \in N(v_t)} x_{u,v_t} |\Delta M_t(u)| \leq 4M_0$$

where $\Delta M_t(u) = \mathbb{E}\big[M_n \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = u\big] - \mathbb{E}\big[M_n \mid \mathcal{H}_{t-1}\big]$.

Proof. Using Lemma 3.8.3 and the fact

$$\sum_{t\in[n]}\sum_{u\in N(v_t)}\psi_c(u,v_t)x_{u,v_t}=\sum_{(u,v)\in E}\psi_c(u,v)x_{u,v}=M_0,$$

we can say that

$$\begin{aligned}
&\sum_{t\in[n]}\sum_{u\in N(v_t)}x_{u,v_t}|\Delta M_t(u)|\\
&=\sum_{t\in[n]}\sum_{u\in N(v_t)}x_{u,v_t}\left|\mathbb{E}[M_n\mid\mathcal{H}_{t-1},\tilde{F}_{v_t}=u]-\mathbb{E}[M_n\mid\mathcal{H}_{t-1}]\right| \qquad (3.8)\\
&\le M_0+\sum_{t\ge1}\Big(\sum_{u\in N(v_t)}x_{u,v_t}\sum_{w\in N_{\bar{u}}(v_t)}\big(\psi_c(w,v_t)x_{w,v_t}+\sum_{k>t}\psi_c(w,v_k)x_{w,v_k}x_{w,v_t}\big)+\\
&\qquad\sum_{u\in N(v_t)}x_{u,v_t}\psi_c(u,v_k)x_{u,v_k}\Big)\\
&\le 2M_0+\sum_{(u,v)\in E}\sum_{w\in N(v)}\psi_c(w,v)x_{u,v}x_{w,v}+\sum_{t\ge1}\sum_{u\in N(v_t)}x_{u,v_t}\sum_{w\in N_{\bar{u}}(v_t)}\sum_{k>t}\psi_c(w,v_k)x_{w,v_k}x_{w,v_t} \qquad (3.9)
\end{aligned}$$

Equation (3.9) is due the fact that

$$\begin{aligned}
\sum_{t\ge1}\sum_{u\in N(v_t)}x_{u,v_t}\psi_c(u,v_k)x_{u,v_k}&\le\sum_{(u,v)\in E}x_{u,v}\sum_{v'\in N(u)}\psi_c(u,v')x_{u,v'}\\
&\le\sum_{(u,v')\in E_c}x_{u,v'}\sum_{v\in\delta(u)}x_{u,v}\le\sum_{(u,v')\in E_c}x_{u,v'}=M_0
\end{aligned}$$

Therefore, we have:

$$\sum_{t\in[n]}\sum_{u\in N(v_t)} x_{u,v_t}|\Delta M_t(u)|$$

$$\leq 2M_0 + \sum_{(w,v)\in E_c} x_{w,v}\sum_{u\in N(v)} x_{u,v} + \sum_{(u,v)\in E} x_{u,v}\sum_{w\in N_{\bar{u}}(v)}\sum_{v'\in N(w)} \psi_c(w,v')x_{w,v'}x_{w,v} \tag{3.10}$$

$$\leq 2M_0 + \sum_{(w,v)\in E_c} x_{w,v} + \sum_{(u,v)\in E}\sum_{w\in N(v)}\sum_{(w,v')\in E_c\cap\delta(w)} x_{w,v}x_{w,v'}x_{u,v} \tag{3.11}$$

$$\leq 2M_0 + M_0 + \sum_{(w,v')\in E_c} x_{w,v'}\sum_{v\in N(w)} x_{w,v}\sum_{u\in N(v)} x_{u,v} \tag{3.12}$$

$$= 3M_0 + \sum_{(w,v')\in E_c} x_{w,v'}\Big(\sum_{v\in N(w)} x_{w,v}\big(\sum_{u\in N(v)} x_{u,v}\big)\Big) \leq 3M_0 + \sum_{(w,v')\in E_c} x_{w,v'} \leq 4M_0$$

Note that each $\tilde{F}_{v_t} = u$ indicates that the edge (u, v_t) satisfies $F_{v_t} = u$ and $A_{u,v_t} = 1$. Therefore, the summation $\sum_{t\in[n]}\sum_{u\in N(v_t)}$ simplifies to a summation over the set of edges E as shown in Equation (3.10). We obtain Equation (3.11) due to the fact that $\sum_{s>t}\sum_{w\in N_{\bar{u}}(v_t)} a_{w,v_s}$ can be upper bounded by summing over all edges of color class c. The inequality (3.12) is true because we change the order of summations from $\sum_{(u,v)\in E}\sum_{w\in N(v)}\sum_{(w,v')\in E_c\cap\delta(w)}$ without changing the value of the summation. The final inequality follows from the fact that $\left(\sum_{v\in N(w)} x_{w,v}\left(\sum_{u\in N(v)} x_{u,v}\right)\right) \leq 1$ for any $w \in U$. This concludes the proof. □

We next consider the case when $\tilde{F}_{v_t} = \perp$. Recall that this means $F_{v_t} = \perp$, or $F_{v_t} = u$ for some $u \in N(v_t)$, yet $A_{u,v_t} = 0$.

Lemma 3.6.8. *For any step* $1 \leq t \leq n$,

$$\sum_{t\in[n]} |\mathbb{E}\big[M_n \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = \perp\big] - \mathbb{E}\big[M_n \mid \mathcal{H}_{t-1}\big]| \leq 2M_0.$$

Proof. In Lemma 3.8.2 we showed that the worst case one-step change ΔM_t when $\tilde{F}_{v_t} = \perp$ occurs. Therefore, we directly apply Equation (3.29) to get $\sum_{u \in N(v_t) \setminus \{u\}} \sum_{k \geq t} \psi_c(u, v_k) x_{u,v_k} x_{u,v_t} + \sum_{u \in N(v_t)} \psi_c(u, v_t) x_{u,v_t}$

$$\begin{aligned}
&\sum_{t \geq 1} \Big| \mathbb{E}[M_n \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = \perp] - \mathbb{E}[M_n \mid \mathcal{H}_{t-1}] \Big| \\
&\leq \sum_{t \geq 1} \sum_{u \in \delta(v_t)} \psi_c(u, v_t) x_{u,v_t} + \sum_{t \geq 1} \sum_{k > t} \sum_{u \in N(v_t)} \psi_c(u, v_k) x_{u,v_k} x_{u,v_t} \\
&= M_0 + \sum_{t \in [n]} \sum_{u \in N(v_t)} x_{u,v_t} \sum_{(u,v_k) \in E_c : k > t} x_{u,v_k} \qquad (3.13)
\end{aligned}$$

Finally, we can say that,

$$\begin{aligned}
\sum_{t \in [n]} \sum_{u \in N(v_t)} x_{u,v_t} \sum_{(u,v_k) \in E_c : k > t} x_{u,v_k} &\leq \sum_{t \in [n]} \sum_{u \in N(v_t)} x_{u,v_t} \sum_{v' : (u,v') \in E_c} x_{u,v'} \\
&\leq \sum_{(u,v) \in E} x_{u,v} \sum_{(u,v') \in E_c \cap \delta(u)} x_{u,v'} \\
&= \sum_{(u,v') \in E_c} x_{u,v'} \sum_{v \in N(u)} x_{u,v} \\
&\leq \sum_{(u,v') \in E_c} x_{u,v'} = M_0. \qquad (3.14)
\end{aligned}$$

Therefore, we get the desired bound by combining Equation (3.13) and Equation (3.14). □

Therefore, by combining Lemma 3.6.7 and Lemma 3.6.8, we can easily derive a bound on the sum of one-step variances i.e., $\sum_{t \in [n]} \mathrm{Var}(\Delta M_t \mid \mathcal{H}_{t-1})$ used in the Freedman's inequality. Formally, we can prove the following.

Lemma 3.6.9. $\sum_{t \in [n]} \mathrm{Var}(\Delta M_t \mid \mathcal{H}_{t-1}) \leq 12 M_0$ *where* $\mathrm{Var}(\Delta M_t \mid \mathcal{H}_{t-1})$ *is the variance in the*

one-step change of M_n at time t and $M_0 = \mathbb{E}[|\mathcal{M}_c|]$.

Proof. Letting $x(t) = \sum_{e \in \delta(v_t)} x_e$, we have the following.

$$\mathbb{E}[|\Delta M_t| \mid \mathcal{H}_{t-1}] = \sum_{u \in \tilde{F}_{v_t} \cup \{\perp\}} \Pr[\tilde{F}_{v_t} = u] \Big| \mathbb{E}[M_n \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = u] - \mathbb{E}[M_n \mid \mathcal{H}_{t-1}] \Big| \tag{3.15}$$

Inequality (3.15) is by definition and the fact that the random variable $\tilde{F}_{v_t}$ is independent of $\mathcal{H}_{t-1}$. The support of $\tilde{F}_{v_t}$ comprises of mainly two types of events (i) an edge $(u, v_t) \in \delta(v_t)$ has been both selected ($F_{v_t} = u$) and survived $A_{u,v_t} = 1$, and (ii) no edge gets sampled at time t i.e., $F_{v_t} = \tilde{F}_{v_r} = \perp$. We let $\tilde{F}_{v_t} = u$ represent the event that an edge (u, v_t) is both selected i.e., $F_{v_t} = u$ and $F_{v_t} \neq w$ for any $w \in N_{\bar{u}}(v_t)$ and $F_{v_t} = \perp$ represent the event that none of edges were selected. Since $\mathbb{E}[M_n \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = u] - \mathbb{E}[M_n \mid \mathcal{H}_{t-1}]$ and $\mathbb{E}[M_n \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = \perp] - \mathbb{E}[M_n \mid \mathcal{H}_{t-1}]$ represent the one-step change in M_n (the expected matching size restricted to color class c) when $\tilde{F}_{v_t} = u$ and $\tilde{F}_{v_t} = \perp$, respectively. By the definition of the random variable $\tilde{F}_{v_t}$ we have $\Pr[\tilde{F}_{v_t} = u] = x_{u,v_t}$ and $\Pr[\tilde{F}_{v_t} = \perp] = (1 - x(t))$, therefore

$$\begin{aligned}
&\mathbb{E}[|\Delta M_t| \mid \mathcal{H}_{t-1}] \\
&= \sum_{u \in N(v_t)} x_{u,v_t} \Big| \mathbb{E}[M_n \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = u] - \mathbb{E}[M_n \mid \mathcal{H}_{t-1}] \Big| + \\
&\quad (1 - x(t)) |\mathbb{E}[M_n \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = \perp] - \mathbb{E}[M_n \mid \mathcal{H}_{t-1}]|.
\end{aligned} \tag{3.16}$$

By summing Equation (3.16) over all time steps $t \in [n]$, we obtain the following after applying

Lemma 3.6.7, and Lemma 3.6.8.

$$\sum_{t\in[n]} \mathbb{E}[|\Delta M_t| \mid \mathcal{H}_{t-1}] \leq 4M_0 + 2M_0 \tag{3.17}$$

The last inequality follows from Lemma 3.6.8 and Lemma 3.6.7. Finally we can conclude that $\sum_{t\in[n]} V_t \leq 12M_0$ by using Observation 3.6.1. □

Therefore, by applying the Freedman's inequality (i.e., Theorem 3.6.2) we get the desired concentration for the size of matching from color class c.

Theorem 3.6.10. *For any color class c, Algorithm 4 returns a matching $\mathcal{M}_c = \mathcal{M} \cap E_c$ that is sharply concentrated around its expected value $M_0 = \mathbb{E}[|\mathcal{M}_c|]$. Specifically, for any $\delta > 0$, we have that:*

$$\Pr\Big[\big||\mathcal{M}_c| - M_0\big| \geq \delta M_0\Big] \leq 2\exp\Big(-\frac{\delta^2 \sum_{e\in E_c} x_e}{28}\Big)$$

Proof. We have $\Lambda = \max_t \Delta M_t \leq 2$. Applying Freedman's inequality from Theorem 3.6.2 to the martingale $(M_t)_{t\in[n]}$, where $M_n := |\mathcal{M}_c|$ represents the random variable denoting the number of edges of color c selected by our algorithm in the returned matching, we obtain:

$$\Pr[|M_0 - M_n| \geq \lambda] \leq 2\exp\Big(\frac{-\lambda^2}{2(\nu + \lambda(2))}\Big) \leq 2\exp\Big(\frac{-\lambda^2}{2(12M_0 + 2\lambda)}\Big).$$

By taking $\lambda = \delta M_0$ we get :

$$\Pr[|M_0 - M_n| \geq \delta M_0] \leq 2\exp\Big(\frac{-\delta^2 M_0^2}{24M_0 + 4\delta M_0}\Big) \leq 2\exp\Big(-\frac{\delta^2 M_0}{28}\Big)$$

as desired. □

This concludes that Property 2 described in Section 3.3 can be satisfied. Finally combining Property 2 together with the fairness constraints in (3.1b), we can conclude the following lemma.

Lemma 3.6.11. *For any $c \in [\ell]$, and $\delta > 0$, Algorithm 4 returns a matching $\mathcal{M}$ that satisfies the following:*

$$\Pr\left[(1-\delta)\alpha \leq \frac{|\mathcal{M}_c|}{|\mathcal{M}|} \leq (1+\delta)\beta\right] \geq 1 - 4\exp\left(-\frac{\delta^2 \sum_{e\in E_c} x_e}{28}\right)$$

Proof. From Theorem 3.6.10 we know that for any color class c the matching size $|\mathcal{M}_c|$ is concentrated sharply around the mean. Therefore, we have the following,

$$\Pr\left[|\mathcal{M}_c| \leq \frac{(1-\delta)}{2}\alpha \sum_{e\in E} x_e\right] \leq \Pr\left[|\mathcal{M}_c| \leq \frac{(1-\delta)}{2}\sum_{e\in E_c} x_e\right] \leq \exp\left(-\frac{\delta^2 \mathbb{E}[|\mathcal{M}_c|]}{28}\right)$$

A similar bound holds for the random variable $|\mathcal{M}|$ which is given by

$$\Pr\left[(1-\delta)\mathbb{E}[|\mathcal{M}|] \leq |\mathcal{M}| \leq (1+\delta)\mathbb{E}[|\mathcal{M}|]\right] \geq 1 - 2\exp\left(-\frac{\delta^2 \mathbb{E}[|\mathcal{M}|]}{28}\right)$$

Therefore, we have the following, for any $\delta_1, \delta_2 \geq 0$ such that $\frac{1-\delta_2}{1+\delta_2} \geq 1$,

$$\Pr\left[\left(|\mathcal{M}_c| \geq \frac{(1-\delta_1)}{2}\alpha \sum_{e\in E} x_e\right) \cap \left(|\mathcal{M}| \leq \frac{(1+\delta_2)}{2}\sum_{e\in E} x_e\right)\right] \geq 1 - \exp\left(\frac{-\delta^2 \sum_{e\in E} x_e}{28}\right) - \exp\left(\frac{-\delta^2 \sum_{e\in E_c} x_e}{28}\right)$$

$$\Pr\left[\frac{|\mathcal{M}_c|}{|\mathcal{M}|} \geq \frac{(1-\delta_1)}{(1+\delta_2)}\alpha\right)\right] \geq 1 - 2\exp\left(\frac{-\delta^2 \sum_{e\in E_c} x_e}{28}\right)$$

$$\Pr\left[\frac{|\mathcal{M}_c|}{|\mathcal{M}|} \geq (1-\delta)\alpha\right] \geq 1 - 2\exp\left(\frac{-\delta^2 \sum_{e\in E_c} x_e}{28}\right)$$

Similarly, we know that

$$\Pr\left[|\mathcal{M}_c| \leq \frac{(1+\delta)}{2}\beta\sum_{e\in E} x_e\right] \geq \Pr\left[|\mathcal{M}_c| \leq \frac{(1+\delta)}{2}\sum_{e\in E_c} x_e\right] \geq 1 - \exp\left(\frac{-\delta^2 \sum_{e\in E_c} x_e}{28}\right)$$

Moreover, we know that for any $\delta_1, \delta_2 > 0$ such that $\frac{1+\delta_1}{1-\delta_2} \leq 1$ we have:

$$\Pr\left[\left(|\mathcal{M}_c| \leq \frac{(1+\delta_1)}{2}\beta \sum_{e\in E} x_e\right) \cap \left(|\mathcal{M}| \geq \frac{(1-\delta_2)}{2}\sum_{e\in E} x_e\right)\right] \geq 1 - \exp\left(\frac{-\delta^2 \sum_{e\in E} x_e}{28}\right) - \exp\left(\frac{-\delta^2 \sum_{e\in E_c} x_e}{28}\right)$$

$$\Pr\left[\frac{|\mathcal{M}_c|}{|\mathcal{M}|} \leq \frac{(1+\delta_1)}{2}\cdot\frac{2}{(1-\delta_2)}\beta\right] \geq 1 - 2\exp\left(\frac{-\delta^2 \sum_{e\in E_c} x_e}{28}\right)$$

$$\Pr\left[\frac{|\mathcal{M}_c|}{|\mathcal{M}|} \leq (1+\delta)\beta\right] \geq 1 - 2\exp\left(\frac{-\delta^2 \sum_{e\in E_c} x_e}{28}\right).$$

Therefore, combining the two-sided fairness bounds we get the following, for any $\delta > 0$,

$$\Pr\left[(1-\delta)\alpha \leq \frac{|\mathcal{M}_c|}{|\mathcal{M}|} \leq (1+\delta)\beta\right] \geq 1 - f_c(\delta, G)$$

where $f_c(\delta, G) = 4\exp\left(\frac{-\delta^2 \sum_{e \in E_c} x_e}{28}\right)$. □

Lemma 3.6.11 proves that Algorithm 4 returns a δ-δ-PROBABLYALMOSTFAIR matching. This lemma, along with the edge selectability guarantees of the OCRS rounding algorithm as stated in Property 1 completes the proof of Theorem 3.5.1.

3.7 Exact fairness for one-sided constraints in Bipartite Graphs

In the previous section we showed that our algorithm provides a $1/2$-approximation of the objective, provided we violate the two-sided fairness constraints by multiplicative factors dependent on δ. Suppose that $\alpha = 0$ and only β imposes the fairness constraint, we can argue that by using a simple *LP perturbation* trick, we can remove the δ-violations and make it an *exact* guarantee while still attaining an asymptotic $1/2$-approximation of the objective. More precisely, we satisfy the β-sided fairness constraint *exactly*. We are also able to provide a better bound on our failure probability. Previously, this was $f_c(\delta, G)$, and it depended on $\sum_{e \in E_c} x_e$. We improve this to a new, *smaller* failure probability $f(\epsilon, G)$, which now only depends on $\beta \sum_{e \in E} x_e$ and a tunable parameter ϵ. This makes our algorithm more robust to the structure of the input instance.

If $\alpha = 0$, we modify our algorithm slightly. We begin by solving a perturbed LP where β in LP-FAIR is replaced by $\beta(1-\epsilon)$ where $0 < \epsilon < 1$. The perturbed LP requires that any feasible solution must satisfy a stronger β-fairness constraint, which results in a slightly weaker solution to our problem (see Lemma 3.7.1). Let $\boldsymbol{x}$ denote the optimal solution to the perturbed LP i.e., LP-FAIR with $\tilde{\beta}$. Since our algorithm still executes an OCRS on each $u \in U$ concurrently, Equation (3.3) ensures that the matching $\mathcal{M}$ has expected weight $\sum_{e \in E} w_e x_e/2$. We now relate $\sum_{e \in E} w_e x_e$ to the optimal solution to LP-FAIR when the fairness constraint is β. This will allow us to lower bound the expected weight

of our matching in terms of OPT in Corollary 3.7.2. (Recall that OPT is the weight of the optimal matching in the original problem with fairness constraint β.)

Lemma 3.7.1. *For any $0 < \epsilon < 1$, the optimal LP solution to* LP-FAIR *with $\tilde{\beta} = (1-\epsilon)\beta$ is at least $\sum_{e\in E} w_e x_e \geq (1-\epsilon)\sum_{e\in E} w_e y_e$, where $\boldsymbol{y} = (y_e)_{e\in E}$ is an optimal LP solution to* LP-FAIR *with β.*

Proof. Any feasible solution to LP-FAIR with $\alpha = 0$ and $\beta(1-\epsilon) < 1$ is also a feasible solution when $\alpha = 0$ and $\beta < 1$. Let $\boldsymbol{x} = (x_e)_{e\in E}$ and $\boldsymbol{y} := (y_e)_{e\in E}$ are optimal fractional solutions to LP-FAIR with $\tilde{\beta} = \beta(1-\epsilon)$ and $\tilde{\beta} = \beta$ respectively. Therefore,

$$\sum_{e\in E} w_e x_e \geq (1-\epsilon)\sum_{e\in E} w_e y_e. \tag{3.18}$$

□

Corollary 3.7.2. *Algorithm 4 returns a matching $\mathcal{M}$ with an expected weight of at least $\frac{1}{2}(1-\epsilon)$ OPT .*

We next argue that the matching $\mathcal{M}$ we output satisfies the β-fairness constraints exactly with a probability of at least $1 - f(\epsilon, G)$ where $f(\epsilon, G) = 2\exp\Big(-\frac{\epsilon^2\beta\sum_{e\in E} x_e}{28}\Big)$. Note that the failure probability of our algorithm now only depends on the parameter ϵ and $\sum_{e\in E} x_e$ (see Lemma 3.7.3). It is also reduced by a factor of 2, since we can use a one-sided version of Freedman's inequality when $\alpha = 0$ and $\beta > 0$. Here, $\sum_{e\in E} x_e$ refers to the size of optimal fractional matching of LP-FAIR with $\tilde{\beta} = \beta(1-\epsilon)$.

Lemma 3.7.3. *Given an optimal solution $\boldsymbol{x}$ to* LP-FAIR *with $\tilde{\beta} = (1-\epsilon)\beta$, then we can recover a solution that satisfies the constraints exactly i.e., no violations in β-fairness constraints. Formally, we can say that for any color c, and $0 < \epsilon < 1$, the size of the matching restricted to color class c is at*

most $\beta|\mathcal{M}|$ with high probability,

$$\Pr\left[\frac{|\mathcal{M}_c|}{|\mathcal{M}|} \leq \beta\right] \geq 1 - 2\exp\left(\frac{-\epsilon^2\beta\sum_{e\in E} x_e}{28}\right).$$

Proof. Recall that $\boldsymbol{x}$ denotes the optimal solution to LP-FAIR with $\tilde{\beta} = (1-\epsilon)\beta$ and $\mathcal{M}$ is the matching returned after the rounding $\boldsymbol{x}$ by Algorithm 4. Therefore we know that

$$\mathbb{E}[|\mathcal{M}_c|] \leq \sum_{e\in E_c} x_e \leq \tilde{\beta}\sum_{e\in E} x_e = (1-\epsilon)\beta\sum_{e\in E} x_e.$$

Using Lemma 3.8.5, we can say that

$$\Pr\left[|\mathcal{M}_c| \geq (1+\delta_1)(1-\epsilon)\beta\sum_{e\in E} x_e\right] \leq \exp\left(-\frac{\tilde{\beta}\delta_1^2\sum_{e\in E} x_e}{28}\right).$$

Additionally, we have that $|\mathcal{M}| \leq (1-\delta_2)\sum_{e\in E} x_e$ with probability $\exp\left(\frac{-\delta_2^2\sum_{e\in E} x_e}{2}\right)$ which follows from Chernoff-like bounds of [108] and the facts $\mathbb{E}[Z_{u,v}] \leq \Pr[F_v = u] = x_{u,v}$ for any $e = (u,v) \in E$. Therefore, we have the desired exact fairness as follows,

$$\Pr\left[\frac{|\mathcal{M}_c|}{|\mathcal{M}|} \geq \frac{(1-\epsilon)(1+\delta_1)}{(1-\delta_2)}\beta\right] \leq 2\exp\left(-\frac{\delta_1^2(1-\epsilon)}{28}\beta\sum_{e\in E} x_e\right)$$

We know that for any $\epsilon \geq \frac{\delta_1+\delta_2}{1+\delta_1}$, the term $\frac{(1-\epsilon)(1+\delta_1)}{(1-\delta_2)} \leq 1$ implying that

$$\Pr\left[\frac{|\mathcal{M}_c|}{|\mathcal{M}|} \leq \beta\right] \geq 1 - 2\exp\left(-\frac{\delta_1^2}{28(1-\epsilon)}\beta\sum_{e\in E} x_e\right) \geq 1 - 2\exp\left(-\epsilon^2\frac{\beta}{28}\sum_{e\in E} x_e\right)$$

The last inequality assumes $\delta_1 \geq (1-\epsilon)\epsilon$. □

Therefore we can conclude that the probability with which our algorithm fails to satisfy β-fairness constraints is now tightened from $f_c(\delta, G)$ which depends on $\sum_{e \in E_c} x_e$ of individual color classes, to a global quantity $\beta \sum_{e \in E} x_e$ in $f(\epsilon, G)$. This concludes the proof of Theorem 3.5.3.

3.7.1 Brute Force Algorithm for Small Instances

Given an instance of PROPORTIONALLYFAIRMATCHING, recall that the performance of Algorithm 4 improves the larger $\sum_{e \in E} \beta x_e$ is, where $\boldsymbol{x} = (x_e)_{e \in E}$ is an optimal solution to LP-FAIR with $\tilde{\beta} = (1 - \varepsilon)\beta$.

We now handle the case where $\sum_{e \in E} \beta x_e \leq C$ by running a simple brute-force algorithm. In order to see how to do this, let $\boldsymbol{y} = (y_e)_{e \in E}$ be an optimal solution to LP-FAIR with β. (We don't actually use $\boldsymbol{y}$ in our algorithm, it is just for the analysis). Define $L = \min_{e \in E} w_e$ and $U = \max_{e \in E} w_e$. Then,

$$U \sum_{e \in E} x_e \geq \sum_{e \in E} w_e x_e \geq (1 - \varepsilon) \sum_{e \in E} w_e y_e \geq (1 - \varepsilon) L \sum_{e \in E} y_e,$$

where the second inequality from the left uses Lemma 3.7.1, and the rest use the trivial upper and lower bounds of U and L. We thus have that

$$\frac{U}{L(1 - \varepsilon)} \sum_{e \in E} x_e \geq \sum_{e \in E} y_e. \tag{3.19}$$

Thus, if $\sum_{e \in E} \beta x_e \leq C$, then $\sum_{e \in E} y_e \leq \frac{U}{L(1-\varepsilon)} \frac{C}{\beta}$ via (3.19). Let us now suppose that M_{OPT} is a maximum weighted matching with respect to fairness constraint β. Then, $L|M_{\mathrm{OPT}}| \leq \sum_{e \in M_{\mathrm{OPT}}} w_e \leq \sum_{e \in E} w_e y_e \leq U \sum_{e \in E} y_e \leq U \frac{U}{\beta L} \frac{C}{1-\varepsilon}$, where the last inequality uses the assumed inequality $\sum_{e \in E} y_e \leq$

$\frac{U}{\beta L}\frac{C}{1-\varepsilon}$. It follows that

$$|M_{OPT}| \leq \frac{U^2}{L^2}\frac{C}{\beta(1-\varepsilon)}. \tag{3.20}$$

Thus, if we run a brute force algorithm where we check matchings which are fair with respect to β, and which contain at most $\frac{U^2}{L^2}\frac{C}{\beta(1-\varepsilon)}$ edges, then (3.20) ensures that we'll return an optimal matching.

Example demonstrating the Worst Case one-step change

Observation 3.7.1. *For any $t \geq 1$ and $\epsilon > 0$, the one-step change in the martingale can be as large as $2-\epsilon$, i.e., $\Delta M_t \geq 2-\epsilon$.*

Proof. We establish this by constructing an example where the one-step change in the martingale is precisely $2-\epsilon$. Consider the Example 3.7.1 where, at time $t=1$, vertex v_t proposes to vertex u. In this case, the one-step change in the matching from the blue color class, $\mathcal{M}_{\text{blue}}$, given $\mathcal{H}_{t-1}$, is $2-\epsilon$. □

Example 3.7.1. Consider an instance of proportional matching in an edge-colored graph $G=(U,V,E)$, where $E = E_{\text{red}} \cup E_{\text{blue}}$. Suppose the optimal fractional solution has $x_{u,v_k} = x_{w,v_t} = 1-\epsilon$ and $x_{u,v_t} = \epsilon$.

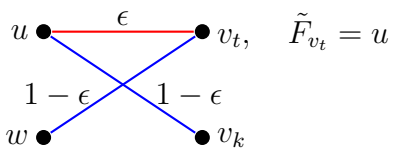


Figure 3.1: At $t=1$ when v_t proposes to u instead of w and $\tilde{F}_{v_t} = u$, then the expected change in the matching restricted to blue edges, $\mathcal{M}_{\text{blue}}$, is bounded by $2(1-\epsilon)$.

3.8 Omitted proofs

Lemma 3.8.1. *For any $t \geq 1$ when vertex v_t is processed, the worse case one-step change given $\tilde{F}_{v_t} = u$ is as follows:*

$$\max_{u \in N(v_t)} |\mathbb{E}[M_n \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = u] - \mathbb{E}[M_n \mid \mathcal{H}_{t-1}]| \leq 2$$

Proof. Recall that Z_{w,v_t} denotes the indicator random variable that the edge (w, v_t) is selected in the matching $\mathcal{M}$. Given the definition of M_t in Equation (3.6) and conditioned on $\tilde{F}_{v_t} = u$ for any $u \in N(v_t)$, the one-step change is given by:

$$\begin{aligned}
&\mathbb{E}\Big[\sum_{k\geq 1} Z(v_k) \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = u\Big] - \mathbb{E}\Big[\sum_{k\geq 1} Z(v_k) \mid \mathcal{H}_{t-1}\Big] \\
&= \sum_{k\geq t} \mathbb{E}\big[Z(v_k) \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = u\big] - \mathbb{E}\big[Z(v_k) \mid \mathcal{H}_{t-1}\big] \\
&= \sum_{k\geq t} \sum_{w \in N(v_k)} \psi_c(w, v_k)\Big(\mathbb{E}\big[Z_{w,v_k} \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = u\big] - \mathbb{E}\big[Z_{w,v_k} \mid \mathcal{H}_{t-1}\big]\Big) \qquad (3.21)
\end{aligned}$$

Notice that for any (w, v_k) with $w \notin N(v_t)$ are not affected by the conditional on $\tilde{F}_{v_t} = u$ i.e.,

$$\mathbb{E}\big[Z_{w,v_k} \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = u\big] = \mathbb{E}\big[Z_{w,v_k} \mid \mathcal{H}_{t-1}\big].$$

Therefore, we need to bound the term $\mathbb{E}[Z_{w,v_k} \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = u] - \mathbb{E}[Z_{w,v_k} \mid \mathcal{H}_{t-1}]$ for any $k \geq t$ and $w \in N(v_t)$ in order to obtain the desired bound stated in the lemma. Additionally, these edges can be classified into one of the following cases.

Case 1: $k = t$, $w = u$, and $\tilde{F}_{v_t} = u$

$$\begin{aligned}\mathbb{E}\big[Z_{u,v_t} \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = u\big] - \mathbb{E}\big[Z_{u,v_t} \mid \mathcal{H}_{t-1}\big] &\leq \Pr\left[u \text{ is safe at } t \mid \mathcal{H}_{t-1}\right](1 - \Pr[\tilde{F}_{v_t} = u \mid \mathcal{H}_{t-1}]) \\ &\leq 1 - \frac{x_{u,v_t}}{2}. \end{aligned} \tag{3.22}$$

The last inequality is from Equation (3.3).

Case 2: $k = t$, $w \neq u$, and $\tilde{F}_{v_t} = u$

$$\mathbb{E}\big[Z_{w,v_t} \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = u\big] - \mathbb{E}\big[Z_{w,v_t} \mid \mathcal{H}_{t-1}\big] \geq 0 - x_{w,v_t} \Pr\left[w \text{ is safe at } t \mid \mathcal{H}_{t-1}\right] \geq -x_{w,v_t} \tag{3.23}$$

Case 3: $k > t$, $w = u$, and $\tilde{F}_{v_t} = u$

$$\mathbb{E}\big[Z_{u,v_k} \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = u\big] - \mathbb{E}\big[Z_{u,v_k} \mid \mathcal{H}_{t-1}\big] = 0 - x_{u,v_k} \Pr\left[u \text{ is safe at } t \mid \mathcal{H}_{t-1}\right] \geq -x_{u,v_k} \tag{3.24}$$

Case 4: $k > t$, $w \neq u$, and $\tilde{F}_{v_t} = u$

$$\mathbb{E}\big[Z_{w,v_k} \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = u\big] - \mathbb{E}\big[Z_{w,v_k} \mid \mathcal{H}_{t-1}\big] \tag{3.25}$$

$$\leq x_{w,v_k}\Big(\Pr\left[w \text{ is safe at } k \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = u\right] - \Pr\left[w \text{ is safe at } k \mid \mathcal{H}_{t-1}\right]\Big)$$

$$\leq x_{w,v_k} x_{w,v_t} \tag{3.26}$$

Inequality (3.26) follows from Lemma 3.6.4. Hence, substituting the bounds from the four cases into Equation (3.21) we have:

$$\begin{aligned}
&\mathbb{E}\left[\sum_{k\geq 1} Z(v_k) \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = u\right] - \mathbb{E}\left[\sum_{k\geq 1} Z(v_k) \mid \mathcal{H}_{t-1}\right] \\
&\leq \psi_c(u, v_t)\big(1 - \frac{x_{u,v_t}}{2}\big) - \sum_{w\in N(v_t)\setminus\{u\}} \psi_c(w, v_t)x_{w,v_t} - \\
&\quad \sum_{k>t} \psi_c(u, v_k)x_{u,v_k} + \sum_{k>t} \sum_{w\in N(v_t)\setminus\{u\}} \psi_c(w, v_k)x_{w,v_k}x_{w,v_t} \\
&\leq \psi_c(u, v_t)\big(1 - \frac{x_{u,v_t}}{2}\big) - \sum_{w\in N_{\tilde{u}}(v_t)} \psi_c(w, v_t)x_{w,v_t} - \sum_{k>t} \psi_c(u, v_k)x_{u,v_k} + \\
&\quad \sum_{k>t} \sum_{w\in N_{\tilde{u}}(v_t)\setminus\{u\}} \psi_c(w, v_k)x_{w,v_k}x_{w,v_t} \\
&\leq 1 - \frac{x_{u,v_t}}{2} - \sum_{w\in N_{\tilde{u}}(v_t)} \psi_c(w, v_t)x_{w,v_t} - \sum_{k>t} \psi_c(u, v_k)x_{u,v_k} + (1 - x_{u,v_t}) \\
&\leq 2.
\end{aligned}$$

The second-to-last inequality follows from the fact:

$$\sum_{k>t} \sum_{w\in N(v_t)\setminus\{u\}} \psi_c(w, v_k)x_{w,v_k}x_{w,v_t} = \sum_{w\in N(v_t)\setminus\{u\}} x_{w,v_t} \sum_{k>t} \psi_c(w, v_k)x_{w,v_k} \leq \sum_{w\in N(v_t)\setminus\{u\}} x_{w,v_t} \leq 1 - x_{u,v_t}.$$

The final inequality is due to:

$$\sum_{w\in N(v_t)} \psi_c(w, v_t)x_{w,v_t} + \frac{1}{2}x_{u,v_t} + \sum_{k>t} \psi_c(u, v_k)x_{u,v_k} \leq 2$$

and this concludes the proof. □

Lemma 3.8.2. *For any $t \geq 1$ when vertex v_t is processed, the worse case one-step change given*

$\tilde{F}_{v_t} = \perp$ *is as follows:*

$$\mathbb{E}[M_n \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = \perp] - \mathbb{E}[M_n \mid \mathcal{H}_{t-1}] \leq 2$$

Proof. Following the definitions of M_t in Equation (3.6) we have:

$$\begin{aligned}
&\mathbb{E}[M_n \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = \perp] - \mathbb{E}[M_n \mid \mathcal{H}_{t-1}] \\
&= \sum_{k \geq 1} \mathbb{E}[Z(v_k) \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = \perp] - \mathbb{E}[Z(v_k) \mid \mathcal{H}_{t-1}] \\
&= \sum_{k \geq t} \sum_{u \in N(v_t)} \psi_c(u, v_k) \big(\mathbb{E}[Z_{u,v_k} \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = \perp] - \mathbb{E}[Z_{u,v_k} \mid \mathcal{H}_{t-1}] \big)
\end{aligned} \tag{3.27}$$

Notice that for any (u, v_k) with $u \notin N(v_t)$ are not affected by the conditional on $\tilde{F}_{v_t} = \perp$ i.e.,

$$\mathbb{E}\big[Z_{u,v_k} \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = \perp\big] = \mathbb{E}\big[Z_{u,v_k} \mid \mathcal{H}_{t-1}\big].$$

Therefore, we need to bound the term $\mathbb{E}[Z_{u,v_k} \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = \perp] - \mathbb{E}[Z_{u,v_k} \mid \mathcal{H}_{t-1}]$ for any $k \geq t$ and $u \in N(v_t)$ in order to obtain the desired bound stated in the lemma. Further, these edges can be classified into one of two cases.

Case 1: $k = t$, $u \in N(v_t)$ and $\tilde{F}_{v_t} = \perp$

$$\begin{aligned}
\mathbb{E}\big[Z_{u,v_t} \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = \perp\big] - \mathbb{E}\big[Z_{u,v_t} \mid \mathcal{H}_{t-1}\big] &= 0 - \Pr\big[u \text{ is safe at } t \mid \mathcal{H}_{t-1}\big] \Pr[\tilde{F}_{v_t} = u] \\
&\geq -x_{u,v_t}
\end{aligned}$$

Case 2: $k > t$, $u \in N(v_t)$, and $\tilde{F}_{v_t} = \perp$

$$\begin{aligned}
&\mathbb{E}\big[Z_{u,v_k} \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = \perp\big] - \mathbb{E}\big[Z_{u,v_k} \mid \mathcal{H}_{t-1}\big] \\
&= \Pr[\tilde{F}_{v_k} = u]\big(\Pr[u \text{ is safe at } k \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = \perp] - \Pr[u \text{ is safe at } k \mid \mathcal{H}_{t-1}]\big) \\
&\leq x_{u,v_k} x_{u,v_t}. \qquad (3.28)
\end{aligned}$$

Inequality (3.28) follows from Lemma 3.6.5. Hence, substituting the bounds from the four cases into Equation (3.27) we have:

$$\begin{aligned}
\mathbb{E}[M_n \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = \perp] - \mathbb{E}[M_n \mid \mathcal{H}_{t-1}] &\leq \sum_{k \geq t} \sum_{u \in N(v_t) \setminus \{u\}} \psi_c(u, v_k) x_{u,v_k} x_{u,v_t} + \sum_{u \in N(v_t)} \psi_c(u, v_t) x_{u,v_t} \\
&\leq \sum_{u \in N(v_t) \setminus \{u\}} \sum_{k \geq t} \psi_c(u, v_k) x_{u,v_k} x_{u,v_t} + \sum_{u \in N(v_t)} \psi_c(u, v_t) x_{u,v_t} \qquad (3.29) \\
&\leq 1 + 1 = 2.
\end{aligned}$$

The last inequality is due to

$$\sum_{u \in N(v_t) \setminus \{u\}} \sum_{k \geq t} \psi_c(u, v_k) x_{u,v_k} x_{u,v_t} = \sum_{u \in N(v_t) \setminus \{u\}} x_{u,v_t} \sum_{k \geq t} \psi_c(u, v_k) x_{u,v_k} \leq \sum_{u \in N(v_t) \setminus \{u\}} x_{u,v_t} \leq 1.$$

This completes the proof. □

Lemma 3.8.3. *Suppose that $\tilde{F}_{v_t} = u$ represent the event that v_t proposed to u and $A_{u,v_t} = 1$, then*

$$\sum_{u \in N(v_t)} x_{u,v_t} \Big| \mathbb{E}[M_n \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = u] - \mathbb{E}[M_n \mid \mathcal{H}_{t-1}] \Big|$$

$$\leq \sum_{u \in N(v_t)} x_{u,v_t} \Big(\psi_c(u, v_t) + \sum_{w \in N(v_t)} \big(a_{w,v_t} x_{w,v_t} + \sum_{k>t} (\psi_c(u, v_k) x_{u,v_k} + a_{w,v_k} x_{w,v_k} x_{w,v_t}) \big) \Big)$$

Proof. By applying Lemma 3.6.6 we have the following,

$$\sum_{u \in N(v_t)} x_{u,v_t} \Big| \mathbb{E}[M_n \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = u] - \mathbb{E}[M_n \mid \mathcal{H}_{t-1}] \Big|$$

$$= \sum_{u \in N(v_t)} x_{u,v_t} \Big| \mathbb{E}\Big[\sum_{k \geq t} Z(v_k) \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = u \Big] - \mathbb{E}\Big[\sum_{k \geq t} Z(v_k) \mid \mathcal{H}_{t-1} \Big] \Big|$$

$$\leq \sum_{u \in N(v_t)} x_{u,v_t} \sum_{k \geq t} \Big| \mathbb{E}\big[Z(v_k) \mid \mathcal{H}_{t-1}, \tilde{F}_{v_t} = u \big] - \mathbb{E}\big[Z(v_k) \mid \mathcal{H}_{t-1} \big] \Big|$$

$$\leq \sum_{u \in N(v_t)} x_{u,v_t} \big(\psi_c(u, v_t) + \sum_{w \in N(v_t)} \psi_c(w, v_t) x_{w,v_t} \big) +$$

$$\sum_{u \in N(v_t)} x_{u,v_t} \sum_{k>t} \big(\psi_c(u, v_k) x_{u,v_k} + \sum_{w \in N_{\tilde{u}}(v_t)} \psi_c(w, v_k) x_{w,v_k} x_{w,v_t} \big) \tag{3.30}$$

$$= \sum_{u \in N(v_t)} x_{u,v_t} \big(\psi_c(u, v_t) + \sum_{w \in N(v_t)} \psi_c(w, v_t) x_{w,v_t} + \sum_{k>t} (\psi_c(u, v_k) x_{u,v_k} + \sum_{w \in N(v_t)} \psi_c(w, v_k) x_{w,v_k} x_{w,v_t}) \big) \tag{3.31}$$

Equation (3.30) follows directly from Lemma 3.6.6. Thus, we obtain the desired bound. □

In order to attain the desired one-sided concentration inequality as stated in Lemma 3.7.3, we will a variant of Freedman's inequality for supermartingales. This inequality is the same as Theorem 3.6.2, except that since it only controls the upper tail of the random variable, the probability term is reduced by a factor of 2.

Theorem 3.8.4 ([109]). *Suppose $(M_t)_{t\in[n]}$ is a supermartingale such that $|\Delta M_t| \leq \Lambda$ for any $1 \leq t \leq n$, and $\sum_{t\in[n]} \mathbb{E}[\Delta M_t^2 \mid \mathcal{H}_{t-1}] \leq \nu$. Then, for all $\lambda > 0$,*

$$\Pr\left[M_n \geq M_0 + \lambda\right] \leq \exp\left(\frac{-\lambda^2}{2(\nu + \lambda\Lambda)}\right)$$

Lemma 3.8.5. *Suppose that $(M_t)_{t\geq 0}$ is the Doob martingale defined in Section 3.6 and $M_0 \leq \mu_H$ i.e., we know an upper bound on the expected size of the matching from color class c, then*

$$\Pr[M_n \geq (1+\delta)\mu_H] \leq \exp\left(\frac{-\delta^2\mu_H}{28}\right)$$

Proof. Consider the following sequence of random variables $(S_t)_{t\geq 0}$ where $S_t := M_t - \frac{\tilde{\beta}}{2}\sum_{e\in E} x_e$. Clearly, $(S_t)_{t\geq 0}$ is a supermartingale since $(M_t)_{t\geq 0}$ is a martingale and $M_0 \leq \frac{\tilde{\beta}}{2}\sum_{e\in E} x_e$. The latter is true because,

$$M_0 = \mathbb{E}[|\mathcal{M}_c|] = \sum_{e\in E_c} x_e/2 \leq \tilde{\beta}\sum_{e\in E} x_e/2 = \mu_H$$

This implies the following upper tail bound on S_n:

$$\Pr[S_n \geq \lambda] \leq \Pr[S_n \geq S_0 + \lambda] \leq \exp\left(-\frac{\lambda^2}{2\left(\sum_{t\in[n]} \mathbb{E}(|\Delta S_t|^2 \mid \mathcal{H}_{t-1}) + \lambda\right)}\right)$$
$$\Pr[M_n \geq \lambda + \tilde{\beta}\sum_{e\in E} x_e/2] \leq \exp\left(-\frac{\lambda^2}{2\left(\sum_{t\in[n]} \mathbb{E}(2|\Delta M_t| \mid \mathcal{H}_{t-1}) + \lambda\right)}\right).$$

The last inequality is due to the fact that $|\Delta S_t| = |\Delta M_t|$. Therefore, we have that

$$\Pr[M_n \geq \tilde{\beta}\sum_{e\in E} x_e/2 + \lambda] \leq \left(-\frac{\lambda^2}{2\big(12M_0 + 2\lambda\big)}\right).$$

By substituting $\lambda = \delta\frac{\tilde{\beta}}{2}\sum_{e\in E} x_e \geq \delta M_0$,

$$\begin{aligned}\Pr[M_n \geq (1+\delta)\mu_H] &\leq \exp\left(-\frac{\lambda^2\delta}{2\big(10\lambda + 2\lambda\delta\big)}\right)\\ &= \exp\Big(-\frac{\lambda\delta}{28}\Big)\\ &= \exp\Big(-\frac{\delta^2\mu_H}{28}\Big)\end{aligned}$$

This concludes the proof. □

3.9 Failure of Azuma-Hoeffding inequality

Theorem 3.9.1 (Azuma–Hoeffding inequality)**.** *Suppose that* $(M_t)_{t\geq 0}$ *is a martingale and* $|M_t - M_{t-1}| \leq c_t$ *for any* $1 \leq t \leq n$*. Then for any* $\delta > 0$

$$\Pr[|M_n - M_0| \geq \delta] \leq 2\exp\left(\frac{-\delta^2}{2\sum_{t=1}^{n} c_t^2}\right).$$

Example 3.9.1. Consider an edge-colored star graph G with 2 colors and $n+2$ vertices. Let u be the central vertex, and let the remaining vertices be $v_1, v_2, \ldots, v_{n+1}$. The edges partitions are as follows: $E_{\text{blue}} = \{(u, v_t) \mid t = 1, \ldots, n\}$ and $E_{\text{red}} = \{(u, v_{n+1})\}$. An optimal solution to the linear program (LP) LP-FAIR is given by $x_{u,v_t} = \frac{\epsilon}{n}$ for $1 \leq t \leq n$ and $x_{u,v_{n+1}} = 1 - \epsilon$.

Let us fix the color class *red* and $\mathcal{M}_{\text{red}}$ denote $\mathcal{M} \cap E_{\text{red}}$ where $\mathcal{M}$ is the matching returned by our algorithm. Let the martingale $(M_t)_{t\geq 0}$ correspond to the matching restricted to color class c. Then we have the following observation.

Observation 3.9.1. *There exists an instance of proportional fair matching where the worst-case sum*

$\sum_{t\geq 1} c_t$ *can grow as large as* $\Theta(n)$*, where* n *represents the number of vertices and* c_t *are positive constants such that* $|M_t - M_{t-1}| \leq c_t$*.*

Proof. For each $1 \leq t \leq n$, when v_t is processed we know that the one-step change in the $\mathcal{M}_c$ is given by $|\Delta M_t| = 1 - \epsilon$. Therefore we have $\sum_{t\in[n]} c_t^2 \leq \sum_{t\in[n]} c_t = n(1-\epsilon)$. □

Therefore, we can conclude that Azuma's inequality may fail to provide concentration for $|\mathcal{M}_c|$ since $\mathbb{E}[|\mathcal{M}_c|] \leq n_c$, where n_c represents the total number of vertices with at least one incident edge from color class c.

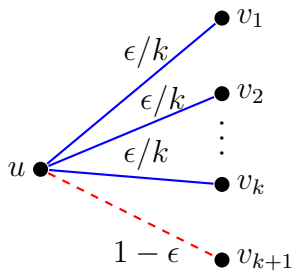


Figure 3.2: In the edge-colored graph G, we have $|\Delta M_t| = 1-\epsilon$ at each time t. Therefore, the sum $\sum_{t\in[n]} |\Delta M_t|$ is $\Theta(n)$.

Chapter 4: Fair Set Packing

This chapter is largely based on our paper [48].

4.1 Introduction

Fielded kidney exchange programs (KEPs) often employ a utilitarian objective, where the overall utility of successful transplants is maximized. This approach, however, can penalize certain classes of hard-to-match patients (e.g., highly-sensitized, older, O-type blood, and others), as highlighted by many studies from the AI/ML [38, 39, 40, 41], economics [42, 43], and medical communities [44]. Subsequent research by [45] explored two group-fairness criteria—lexicographical fairness and weighted fairness—under random-graph models, and determined that the tradeoff between efficiency and fairness, known as the "price of fairness," is relatively small. However, it should be noted that these guarantees apply only under the assumption of stochastic models in kidney exchange and there is currently a lack of fair algorithms with provable guarantees for the general problem—*directed cycle packing*—as originally defined [46, 110]. Several works studied approximation algorithms for KEPs under this model [46, 47, 111] but none of those algorithms consider fairness; therefore the problem of group fairness in kidney exchange on arbitrary patient-donor graphs presents a unique and intriguing challenge, which can be approached from both theoretical computer science and economics perspectives.

KEPs are often formulated as cycle-packing problems with bounded length cycles, say cycle length at most k. This, in turn, can be reduced to k-set packing[1] by naively enumerating the cycles in $\mathcal{O}(n^k)$ worst-case time [113]; each cycle, which has l (less than or equal to k) vertices, represents a subset of l elements in the k-set packing formulation. For the rest of the paper, we lean into kidney exchange to illustrate the need for fairness in the well-known k-set packing problem, though other suitable applications exist such as crew scheduling and barter exchanges. We conduct experiments over a realistic kidney-exchange dataset.

We study here the notion of group-fairness in k-set packing problems. In our model, each of the n elements in the universe $\mathcal{U}$ is assigned a color (category), and our goal is to compute a valid packing such that (i) each group is *fairly* represented satisfying exogenous proportionality constraints and (ii), subject to (i), the overall weight of the packing is maximized. On the flip side, our work is not solely meant to correct so-called unfair algorithms. Rather, the tools we develop also serve as a basis to understand and audit the group-level behavior of algorithms. That is to say, the lack of studies of group-level fairness in k-set packing is representative of missing, but necessary, work towards understanding group outcomes in kidney exchange, equitable crew scheduling, and k-set packing's other applications. Thus, we view our work as an important step towards understanding group-level fairness in kidney exchange.

Let $[t]$ denote the set $\{1, 2, \ldots, t\}$. In *fair* k-set packing, denoted $\overline{\text{FAIRSP}}$, we are given a set of elements $\mathcal{U} = [n]$ and a collection $\mathcal{S} = \{S_1, \ldots, S_m\}$ of subsets of $\mathcal{U}$ where $|S_i| \leq k$ for each S_i. Each subset $S_i \in \mathcal{S}$ has weight $w_i \geq 0$. Moreover, each $u \in \mathcal{U}$ is assigned one of ℓ colors by $\mathcal{C} : \mathcal{U} \to [\ell]$; i.e., each u belongs to one of ℓ groups. We are also given a proportionality vector $\vec{\alpha} \in [0, 1]^\ell$ such that

[1]The k-set packing problem is set packing over a universe $\mathcal{U}$ and collection of subsets $\mathcal{S}$ of $\mathcal{U}$ restricted to each $S \in \mathcal{S}$ having at most k elements. For $k \geq 3$, it is NP-hard [112].

$\sum_{g\in[\ell]} \alpha_g = 1$. The objective of $\overline{\text{FAIRSP}}$ is to select a packing[2] of maximum weight subject to a α_g fraction of the packing consisting of elements with color g: i.e., letting $y_i = 1$ if S_i is selected and $y_i = 0$ otherwise, we aim to maximize $\sum_{i\in[m]} w_i y_i$ subject to the proportionality constraint.

However, many instances become immediately infeasible under these additional constraints. For example, let us consider an instance with $\mathcal{U} = \{r, b_1, b_2, g_1, g_2\}$, $S_1 = \{r, b_1, b_2\}$, $S_2 = \{r, g_1, g_2\}$, and $\vec{\alpha} = [1/3, 1/3, 1/3]$; here r, b_i's, and g_i's are colored red, blue, and green, respectively. Clearly, there does not exist a feasible (non-empty) fair packing. Instead, the fairness constraints can be satisfied in expectation by switching our target to a *distribution* over packings (say, pick each set with probability $1/2$).

Observation 4.1.1. *The integrality gap of the LP corresponding to* $\overline{\text{FAIRSP}}$ *is unbounded.*

Therefore, we shift our attention to randomized notions of fairness. Such probabilistic fairness guarantees have been previously studied by [114, 115]. This motivates our work's exploration of randomized algorithms and probabilistic fairness.

Any randomized packing algorithm ALG is captured by a distribution $\mathcal{D}$ over the set of all packings; i.e., running ALG is akin to sampling from the distribution $\mathcal{D}$. Let $Z_i = 1$, $i \in [m]$, indicate that S_i is selected in the packing returned by ALG ($Z_i = 0$ otherwise). Letting $v_{i,g}$ be the number of elements of color g in S_i, we have $N_g := \sum_{i\in[m]} v_{i,g} Z_i$ and $N := \sum_{g\in[\ell]} N_g$. We focus our study on FAIRSP, defined as follows: Given $\mathcal{U}$, $\mathcal{S}$, $w_i \geq 0$, $\mathcal{C} : \mathcal{U} \to [\ell]$ (as defined in $\overline{\text{FAIRSP}}$), FAIRSP seeks a distribution $\mathcal{D}$ such that $\mathbb{E}_{Z\sim\mathcal{D}}[\sum_{i\in[m]} w_i Z_i]$ is maximized while satisfying the proportional constraints in *expectation* i.e., for any color ℓ, $\mathbb{E}[N_g] = \alpha_g \mathbb{E}[N]$ holds. We require $\mathbb{E}[N] > 0$ to rule out the distribution with support only over the empty set. In this paper, we study randomized algorithms

[2] A packing is a collection of pairwise disjoint subsets of $\mathcal{S}$.

for FAIRSP, with a focus on two specific randomized fairness notions outlined in Section 4.2. We drop the word *probabilistic* while referring to the proportionality constraints of FAIRSP for brevity.

4.2 Preliminaries and Main Contributions

Throughout this paper, we denote $[n] = \{1, \ldots, n\}$ for any positive integer n; we use OPT to denote both an optimal algorithm and its objective value; and analogously ALG for both a generic algorithm and its objective value. We use ϵ_n to denote a vanishing term i.e., $\lim_{n\to 0} \epsilon_n = 0$. We use either $\{Y_i\}$ or $\{Y_i\}_{i\in[m]}$ to refer to the set of variables $\{Y_1, Y_2, \ldots, Y_m\}$ interchangeably.

4.2.1 Approximation Factor

For NP-hard combinatorial optimization problems, we utilize the powerful framework of approximation algorithms, where the aim is to design efficient algorithms (polynomial running time) with provably near-optimal objectives values. We say ALG achieves an approximation factor of $c \geq 1$ if $c\mathbb{E}[\text{ALG}] \geq \text{OPT}$ across all feasible FAIRSP instances.

4.2.2 Parameters

The complexity of FAIRSP can depend on the following parameters; number of colors c; maximum element frequency $f \in [m]$, where the frequency of an element is the number of sets it belongs to; and $k \in [n]$. The k-restrained variant is *necessary* in applications such as crew scheduling and kidney exchange, as motivated by prior work [116]. Moreover, the natural LP for k-set packing (LP (4.3) without the constraints (4.3c)) has integrality gap of at least $k - 1 + \frac{1}{k}$ and there exists a $(k - \epsilon_k)$-approximate algorithm by [117]. Note our algorithmic results are parameterized on k, ℓ and f.

We study algorithms with approximation guarantees on the packing objective and fairness of FAIRSP. We make this precise via two related fairness guarantees for randomized algorithms of FAIRSP.

4.2.3 Randomized notions of fairness

We define the fairness notions for a randomized algorithm ALG as follows. Let N_g be the random variable representing the number of elements selected by ALG from group g.

1. **Randomized Fairness (RF)**: An algorithm ALG satisfies Randomized Fairness (RF) constraints if, for any two groups $g, h \in [\ell]$:

$$\frac{\mathbb{E}[N_g]}{\mathbb{E}[N_h]} \leq \gamma \left(\frac{\alpha_g}{\alpha_h}\right) \tag{4.1}$$

   This guarantee ensures fairness ex-ante (in expectation). However, it is possible for a single realized packing to be very unfair.

2. **Strong Randomized Fairness (SRF)**: To ensure fairness in actual realizations, we introduce a stronger notion called γ-PROBABLYAPPROXFAIR (SRF). An algorithm is said to satisfy γ-PROBABLYAPPROXFAIR if the following holds:

$$\Pr\left[\bigwedge_{g,h\in[\ell]} \left(\frac{N_g}{N_h} \leq \gamma \left(\frac{\alpha_g}{\alpha_h}\right)\right)\right] \geq 0.99. \tag{4.2}$$

In Section 4.3 we give algorithms for FAIRSP with varied approximation guarantees on the packing objective and RF or SRF guarantees with *fairness factor* γ.

4.2.4 Connections to existing works

For $k \geq 3$, the problem of k-set packing is well-studied, and several previous works (such as [117, 118, 119]) have focused on developing algorithms based on the natural LP relaxation of k-set packing. The primary motivation for using LP rounding techniques is the ease of incorporating proportionality constraints as *additional* constraints in the LP, as demonstrated in LP (4.3). Our main *technical challenge* is to adapt these rounding algorithms in such a way that they produce packings with a small approximation factor and fairness ratio γ simultaneously.

4.2.5 Choosing a proportionality vector

Previous studies such as [120, 121] have examined a concept of proportional fairness that is broader than the one used in our work. They apply upper and lower bound constraints on the desired proportional representation of each group in the solution. In contrast, we use a special case where the upper and lower bounds coincide, meaning we insist on the final packing solution exactly preserving the proportional constraints.

Our algorithms take an input parameter $\vec{\alpha}$ and output a corresponding fair solution, assuming that the given instance is feasible. Determining an appropriate $\vec{\alpha}$ is intentionally deferred to stakeholders and policymakers. The question of how to fairly allocate resources to groups is an ongoing topic of debate which we make no attempt to conclude (e.g. see [122]). Nevertheless, we view our work as progress toward formalizing and facilitating discussion of group-level fairness in kidney exchange [45] and other k-set packing applications. For example, the $\vec{\alpha}$ corresponding to an optimal packing has optimal Price of Fairness (PoF); i.e., there is no cost to the objective. On the other hand, other choices of $\vec{\alpha}$ may come at a great PoF. For a policymaker to advocate for proportional group fairness,

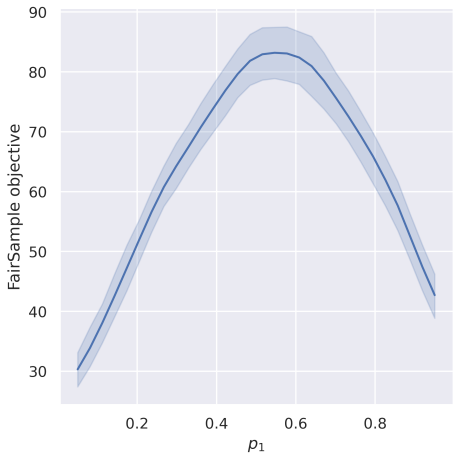


Figure 4.1: $\vec{\alpha} = [\alpha_1, \alpha_2]$ vs. packing objective achieved by Algorithm 5.

it is imperative they be able to articulate this trade-off. Our work is a step towards formalizing these necessary tools. Indeed, we explicitly view our work as descriptive, not prescriptive—we can map out a form of Pareto frontier (as in Figure 4.1) balancing the "tightness" imposed by the proportionality vector (x-axis) against the overall utility of the solution (y-axis). Then, a domain expert could use this for decision support when choosing the proper balance.

4.2.6 Main contributions

We study the problem of FAIRSP, a probabilistic fair variant of the k-set packing problem. We motivate the probabilistic fairness notions (RF and SRF as defined in (4.1) and (4.2) resp.) by first showing that the deterministic fair variant $\overline{\text{FAIRSP}}$ has unbounded integrality gap for the natural LP relaxation of the problem.

We provide two algorithms FAIRSAMPLE and FAIRELIMINATE that respectively guarantee RF (with $\gamma \approx 1$ w.h.p. and approximation factor $c \leq k + \epsilon_n$) and SRF (with $\gamma = \mathcal{O}(1)$ and $c = \mathcal{O}(\ell k^2)$ where ℓ is the number of colors). We say *sufficiently large packing size* to mean the fractional number of elements selected in the optimal LP (4.3) solution, i.e., $\sum_{i \in [m]} \sum_{g \in [\ell]} v_{i,g} y_i^*$, is sufficiently large.

Theorem 4.2.1. *For any instance of* FAIRSP, FAIRSAMPLE *is a randomized, polynomial-time algorithm with an approximation factor* $(k + \epsilon_k)$ *on the packing objective. Moreover, with probability* $1 - \epsilon_L$, FAIRSAMPLE *guarantees RF with* $\gamma = 1 + \epsilon_L$ *where* L *is a tunable parameter.*

Theorem 4.2.2. *For any instance of* FAIRSP *satisfying* $f = o(\sqrt{n})$ *and* $\alpha_g = \Theta(1/\ell)$ *for all* $g \in [\ell]$, FAIRELIMINATE *is a randomized, polynomial-time algorithm with an approximation factor* $\mathcal{O}(\ell k^2)$ *on the packing objective. Moreover, for instances with sufficiently large packing size,* FAIRELIMINATE *guarantees SRF with* $\gamma = \mathcal{O}(1)$.

Theorem 4.2.3. *For any instance of* FAIRSP *satisfying* $f = o(n/k^6\ell^4)$ *and* $\alpha_g = \Theta(1/\ell)$ *for all* $g \in [\ell]$, FAIRELIMINATE *is a randomized, polynomial-time algorithm with an approximation factor of* $\mathcal{O}(\ell k^2)$ *on the packing objective. Moreover, for instances with sufficiently large packing size,* FAIRELIMINATE *guarantees SRF with* $\gamma = \mathcal{O}(1)$.

Note the stronger fairness guarantees of FAIRELIMINATE holds for the family of FAIRSP instances satisfying: (i) the frequency of any element is bounded by $n^{1-o(1)}$, e.g., $f = o(n)$ for $k = \mathcal{O}(\log n)$ and $\ell = \mathcal{O}(\log n)$, (ii) the proportionality vector satisfies $\alpha_g = \Theta(1/\ell)$ for all g, and (iii) the packing size is sufficiently large.

The input parameter a to FAIRELIMINATE is carefully chosen to balance both γ and c. For the family of instances with constant number of colors c, and sufficiently large packing size we prove that $\gamma = \mathcal{O}(1)$ and $c = \mathcal{O}(k^2)$. For many practical real-world applications, like kidney-exchange markets, k is a very small number, say 2 or 3, thus resulting in a reasonable (constant) approximation factor for both FAIRSAMPLE and FAIRELIMINATE, along with the desired fairness guarantees.

Notice that k-set packing is a special case of FAIRSP with $\ell = 1$, hence the integrality gap of LP (4.3) is at least $k - 1 + 1/k$ [123]. Therefore, our approximation factor for FAIRSAMPLE almost

matches the lower bound and further guarantees RF with γ arbitrarily close to optimal i.e., 1. Whereas, FAIRSAMPLE only guarantees fairness in expectation—with no promises on the ex-post fairness of the allocation—FAIRELIMINATE guarantees the stronger SRF with $\gamma = \mathcal{O}(1)$ and approximation factor $c = \mathcal{O}(ck^2)$.

Finally, in Section 4.6, we support these theoretical results with experiments on synthetic datasets, including a realistic dataset drawn from a real-world kidney exchange.

4.3 Randomized algorithms

In this section we focus on randomized algorithms for FAIR-K-SETPACKING with γ-APPROXIMATEFAIR and γ-PROBABLYAPPROXFAIRguarantees. Our algorithms are based on rounding optimal Linear Program (LP) solutions. The LP formulation for FAIR-K-SETPACKING, found in LP (4.3), is the standard LP formulation of set packing with added fairness constraints. Due to space constraints, we omit the proofs from this section.

$$\max \quad \sum_{i\in[m]} w_i y_i \tag{4.3a}$$

$$\text{subject to} \quad \sum_{i:u_j\in S_i} y_i \leq 1, \qquad \forall j \in [n] \tag{4.3b}$$

$$\sum_{i\in[m]} v_{i,g} y_i = \alpha_g \sum_{g\in[\ell]} \sum_{i\in[m]} v_{i,g} y_i, \qquad \forall g \in [\ell] \tag{4.3c}$$

$$y_i \in [0,1], \qquad \forall i \in [m]. \tag{4.3d}$$

Theorem 4.3.1. *The optimal value of LP* (4.3) *is an upper bound on the objective of* FAIR-K-SETPACKING.

Proof. Consider an optimal randomized algorithm OPT[3] for FAIR-K-SETPACKING. For each $i \in [m]$, let y_i be the probability that S_i is packed by the randomized algorithm OPT. We can verify that the LP objective (4.3a) captures the exact value of OPT (i.e., the max expected weight of the packing). Therefore, to prove our claim it suffices to show that $\{y_i\}_{i\in[m]}$ is a feasible point of LP (4.3). For each $i \in [m]$, let Y_i denote the indicator random variable that set S_i is selected by OPT. OPT can be viewed as a distribution over all feasible deterministic algorithms, any realization $\{Y_i\}$ satifsies $\sum_{j\in S_i} Y_i \leq 1$ for each $j \in [n]$. Therefore, $\mathbb{E}[\sum_{j\in S_i} Y_i] \leq 1$ which satisfies (4.3b). Since OPT is optimal for FAIR-K-SETPACKING, for each color $c \in [\ell]$, $\{Y_i\}$ must satisfy proportional constraints in *expectation* i.e., $\mathbb{E}[N_g] = \alpha_g \mathbb{E}[N]$, so (4.3c) is satisfied. Lastly, since $\{y_i\}$ are probabilities, (4.3d) is satisfied. □

Lemma 4.3.2. *An instance of* FAIR-K-SETPACKING *is feasible if and only if its corresponding LP* (4.3) *has a non-zero feasible solution.*

By Lemma 4.3.2 we can efficiently verify whether the underlying FAIR-K-SETPACKING instance is feasible. Moreover, this implies our algorithms abort *only when* the underlying FAIR-K-SETPACKING instance is infeasible.

Algorithmic challenges and techniques:

As highlighted in the introduction, randomized algorithms can be substantially more powerful than deterministic ones on the objective studied here; however, the design and analysis of a randomized algorithm can be technically challenging mainly due to the craft involved in using randomness and analysis of such random variables. Several existing dependent rounding methods [117, 118, 119] for

[3]We use OPT for both the optimal algorithm and the optimal objective value interchangeably.

k-set packing are based on *randomized rounding* (which samples subsets based on an LP solution) followed by *suitable alterations* (which drops some sets to ensure a packing is returned) that lead to a feasible packing with good approximation guarantees. However, the packing returned by these algorithms is far from satisfying any fairness constraints. Thus the main challenge we address is designing packing algorithms guaranteeing γ-APPROXIMATEFAIR and γ-PROBABLYAPPROXFAIR.

Our primary contribution is the two algorithms, FAIRSAMPLE and FAIRELIMINATE, which are tailored to guarantee γ-APPROXIMATEFAIR and γ-PROBABLYAPPROXFAIR, respectively. Our algorithms utilize a randomized dependent rounding method, drawing inspiration from the works of [118] and [119], to optimize the packing objective function. Additionally, we modify existing algorithms to further provide probabilistic fairness guarantees (minimizing γ). Throughout the rest of the paper we use simulation to refer to Monte Carlo simulation.

1. We augment the rounding method of [119] with a simulation based method to tightly bound the probability of selecting any subset S_i, $i \in [m]$. Thus, guaranteeing γ-APPROXIMATEFAIR with near-optimal γ (i.e., $\gamma = 1 + \epsilon_L$[4]). (See Theorem 4.2.1)

2. Although FAIRSAMPLE provides a better approximation on the packing objective than FAIRELIMINATE, it is hard to provide a useful lowertail bound of N_g since the random variables $\{Z_i\}_{i\in[m]}$ are highly dependent. We use a much more intricate analysis on the simpler FAIRELIMINATE using concentration bounds from [124] which guarantee γ-PROBABLYAPPROXFAIR with $\gamma = \mathcal{O}(1)$ (a small constant), under reasonable assumptions on the input instances (See Theorems 4.2.2 and 4.2.3).

[4] L is the number of simulations in FAIRSAMPLE.

4.3.1 Theoretical guarantees of FAIRSAMPLE

We obtain FAIRSAMPLE, Algorithm 5, by augmenting the rounding algorithm from [119] with an additional simulation step (step 13). The purpose behind the simulation-based sampling is to "tightly" bound $\mathbb{E}[N_g]$ for each color g; the importance of this step is discussed in the "Intuition behind the analysis of FAIRSAMPLE". The probability of satisfying γ-APPROXIMATEFAIR with $\gamma = 1 + \epsilon_L$ depends on the number of simulations L. Nevertheless, L needs only be polynomial in the problem size for high-quality results. For more details on the simulation-based sampling method see the works of [81, 125].

Steps 1-4 FAIRSAMPLE solves the LP of the given FAIR-K-SETPACKING instance or aborts if the LP is infeasible. **Steps 7-10** Next, it runs randomized rounding on each variable using a carefully chosen function $f(y_i^*)$ on the optimal LP solution $\{y_i^*\}$. In the alteration step, the variables rounded to 1 are dropped based on a randomly ordered permutation generated in the previous step. These steps are repeated for L rounds. **Steps 13- 15** Finally, with the previous L simulations, we have estimated $q_i' \approx \Pr(S_i \in \mathcal{F}_2)$ for each $i \in [m]$. This ensures that $\Pr(S_i \in \mathcal{F}) \approx \frac{y_i^*}{k + \frac{k}{\exp k}}$, which is crucial to the analysis. Theorem 4.2.1 formalizes the approximation guarantees provided by FAIRSAMPLE for the objective and γ-APPROXIMATEFAIR fairness ratio.

Algorithm 5 FAIRSAMPLE

Input: Number of Monte Carlo simulations L.

1: Solve LP (4.3) to get an optimal fractional solution $\{y_i^*\}_{i\in[m]}$.
2: **if** LP (4.3) is infeasible **then**
3: **return** "Infeasible"
4: **end if**
5: For all $i \in [m]$, set $L_i = 0$.
6: **for** $j = 1, \ldots, L$ **do**
7: Construct $\mathcal{F}_1$ by sampling each set S_i with probability $f(y_i^*)$ where $f(x) = x\left(1 - \frac{x}{2}\right)$.
8: For each $S_i \in \mathcal{F}_1$, sample $x_i \sim U(0, 1)$.
9: Build the valid packing $\mathcal{F}_2$ as follows. In increasing order of x_i, pack S_i if it does not intersect a previously selected set.
10: For each $S_i \in \mathcal{F}_2$, increment L_i by 1.
11: **end for**
12: Keep the packing $\mathcal{F}_2$ from the last run of the above loop.
13: Set $q_i' = L_i/L$ for each $i \in [m]$.
14: Shrink $\mathcal{F}_2$ into $\mathcal{F}$ as follows. For each $S_i \in \mathcal{F}_2$, keep S_i with probability $\frac{y_i^*}{(q_i'+\epsilon)\left(k+\frac{k}{\exp(k)}\right)}$.
15: **return** The packing $\mathcal{F}$.

4.3.1.1 Intuition behind the analysis of FAIRSAMPLE

The first part of FAIRSAMPLE, steps 1-11, is similar to that of [119], thus leading to $k + \epsilon_k$ approximation on packing objective. The remaining steps of FAIRSAMPLE ensure a tight upper bound on $\mathbb{E}[N_g]$ for each color g. This is crucial to upper bounding $\mathbb{E}[N_g]/\mathbb{E}[N_h]$ for each pair of colors g and h, thus giving a suitable γ.

Theorem 4.2.1 demonstrates that FAIRSAMPLE has an approximation factor of $(k + \frac{k}{\exp(k)})$ for the packing objective, and it guarantees γ-APPROXIMATEFAIR with a $\gamma = 1 + \epsilon_L$, which can be made arbitrarily close to optimal ($\gamma = 1$) with more simulations. However, it should be noted that the fairness guarantees of FAIRSAMPLE only hold with high probability, as they depend on the simulation step. Nevertheless, the experimental results in Section 4.6 show FAIRSAMPLE consistently achieves $\gamma \approx 1$ even with only a few simulations.

It is important to remember that Theorem 4.2.1 only guarantees fairness in expectation—there are no guarantees on the ex-post fairness of the allocation. This shifts our attention to FAIRELIMINATE.

4.3.2 Theoretical guarantees of FAIRELIMINATE

FAIRELIMINATE, Algorithm 6, incorporates *randomized rounding* followed by *alterations*, similar to [118]. In the *randomized rounding* step, we select each S_i independently based on y_i^* and a. The main difference between FAIRELIMINATE and existing algorithms is allowing for an appropriate parameter a that results in a collection of sets with acceptable "expected packing weight" and "expected cardinality[5]" while limiting the number of sets selected in this step. Thereby, reducing the number of sets that are eventually dropped in the *alteration* step. In the *alteration* step, we ensure the collection of sets is a packing by discarding intersecting sets while still maintaining good expected weight and not losing too many elements of any color.

To guarantee γ-PROBABLYAPPROXFAIR, we need concentration bounds on the upper and lower tails of N_g. Our randomized rounding procedure lets us upper bound N_g as a sum of independent random variables, making it easy to apply Hoeffding bounds on the upper tail. However, the critical part of the analysis is establishing a *concentration on the lowerbound* of N_g. We solve this problem by a clever application of the *Deletion Method* by [124].

Steps 1-4 FAIRELIMINATE solves the LP of the given FAIR-K-SETPACKING instance or aborts if the LP is infeasible. **Steps 5-6** Next, the algorithm applies randomized rounding to each variable using a function $\frac{ay_i^*}{k}$ on the optimal LP solution for a small constant $a > 0$; let $\mathcal{F}$ denote the set of all subsets that are sampled in this step. **Steps 7-12** Then, in the alteration step, the algorithm removes any sets that intersect with other sets in $\mathcal{F}$, and returns the remaining sets as the final packing.

[5]Expected cardinality refers to the expected number of elements selected in this step i.e., $\mathbb{E}[\sum_{i\in[m]} |S_i|Y_i]$.

Theorems 4.2.2 and 4.2.3 state the approximation guarantees provided by FAIRELIMINATE on both the objective and γ-PROBABLYAPPROXFAIR.

Algorithm 6 FAIRELIMINATE

Input: $a < \alpha^* \frac{1-\delta}{6k}$ where $\delta = \frac{1}{n^{0.25}}$ and $\alpha^* = \min_{g \in [\ell]} \alpha_g$

Solve LP (4.3) to get an optimal fractional solution $\{y_i^*\}$
if LP (4.3) is infeasible **then**
  return return "Infeasible"
end if
For each $i \in [m]$ sample $Y_i \sim Ber(\frac{ay_i^*}{k})$ for a small constant $a > 0$
$\mathcal{F} = \{S_i \in \mathcal{S} \mid Y_i = 1\}$
$\mathcal{F}_1 \leftarrow \varnothing$
for S_i and S_j such that $S_i \cap S_j \neq \varnothing$ **do**
  if $Y_i = 1$ and $Y_j = 1$ **then**
    $\mathcal{F}_1 \leftarrow \mathcal{F}_1 \cup \{S_i, S_j\}$
  end if
end for
return $\mathcal{F} - \mathcal{F}_1$

Intuition behind the analysis of FAIRELIMINATE

The key idea behind FAIRELIMINATE is to choose an appropriate value for a[6] such that *randomized rounding* results in a set of variables $\{Y_i\}$ with an acceptable "expected weight". We then prove that the expected number of variables dropped in the alteration step is minimal, resulting in a high-quality approximation of the packing objective.

It's important to note that the $\{Z_i\}$ are dependent random variables with both negative and positive correlations between them. As a result, we cannot directly use the Chernoff-Hoeffding bounds that are applied to sums of negatively correlated random variables, as outlined in [108]. This creates a challenge in achieving concentration on the desired value of N_g. Instead we express Z_i as a function of the independent $\{Y_i\}$, allowing us to utilize various results on the polynomials of independent random

[6]The value of $a > 0$ is selected via the analysis of the algorithm.

variables such as [124, 126, 127].

We know that the random variable Z_i takes the value either 1 or 0 based on two events: (i) the random variable $Y_i = 1$ in the *randomized rounding* step, and (ii) none of its neighbors $j \in N(i)$[7] have $Y_j = 1$. Therefore, $Z_i = Y_i(1 - \max_{j \in N(i)} Y_j)$, allowing us to express N_g as follows:

$$N_g = \sum_{i \in [m]} v_{i,g}(Y_i - Y_i \max_{j \in N(i)} Y_j) \tag{4.4}$$

$$\geq \sum_{i \in [m]} v_{i,g}(Y_i - Y_i \sum_{j \in N(i)} Y_j) \tag{4.5}$$

$$\geq \underbrace{\sum_{i \in [m]} v_{i,g} Y_i}_{A_g} - 2k \underbrace{\sum_{\substack{1 \leq i < j \leq m \\ S_i \cap S_j \neq \varnothing}} Y_i Y_j}_{B} . \tag{4.6}$$

By applying the Hoeffding bound [128], we can show that $A_g = \sum_{i \in [m]} v_{i,g} Y_i$ is concentrated around its mean with probability p. Additionally, we can use the Deletion method by [124] to prove that B is also concentrated around its mean with probability q. The crucial step in applying the Deletion Method is the construction of a hypergraph on the random variables $\{Y_i\}$, which can be challenging. Finally, by a union bound and suitable p and q, we find concentration on the lower bound of $A_g - 2kB$.

Theorem 4.2.2 is a direct outcome of this approach. Theorem 4.2.3 is an enhancement of Theorem 4.2.2 resulting from a more refined analysis of the concentration of B in the Deletion Method. The complete justification for these theorems can be found in the full version of the paper. We present the improved result in Theorem 4.2.3 as our final conclusion.

[7] $N(i) = \{S_j : S_i \cap S_j \neq \varnothing\}$.

4.4 Instance feasibility for FAIR-K-SETPACKING

Lemma 4.3.2. *An instance of* FAIR-K-SETPACKING *is feasible if and only if its corresponding LP* (4.3) *has a non-zero feasible solution.*

Proof. The backward direction is clear from the existence of FAIRSAMPLE, Algorithm 6, since LP (4.3) is feasible, just take FAIRSAMPLE and instead of running simulations for q_i', set $q_i' = \Pr[E_i] = q_i$; recall E_i is the event that the set S_i is added to the packing $\mathcal{F}$ returned by FAIRSAMPLE.

For the other direction, note any randomized packing algorithm ALG with γ-APPROXIMATEFAIR guarantees defines random variables $\{Z_i\}_{i\in[m]}$ indicating whether S_i is packed by ALG. We claim $y_i = \Pr[Z_i = 1]$ for $i \in [m]$ is feasible to LP (4.3). We can see $\{y_i\}$ satisfies all constraints of LP (4.3) in the same way as in the proof of Theorem 4.3.1. □

4.5 Approximation algorithms for FAIR-K-SETPACKING

Lemma 4.5.1 ([119])**.** *Assume $f : [0,1] \to [0,1]$ and it has second derivatives over $[0,1]$. Then we have that $\ln(1 - tf(x))$ is a convex function for any given $t \in (0,1)$ if and only if $(1-f)(-f'') \geq f'^2$ for all $x \in [0,1]$.*

Theorem 4.2.1. *For any instance of* FAIRSP, FAIRSAMPLE *is a randomized, polynomial-time algorithm with an approximation factor $(k + \epsilon_k)$ on the packing objective. Moreover, with probability $1 - \epsilon_L$,* FAIRSAMPLE *guarantees RF with $\gamma = 1 + \epsilon_L$ where L is a tunable parameter.*

Proof. The first part of the analysis is similar to the hypergraph matching (Theorem 3) in [119]. For each set $S_i \in \mathcal{S}$, let E_i be the event that the set S_i is added to the feasible packing $\mathcal{F}$ in FAIRSAMPLE. First, we perform *randomized rounding* to obtain $\mathcal{F}_1$ by sampling each S_i with probability $f(y_i^*)$. Then,

for each $S_i \in \mathcal{F}_1$ generate a random sample $x_i \in [0,1]$ independently and uniformly at random. This induces a permutation π on the sampled sets $\mathcal{F}_1$ given by ordering x_i in ascending order. Now fix i and for each element $u_j \in S_i$ let L_j be the event that u_j is not already packed when considering S_i in the permutation π.

$$\Pr[E_i] = f(y_i^*) \int_0^1 \Pr\Big[\bigwedge_{j:u_j \in S_i} L_j \mid x_i = t \Big] \, dt \tag{4.7}$$

$$= f(y_i^*) \int_0^1 \prod_{u_j \in S_i} \prod_{r:u_j \in S_r} \Pr\Big[(x_r > x_i) \vee S_r \notin \mathcal{F}_1 \mid x_i = t \Big] \, dt \tag{4.8}$$

$$= f(y_i^*) \int_0^1 \prod_{u_j \in S_i} \prod_{r:u_j \in S_r} \Big(1 - t f(y_r^*) \Big) \Big) \, dt \tag{4.9}$$

$$= f(y_i^*) \int_0^1 \prod_{(S_i \cap S_r) \neq \varnothing} \exp\Big(\ln\big(1 - t f(y_r^*)\big) \Big) \, dt \tag{4.10}$$

$$= f(y_i^*) \int_0^1 \exp\Big(\sum_{(S_i \cap S_r) \neq \varnothing} \ln\big(1 - t f(y_r^*)\big) \Big) \, dt. \tag{4.11}$$

Thus we would like to minimize $\exp\Big(\sum_{(S_i \cap S_r) \neq \varnothing} \ln\big(1 - t f(y_r^*)\big) \Big)$ to get a valid lower bound on the probability of picking each set S_i. Let $g(y) = y\big(1 - \frac{y}{2}\big)$. We know g is convex, $g(0) = 0$, $g'(0) > 0$, and $g'' = 1$. By Lemma 4.5.1, $\ln(1 - t g(y))$ is also convex for $t \in (0,1)$. Additionally, we know $\sum_{S_i \cap S_j \neq \varnothing} y_j \leq k(1 - y_i^*)$. Therefore, to minimize $\Big(\sum_{(S_i \cap S_r) \neq \varnothing} \ln\big(1 - t f(y_r^*)\big) \Big)$, the worst case scenario is $y_r = \epsilon (\to 0)$ for all r such that $S_r \cap S_i \neq \varnothing$. Hence,

$$\min \exp\left(\sum_{(S_i \cap S_r) \neq \varnothing} \ln\left(1 - t f(y_r^*)\right) \right) = \lim_{\epsilon \to 0} \big(1 - t f(\epsilon)\big)^{k(1 - y_i^*)/\epsilon}. \tag{4.12}$$

Let $z = \lim_{\epsilon \to 0} \left(1 - tf(\epsilon)\right)^{\frac{k(1-y_i^*)}{\epsilon}}$. By taking $\ln$ on both sides and using L'Hôpital's rule,

$$\ln z = \lim_{\epsilon \to 0} \frac{k(1-y_i^*)}{\epsilon} \ln\left(1 - tf(\epsilon)\right) \tag{4.13}$$

$$= \lim_{\epsilon \to 0} -t\frac{k(1-y_i^*)f'(\epsilon)}{\left(1 - tf(\epsilon)\right)}. \tag{4.14}$$

And so, substituting $f(0) = 0$ and $f'(0) = 1$,

$$z = \exp\left(-tk(1-y_i^*)\right). \tag{4.15}$$

Therefore, we have

$$\Pr[E_i] \geq f(y_i^*) \int_0^1 \exp\left(-tk(1-y_i^*)\right) dt \tag{4.16}$$

$$= f(y_i^*)\big) \int_0^1 \exp\left(-tk(1-y_i^*)\right) dt \tag{4.17}$$

$$\geq f(y_i^*) \frac{1 - \exp(-k(1-y_i^*))}{k(1-y_i^*)} \geq \frac{y_i^*}{k + \frac{k}{\exp(k)}}. \tag{4.18}$$

Using Monte Carlo simulations, we can get estimates for $\Pr[E_i]$. Let q_i and q_i' denote the true and estimated values of $\Pr[E_i]$ such that $\Pr[|q_i - q_i'| \leq \epsilon_L] > 1 - \epsilon_L$ where L is the number of simulations, which is a parameter to FAIRSAMPLE. Let n_g denote the fractional number of elements of color ℓ in the LP solution, i.e., $n_g = \sum_{i \in [m]} v_{i,g} y_i^*$, and let N_c be the number of elements of color c in the final

packing. Let Y_i denote the indicator random variable that set S_i is in the final packing $\mathcal{F}$. Then,

$$\mathbb{E}[N_g] = \sum_{i\in[m]} v_{i,g}\mathbb{E}[Y_i] \tag{4.19}$$

$$= \sum_{i\in[m]} v_{i,g} \Pr[S_i \in \mathcal{F}_2] \Pr[S_i \in \mathcal{F} \mid S_i \in \mathcal{F}_2] \tag{4.20}$$

$$= \sum_{i\in[m]} v_{i,g} q_i \left(\frac{\frac{y_i^*}{k+\frac{k}{\exp(k)}}}{q_i{}' + \epsilon} \right) \tag{4.21}$$

$$\leq \frac{1}{k + \frac{k}{\exp(k)}} n_c. \tag{4.22}$$

Inequality (4.22) follows because $q_i \leq q_i' + \epsilon_L$ with high probability. Now, from the final sampling step we also have,

$$\mathbb{E}[N_g] = \sum_{i\in[m]} v_{i,g}\mathbb{E}[Y_i] \tag{4.23}$$

$$= \sum_{i\in[m]} v_{i,g} \Pr[S_i \in \mathcal{F}_2] \Pr[S_i \in \mathcal{F} \mid S_i \in \mathcal{F}_2] \tag{4.24}$$

$$= \sum_{i\in[m]} v_{i,g} q_i \left(\frac{\frac{y_i^*}{k+\frac{k}{\exp(k)}}}{q_i{}' + \epsilon_L} \right) \tag{4.25}$$

$$\geq \sum_{i\in[m]} v_{i,g} q_i \left(\frac{\frac{y_i^*}{k+\frac{k}{\exp(k)}}}{q_i + \epsilon_L + \epsilon_L} \right) \tag{4.26}$$

$$= \sum_{i\in[m]} v_{i,g} \left(\frac{\frac{y_i^*}{k+\frac{k}{\exp(k)}}}{1 + 2\epsilon_L/q_i} \right) = \frac{n_g}{(k + \frac{k}{\exp(k)})(1 + 2\epsilon_L/p)}. \tag{4.27}$$

We choose a sufficiently large L such that $\epsilon_L/p \to 0$. From (4.22) and (4.27), we get, for all colors $g, h \in [\ell]$,

$$\frac{\mathbb{E}[N_g]}{\mathbb{E}[N_h]} \leq \left(\frac{1}{1 + 2\epsilon_L/p}\right)\frac{n_g}{n_h} = (1 + \epsilon_L)\frac{\alpha_g}{\alpha_h}. \tag{4.28}$$

Therefore, $\gamma = 1 + \epsilon_L$ where $\frac{\epsilon_L}{p} \to 0$. However, it is to be noted that this occurs only with high probability since the Monte Carlo simulation used to estimate the values $\Pr[E_i]$ is reliable with high probability i.e., probability of at least $1 - \epsilon_L$. □

FAIRELIMINATE

Let N_g be the number of elements of color g selected by FAIRELIMINATE, Algorithm 6. Our goal is to prove upper and lower concentration on N_g in order to show that there exists an algorithm for the FAIR-K-SETPACKING that guarantees γ-PROBABLYAPPROXFAIR with some small $\gamma \geq 1$.

Let Z_i be the indicator that set S_i is selected by FAIRELIMINATE, then $N_g = \sum_{i \in [m]} v_{i,g} Z_i$. Lemmas 4.5.2 and 4.5.3 together provide approximation guarantees on the objective of FAIR-K-SETPACKING. Let Y_i, $i \in [m]$, be the indicator random variables for the event that S_i is selected in the *randomized rounding* step of FAIRELIMINATE. Then probability that S_i is selected is given by $\mathbb{E}[Y_i]$. Lemma 4.5.2 gives an upper and lower bound on the probability that S_i on $\Pr[Z_i = 1]$ or $\mathbb{E}[Z_i]$.

Lemma 4.5.2. *For each $i \in [m]$ the probability that the set S_i is selected by* FAIRELIMINATE *lies within a constant factor of the probability that S_i is selected by randomized rounding step i.e.,*

$$(1 - a)\mathbb{E}[Y_i] \leq \mathbb{E}[Z_i] \leq \mathbb{E}[Y_i] \tag{4.29}$$

Proof. The upper bound on $\mathbb{E}[Z_i]$ directly follows from the fact that the expected value of Z_i is bounded by the expected value of the intermediate variables Y_i from the randomized rounding step. Therefore, for each $i \in [m]$,

$$\mathbb{E}[Z_i] \leq \mathbb{E}[Y_i].$$

To obtain the lower bound, we know that the probability that $Z_i = 1$ is given by

$$\begin{aligned}\Pr[Z_i = 1] &= \Pr[Y_i = 1, \bigwedge_{j \in N(i)} (Y_j = 0)] \\ &= \frac{ay_i^*}{k}\Big(1 - \sum_{j \in N(i)} \Pr[Y_j = 1]\Big) \\ &\geq \frac{ay_i^*}{k}\Big(1 - \sum_{j \in N(i)} \frac{ay_j^*}{k}\Big).\end{aligned}$$

Since $\{y_i^*\}$ are the LP optimal variables and defining $C(u) = \{i : u \in S_i, i \in [m]\}$, we know that

$$\sum_{j \in N(i)} y_j^* = \sum_{u \in S_i} \sum_{j \in C(u), j \neq i} y_j^* \leq k \max_{u \in S_i} \sum_{j \in C(u)} y_j^* \leq k.$$

By using the inequality above,

$$\Pr[Z_i = 1] \geq \frac{ay_i^*}{k}\Big(1 - \sum_{j \in N(i)} \frac{ay_j^*}{k}\Big) \geq \frac{ay_i^*}{k}\Big(1 - \frac{a}{k}k\Big) = (1-a)\frac{ay_i^*}{k} = (1-a)\mathbb{E}[Y_i].$$

Therefore, we establish the upper and lower bounds on $\Pr[Z_i = 1]$. □

Lemma 4.5.3. FAIRELIMINATE *returns a packing that is at least* $\big(\frac{k}{a(1-a)}\big)-$*approximate for the* FAIR-K-SETPACKING *problem.*

Proof. We know that the expected weight of the packing returned by FAIRELIMINATE is given by

$$\mathbb{E}\Big[\sum_{i \in [m]} w_i Z_i\Big] = \sum_{i \in [m]} w_i \mathbb{E}[Z_i] \geq \sum_{i \in [m]} w_i (1-a)\mathbb{E}[Y_i].$$

However, we know that from Theorem 4.3.1 that the optimal LP objective $\sum_{i \in [m]} w_i y_i^*$ is a valid upper bound for the maximum packing of any feasible FAIR-K-SETPACKING instance i.e., OPT $\leq$

$\sum_{i\in[m]} w_i y_i^*$. Therefore, the approximation factor is

$$\frac{\text{OPT}}{\mathbb{E}[\text{ALG}]} \leq \frac{\sum_{i\in[m]} w_i y_i^*}{\sum_{i\in[m]} w_i(1-a)\mathbb{E}[Y_i]} \leq \frac{k}{a(1-a)}.$$

□

The below corollary follows from the fact that $a < \alpha^* \frac{1-\delta}{6k}$, and we know by definition that $\alpha^* = \Theta(1/\ell)$. Therefore, $a(1-a) = a - a^2 \geq \frac{1}{6\ell k}$, leading to an approximation factor $\mathcal{O}(\ell k^2)$.

Corollary 4.5.4. *The approximation factor for* FAIRELIMINATE *is atmost* $\mathcal{O}(\ell k^2)$.

Note that the variables Z_i are dependent random variables with both positive and negative correlations. Hence we can't directly apply the Chernoff-Hoeffding bounds. However the variables Y_i's are independent random variables and therefore, we express Z_i's in terms of $\{Y_1, Y_2, \ldots, Y_m\}$. Let $N(i)$ denote the indices of the set of the neighbors of S_i i.e., $N(i)$ is $\{j : S_i \cap S_j \neq \varnothing\}$.

$$N_g = \sum_{i\in[m]} v_{i,g} Z_i \tag{4.30}$$

$$= \sum_{i\in[m]} (Y_i - Y_i \max_{j\in N(i)} Y_j) \tag{4.31}$$

$$\geq \sum_{i\in[m]} v_{i,g}(Y_i - Y_i \sum_{j\in N(i)} Y_j) \tag{4.32}$$

$$\geq \sum_{i\in[m]} v_{i,g} Y_i - 2k \sum_{\substack{1\leq i<j\leq m \\ S_i\cap S_j\neq\varnothing}} Y_i Y_j. \tag{4.33}$$

Let $A_g = \sum_{i\in[m]} v_{i,g} Y_i$ and $B = \sum_{i<j:S_i\cap S_j\neq\varnothing} Y_i Y_j$ where $N \geq A_g - 2kB$.

Concentration on A_g

Given a random variable $Z \in [a, b]$ is sub-gaussian with variance proxy $\frac{(b-a)^2}{4}$. It follows that the sum of n independent bounded random variables $X_1, X_2, \ldots, X_n$ with $X_i \in [a_i, b_i]$ is a sub-gaussian with variance proxy $\sum_{i\in[n]} \frac{(b_i-a_i)^2}{4}$. [128] proved the following result for concentration of such bounded random variables.

Theorem 4.5.5. *[129] Given bounded random variables $X_1, X_2, \ldots, X_n$ where $X_i \in [a_i, b_i]$, for any $t \in \mathbb{R}$ and $\epsilon > 0$*

$$\Pr\Big[\Big|\sum_{i\in[n]} X_i - \mathbb{E}[\sum_{i\in[n]} X_i]\Big| \geq \epsilon\Big] \leq 2\exp\Big(-\frac{\epsilon^2}{\sum_{i\in[n]}(b_i-a_i)^2/4}\Big). \tag{4.34}$$

Theorem 4.5.6. *If optimal fractional solution OPT_{LP} is sufficiently large i.e., $\sum_{i\in[m]} v_{i,g}y_i^* = \Omega(\sqrt{n}\log n)$, then for any $\delta > 0$ and a small constant $C < \frac{2\alpha_g a\delta}{k^{1.5}}$,*

$$\Pr[A_g < (1-\delta)\alpha_g \frac{a}{k} OPT_{LP}] \leq \frac{1}{n^\ell}. \tag{4.35}$$

Proof. By definition of A_g, we have

$$\mathbb{E}[A_g] = \alpha_g \sum_{i\in[m]} |S_i|\mathbb{E}[Y_i] = \alpha_g \frac{a}{k}\mathrm{OPT}_{\mathrm{LP}} \geq \alpha_g \frac{a\sqrt{n}\log n}{k}. \tag{4.36}$$

Additionally, we also have

$$\sum_{i\in[m]} v_{i,g}^2 \leq k\sum_{i\in[m]} v_{i,g} \leq kn. \tag{4.37}$$

By setting $\epsilon = \frac{\alpha_g a \delta \text{OPT}_{\text{LP}}}{k}$ and applying inequalities (4.36) and (4.37) to Theorem 4.5.5 we get

$$\Pr[A_g \leq (1-\delta)\frac{\alpha_g a}{k}\text{OPT}_{\text{LP}}] \leq \exp(-4\frac{\alpha_g^2 a^2 n \log^2 n}{k^2 \times kn}) = \exp(-4\frac{\alpha_g^2 a^2 \delta^2 \log^2 n}{k^3}) \leq \frac{1}{n^\ell}.$$

We note $\ell < \frac{2\alpha_g a \delta}{k^{1.5}}$ is a positive constant. δ is quality parameter of our algorithm FAIRELIMINATE that determines the tail probabilities of A_g. Note that a higher ℓ value gives us a tight bound. □

Corollary 4.5.7. *If the optimal fractional solution OPT$_{LP}$ is sufficiently large i.e., $\sum_{i\in[m]} v_{i,g} y_i^* = \Omega(\sqrt{n}\log n)$, then $\Pr[A_g \geq (1+\delta)\alpha_g \frac{a}{k}OPT_{LP}] \leq \frac{1}{n^\ell}$ where $\ell < \frac{2\alpha_g a\delta}{k^{1.5}}$ is a small constant.*

Corollary 4.5.7 follows from a similar analysis as the proof of Theorem 4.5.6.

Deletion Method for Concentration on the Upper tail of B :

Let $\mathcal{H} = (\mathcal{I}, \mathcal{U})$ is the hypergraph such that $|\mathcal{I}| = m$ and $|\mathcal{U}| = n$ and $|I| \leq k$ for each $I \in \mathcal{I}$. Let X_I, $I \in \mathcal{I}$ denote a family of non-negative random variables. There is a local-dependency graph on $\mathcal{I}$, i.e., $X_I \sim X_J$ if $I \cap J \neq \varnothing$ for any $I, J \in \mathcal{I}$.

Theorem 4.5.8. *$\mathcal{H}$ be a family of subsets of $[n]$ of size at most k for some $k \leq n$ and let $(X_I, I \in \mathcal{I})$ be a family of non-negative random variables such that each X_I is independent of $(X_J | I \cap J = \varnothing)$. Let $Z := \sum_{I\in\mathcal{I}} X_I$ and $\mu := \mathbb{E}[Z] = \sum_I \mathbb{E}[X_I]$. Further, for $I \subseteq [n]$, let $Z_I := \sum_{I\subseteq J} X_J$ and let $Z_1^* := \max_u Z_{\{u\}}$. If $t > 0$, then for every real $r > 0$,*

$$\begin{aligned}\Pr[Z \geq \mu + t] &\leq (1+t/\mu)^{-r/2} + \Pr\left[Z_1^* > \frac{t}{2kr}\right] \\ &\leq (1+t/\mu)^{-r/2} + \sum_u \Pr\left[Z_{\{u\}} > \frac{t}{2kr}\right].\end{aligned}$$

We construct the hypergraph $\mathcal{H} = (\mathcal{U}, \mathcal{I})$ as following to apply the Deletion Method for finding the upper tail bound on polynomial $B = \sum_{Y_i} \sum_{j \in N(i)} Y_i Y_j$.

Construction of the Hypergraph for Deletion Method

We construct a hypergraph as follows: Let $\mathcal{U} = \{1, 2, \ldots, m\}$ where $\mathcal{I} = \{\{i, j\} : S_i \cap S_j \neq \varnothing\}$. We define the random variables $X_I = Y_i Y_j$ where $I = \{i, j\}$. Note that $k = 2$ since $|I| \leq 2, I \in \mathcal{I}$. Therefore, we define the element of study Z as follows:

$$Z = \sum_{I \in \mathcal{I}} X_I = \sum_{i,j : S_i \cap S_j \neq \varnothing} Y_i Y_j = \sum_{i \in [m]} Y_i \sum_{j \in N(i)} Y_j.$$

Therefore, we define the element of study for our algorithm is $B := \sum_{i \in [m]} \sum_{j \in N(i)} Y_i Y_j$ with $k = 2$. Subsequently,

$$B_1^* = \max_{i \in [m]} Y_i \sum_{j \in N(i)} Y_j \tag{4.38}$$

$$\leq \max_{i \in [m]} \sum_{j \in N(i)} Y_j \tag{4.39}$$

where Y_j's are independent Bernoulli random variables with mean $\frac{a y_i^*}{k}$ for each $j \in [m]$. And our goal is to find an estimate on $\Pr[B_1^* > \frac{t}{4r}]$.

Lemma 4.5.9. $\mathbb{E}[B] \leq \frac{a^2}{k} OPT_{LP}$.

Proof. By definition of B and taking expectation on both sides, we get the following:

$$\mathbb{E}[B] = \sum_{i\in[m]} \sum_{j\in N(i)} \mathbb{E}[Y_i]\mathbb{E}[Y_j] \tag{4.40}$$

$$= \sum_{i\in[m]} \sum_{j\in N(i)} \frac{a^2}{k^2} y_i^* y_j^* \tag{4.41}$$

$$= \sum_{i\in[m]} \frac{a^2}{k^2} y_i^* \sum_{j\in N(i)} y_j^* \tag{4.42}$$

$$\leq \sum_{i\in[m]} \frac{a^2}{k^2} y_i^* \times k \tag{4.43}$$

$$\leq \frac{a^2}{k} \text{OPT}_{\text{LP}}. \tag{4.44}$$

Inequality (4.44) follows from the fact that $|S_i| \leq k$ and $\sum_{i:u\in S_i} y_i^* \leq 1$ for any $u \in \mathcal{U}$. □

Lemma 4.5.10. *Suppose that* $B_1^* = \max_{i\in[m]} \sum_{j\in N(i)} Y_j$ *and* k *and* f *are* FAIR-K-SETPACKING *parameters. Then, the following statements are true.*

1. $B_1^* \leq kf \max_{i\in[m]} Y_i$.
2. $B_1^* \leq k \max_{u\in\mathcal{U}} \sum_{j\in C(u)} Y_j$ *where* $C(u) = \{j : u \in S_j\}$ *where* $|C(u)| \leq f$.
3. $\Pr[B_1^* > \frac{t}{4r}] \leq \Pr[\max_{i\in[m]} Y_i > \frac{t}{4kfr}]$ *and* $\Pr[B_1^* > \frac{t}{4r}] \leq \Pr[\max_{u\in\mathcal{U}} \sum_{j\in C(u)} Y_j > \frac{t}{4kr}]$.

Proof. We know that $\max_{i\in[m]} \sum_{j\in N(i)} Y_j$ has at most $f \times k$ variables where f is the maximum frequency of any element and k is the maximum number of elements in any subset. Therefore,

$$B_1^* \leq k \times f \max_{i\in[m]} Y_i. \tag{4.45}$$

and inequality (4.45) establishes bound on the first item (1). We know that for each $i \in [m]$, since

$|S_i| \leq k$ we get

$$\sum_{j \in N(i)} Y_j \leq k \max_{u \in S_i} \sum_{j \in C(u)} Y_j. \tag{4.46}$$

Thus we establish the second item (2) using inequality (4.46) as follows:

$$\max_{i \in [m]} \sum_{j \in N(i)} Y_j \leq k \max_{i \in [m]} \max_{u \in S_i} \sum_{j \in C(u)} Y_j = k \max_{u \in \mathcal{U}} \sum_{j \in C(u)} Y_j. \tag{4.47}$$

Finally, (3) follows directly from inequalities (4.45) and (4.47). □

Lemma 4.5.11. $\Pr[B_1^* > \frac{t}{4r}] \leq \frac{4krf}{t} \mathbb{E}\big[\max_{i \in [m]} Y_i\big]$

Proof. Using Lemma 4.5.10 followed by Markov's inequality, we get

$$\begin{aligned} \Pr[B_1^* > \frac{t}{4r}] &\leq \Pr\Big[\max_{i \in [m]} Y_i > \frac{t}{4fkr}\Big] \\ &\leq \frac{4krf}{t} \mathbb{E}\big[\max_i Y_i\big]. \end{aligned}$$

□

Lemma 4.5.12. $\mathbb{E}\big[\max_i Y_i\big] \leq e\left(\frac{a}{k}\right)^{\frac{1}{\log m}}$ *for any* $n > e$ *where* e *denotes the base of natural logarithm.*

Proof. By using the Norm Equivalence inequality: for any $p > r > 0$ and $x \in \mathbb{R}^d$, $\|x\|_p \leq \|x\|_r \leq d^{(1/r - 1/p)} \|x\|_p$. Therefore, taking $p = \infty$ and $r = \log m$ for the vector $[Y_1, Y_2, \ldots, Y_m]$, we get the following

$$\begin{aligned} \mathbb{E}[\max_{i \in [m]} Y_i] &\leq \mathbb{E}\Big[\Big(\sum_{i \in [m]} Y_i^{\log m}\Big)^{1/\log m}\Big] \\ &\leq \mathbb{E}\Big[\Big(\sum_{i \in [m]} Y_i\Big)^{1/\log m}\Big] \end{aligned}$$

Since $a^{1/\log(x)}$ is a concave function, using Jensen's inequality we get the following,

$$\begin{aligned}
\mathbb{E}[\max_{i\in[m]} Y_i] &\leq \Big(\mathbb{E}\Big[\sum_{i\in[m]} Y_i\Big]\Big)^{\frac{1}{\log m}} \\
&\leq \Big(\frac{a}{k}\sum_{i\in[m]} y_i^*\Big)^{\frac{1}{\log m}} \\
&\leq (\text{OPT}_{\text{LP}})^{\frac{1}{\log m}}\Big(\frac{a}{k}\Big)^{\frac{1}{\log m}} \\
&\leq n^{\frac{1}{\log m}}\Big(\frac{a}{k}\Big)^{\frac{1}{\log m}} \\
&\leq n^{\frac{1}{\log n}}\Big(\frac{a}{k}\Big)^{\frac{1}{\log m}} \\
&= e\Big(\frac{a}{k}\Big)^{\frac{1}{\log m}}.
\end{aligned}$$

The last three inequalities follow from the facts that $\sum_{i\in[m]} y_i^* \leq \text{OPT}_{\text{LP}} \leq n$ and $m \geq n$. □

Corollary 4.5.13. $\Pr\Big[B_1^* > \frac{t}{4r}\Big] \leq \frac{4krfe}{t}\Big(\frac{a}{k}\Big)^{\frac{1}{\log n}}$.

Proof. Now we finally prove an upper bound on the probability $\Pr[B_1^* > \frac{t}{4r}]$ by using Lemmas 4.5.11 and 4.5.12. Thus, we get

$$\begin{aligned}
\Pr[B_1^* > \frac{t}{4r}] &\leq \frac{4krfe}{t}\mathbb{E}\big[\max_i Y_i\big] \\
&\leq \frac{4krfe}{t}\Big(\frac{a}{k}\Big)^{\frac{1}{\log m}}.
\end{aligned}$$

□

Theorem 4.5.14. $\Pr[B > 2\big(\frac{a^2}{k}\big)OPT_{LP}] \leq \frac{1}{n} + \frac{4k^3 f}{3a^2\sqrt{n}}$.

Proof. From Theorem 4.5.8, we know that

$$\Pr[B \geq \mathbb{E}[B] + t] \leq \left(1 + \frac{t}{\mathbb{E}[B]}\right)^{-r/2} + \Pr\left[B_1^* > \frac{t}{4r}\right]$$
$$\Pr[B \geq \frac{a^2}{k}\mathbf{OPT}_{\text{LP}} + t] \leq \left(1 + \frac{t}{\mathbb{E}[B]}\right)^{-r/2} + \frac{4krfe}{t}\left(\frac{a}{k}\right)^{\frac{1}{\log n}}.$$

Now letting $t = \frac{a^2}{k}\mathbf{OPT}_{\text{LP}}$ and $r = \log n$, we get the following estimate since $t > \mathbb{E}[B]$ and $\mathbf{OPT}_{\text{LP}} \geq \sqrt{n}\log n$.

$$\begin{aligned}
\Pr[B > (\frac{a^2}{k} + \frac{a^2}{k})\mathbf{OPT}_{\text{LP}}] &\leq \exp(-\frac{r}{3}) + \frac{4krfe}{t}\left(\frac{a}{k}\right)^{\frac{1}{\log m}} \\
&\leq \exp(-r/3) + \frac{4k^2rfe}{a^2\mathbf{OPT}_{\text{LP}}}\left(\frac{a}{k}\right)^{\frac{1}{\log m}} \\
&\leq \frac{1}{n^{1/3}} + \frac{4k^2fe}{a^2\sqrt{n}}\left(\frac{a}{k}\right)^{\frac{1}{\log m}} \\
&\leq \frac{1}{n^{1/3}} + \frac{4k^2fe}{a^2\sqrt{n}}\left(\frac{a}{k}\right)^{\frac{1}{\log m}}.
\end{aligned}$$

The last three inequalities follow from the following facts:

$$(1 + \frac{t}{\mu_B})^{-r/2} \leq \exp(-\frac{r}{3})$$

if $t > \mu_B$, $\mathbf{OPT}_{\text{LP}} \geq \sqrt{n}\log n$, and $e(\frac{a}{k})^{\frac{1}{\log m}} \leq 1$. □

However, note that we can obtain a tighter bound on the value $(\frac{a}{k})^{\frac{1}{\log m}}$ by using basic properties of the logarithms.

$$\left(\frac{a}{k}\right)^{\frac{1}{\log m}} = \left(m^{\log_m \frac{a}{k}}\right)^{1/\log m} \leq \left(m^{\log_m \frac{a}{k}}\right)^{1/\log m} \tag{4.48}$$

$$= m^{-\log \frac{a}{k}} = \frac{1}{m^{\frac{a}{k}}} \tag{4.49}$$

$$\leq \frac{1}{n^{\frac{a}{k}}}. \tag{4.50}$$

Inequalities (4.49) and (4.50) are due to the facts that $\frac{a}{k} \leq 1$, $-\log m \leq \frac{1}{\log m}$ (since $m \geq n > e$) and $m \geq n$ respectively. Therefore, we can improve the lowerbound on the value of f from $o(\sqrt{n})$ to $o(n^{0.5+\frac{a}{k}})$. For the sake of brevity, we just say, $f = o(\sqrt{n})$.

Theorem 4.5.15. *Assuming that* $f = o(\sqrt{n})$ *and* $a < \frac{(1-\delta)\alpha_g}{4k}$,

$$\Pr[N_g \geq (\frac{\alpha_g^2}{2} - 4ak)\frac{a}{k}\mathit{OPT}_{\mathit{LP}}] \geq 1 - \epsilon_n.$$

Proof. Using the fact $N_g \geq A_g - 2kB$, followed by applying a union bound on Theorems 4.5.6 and 4.5.14, we get

$$\Pr\left[N_g \leq (1-\delta)\alpha_g \frac{a}{k}\text{OPT}_{\text{LP}} - \frac{4ka^2}{k}\text{OPT}_{\text{LP}}\right] \leq \frac{1}{n^{1/3}} + \frac{4k^2 f}{a^2\sqrt{n}} + \frac{1}{n^{\ell}}.$$

More simply,

$$\Pr\left[N_g \leq ((1-\delta)\alpha_g - 4ak)\frac{a}{k}\text{OPT}_{\text{LP}}\right] \leq \epsilon_n. \tag{4.51}$$

□

Theorem 4.5.16. *For any feasible instance of* FAIR-K-SETPACKING, *if* $\mathit{OPT}_{\mathit{LP}} \geq \sqrt{n}\log n$, $f =$

$o(\sqrt{n})$, $\delta > 0$, *and* $a < \frac{(1-\delta)\alpha_g}{4k}$, *then*

$$\Pr\left[\left((1-\delta)\alpha_g - 4ak\right)\frac{a}{k}\mathit{OPT}_{LP} \leq N_g \leq (1+\delta)\alpha_g\frac{a}{k}\mathit{OPT}_{LP}\right] \geq 1 - \epsilon_n.$$

Proof. This result is a direct consequence of Theorems 4.5.15 and 4.5.7 combined with a union bound. □

Theorem 4.2.2. *For any instance of* FAIRSP *satisfying* $f = o(\sqrt{n})$ *and* $\alpha_g = \Theta(1/\ell)$ *for all* $g \in [\ell]$, FAIRELIMINATE *is a randomized, polynomial-time algorithm with an approximation factor* $\mathcal{O}(\ell k^2)$ *on the packing objective. Moreover, for instances with sufficiently large packing size,* FAIRELIMINATE *guarantees SRF with* $\gamma = \mathcal{O}(1)$.

Proof. The theorem follows from Theorem 4.5.16. □

We further get improved estimates on $\Pr[B_1^* > \frac{t}{4r}]$. Definition 4.5.17 and Theorem 4.5.18 can be found in [129].

Definition 4.5.17 (Sub-Gaussian Random Variables)**.** A random variable $X \in \mathbb{R}$ is said to be sub-Gaussian with variance proxy σ^2 if $\mathbb{E}[X] = 0$ and its moment generating function satisfies

$$\mathbb{E}[\exp(sX)] \leq \exp\left(\frac{s^2\sigma^2}{2}\right). \tag{4.52}$$

Theorem 4.5.18. *[129] Let* $X_1, X_2, \cdots, X_n$ *be* n *independent random variables such that* $X_i \sim subG(\sigma^2)$. *Then for any* $a \in \mathbb{R}^n$ *we have*

$$\Pr\left[\sum_{i\in[n]} a_i X_i > t\right] \leq \exp\left(-\frac{t^2}{2\sigma^2|a|_2^2}\right), \tag{4.53}$$

$$\Pr\Big[\sum_{i\in[n]} a_i X_i < -t\Big] \leq \exp\Big(-\frac{t^2}{2\sigma^2|a|_2^2}\Big). \tag{4.54}$$

Theorem 4.5.18 can be extended to independent random variables with non-zero mean. We know that $Y_1, Y_2, \ldots, Y_m$ are independent Bernoulli random variables with mean $\frac{ay_i^*}{k}$ for each $i \in [m]$.

Lemma 4.5.19. *Let X be any real-valued random variable such that $a \leq X \leq b$ almost surely, i.e. with probability one. Then, for all $\lambda \in \mathbb{R}$,*

$$\mathbb{E}\left[e^{\lambda(X-\mathbb{E}[X])}\right] \leq \exp\Big(\frac{\lambda^2(b-a)^2}{8}\Big). \tag{4.55}$$

From Hoeffding's Lemma stated above we can conclude that every Bernoulli random variable is a sub-Gaussian with variance proxy $\sigma^2 = \frac{1}{4}$.

Fact 4.5.1. *Any Bernoulli variable X is sub-Gaussian with variance factor $\sigma^2 = 1/4$.*

Lemma 4.5.20. *If $t = \frac{2a^2}{k} OPT_{LP}$ and $r = \sqrt{\log n}$ then $\Pr\Big[\max_{u\in\mathcal{U}} \sum_{u\in S_i} Y_i > \frac{t}{4r}\Big] \leq \frac{1}{n^{\frac{2a^4 n}{fk^2}-1}}$.*

Proof. By applying union bound and applying Theorem 4.5.18 we get,

$$\begin{aligned}
\Pr\Big[\max_{u\in\mathcal{U}} \sum_{u\in S_i} Y_i > \frac{t}{4r}\Big] &\leq \sum_{u\in\mathcal{U}} \Pr\Big[\sum_{u\in S_i} Y_i > \frac{t}{4r}\Big] \\
&\leq n\exp\Big(-\frac{t^2}{16r^2\times f\times 1/4}\Big) \\
&\leq n\exp\Big(-\frac{t^2}{4r^2\times f}\Big).
\end{aligned}$$

By substituting $t = \frac{2a^2}{k}\text{OPT}_{\text{LP}}$ and $r = \sqrt{\log n}$, we get

$$\Pr\Big[\max_{u\in\mathcal{U}} \sum_{u\in S_i} Y_i > \frac{t}{4r}\Big] \leq n\exp\Big(-\frac{9a^4 n\log^2 n}{4k^2\log n\times f}\Big) \leq n\exp\Big(-\frac{2a^4 n\log n}{k^2 f}\Big) \leq \frac{1}{n^{\frac{2a^4 n}{fk^2}-1}}.$$

□

Theorem 4.5.21. $\Pr[B > 3\left(\frac{a^2}{k}\right)OPT_{LP}] \leq \frac{1}{n} + \frac{1}{n^{\frac{2a^4n}{fk^2}-1}}$.

Proof. From Theorem 4.5.8, we know that

$$\Pr[B \geq \mathbb{E}[B] + t] \leq \left(1 + \frac{t}{\mathbb{E}[B]}\right)^{-r/2} + \Pr\left[B_1^* > \frac{t}{4r}\right]$$

and thus,

$$\Pr[B \geq \frac{a^2}{k}\text{OPT}_{\text{LP}} + t] \leq \left(1 + \frac{t}{\mathbb{E}[B]}\right)^{-r/2} + \Pr\left[\max_{u \in \mathcal{U}} \sum_{u \in S_i} Y_i > \frac{t}{4r}\right].$$

Letting $t = \frac{2a^2}{k}\text{OPT}_{\text{LP}}$ and $r = \sqrt{\log n}$, we get the following estimate by applying Lemma 4.5.20, since $t > \mathbb{E}[B]$ and $\text{OPT}_{\text{LP}} \geq \sqrt{n}\log n$.

$$\begin{aligned}
\Pr[B > \frac{3a^2}{k}\text{OPT}_{\text{LP}}] &\leq \exp(-\frac{r}{3}) + \frac{1}{n^{\frac{2a^4n}{fk^2}-1}} \\
&= \exp(-\sqrt{\log n}/3) + \frac{1}{n^{\frac{2a^4n}{fk^2}-1}} \\
&= \frac{1}{\exp^{\sqrt{\log n}/3}} + \frac{1}{n^{\frac{2a^4n}{fk^2}-1}}.
\end{aligned}$$

□

Theorem 4.5.22. *Assuming that $a < \alpha_g \frac{1-\delta}{6k}$, $f \leq \frac{a^4n}{k^2}$, and for any $\delta > 0$, we have*

$$\Pr[N_g \geq ((1-\delta)\alpha_g - 6ak)\frac{a}{k}OPT_{LP}] \geq 1 - \epsilon_n.$$

Proof. Since $N_g \geq A - 2kB$ and applying union bound on Theorems 4.5.6 and 4.5.21, we get

$$\begin{aligned}\Pr[N_g \leq ((1-\delta)\alpha_g - 6ak)\frac{a}{k}\mathrm{OPT}_{\mathrm{LP}}\Big] &\leq \frac{1}{\exp^{\sqrt{\log n}/3}} + \frac{1}{n^{\frac{2a^4n}{fk^2}-1}} + \frac{1}{n^\ell} \\ &\leq \frac{1}{\exp^{\sqrt{\log n}/3}} + \frac{1}{n^{\frac{2a^4n}{fk^2}-1}} + \frac{1}{n^\ell}.\end{aligned}$$

Therefore, assuming $f \leq \frac{a^4 n}{k^2}$, we get

$$\Pr[N_g \geq ((1-\delta)\alpha_g - 6ak)\frac{a}{k}\mathrm{OPT}_{\mathrm{LP}}\Big] \geq 1 - \epsilon_n. \tag{4.56}$$

□

Theorem 4.5.23. *For any feasible instance of* FAIR-K-SETPACKING*, if* $OPT_{LP} \geq \sqrt{n}\log n$*,* $f \leq \frac{a^4 n}{k^2}$*,* $\delta > 0$*, and* $a < \frac{(1-\delta)\alpha_g}{4k}$*, then*

$$\Pr\Big[\big((1-\delta)\alpha_g - 6ak\big)\frac{a}{k}OPT_{LP} \leq N_g \leq (1+\delta)\alpha_g\frac{a}{k}OPT_{LP}\Big] \geq 1 - \epsilon_n.$$

Proof. This is a direct consequence of Theorems 4.5.21 and 4.5.7 combined with a union bound. □

Theorem 4.2.3. *For any instance of* FAIRSP *satisfying* $f = o(n/k^6\ell^4)$ *and* $\alpha_g = \Theta(1/\ell)$ *for all* $g \in [\ell]$*,* FAIRELIMINATE *is a randomized, polynomial-time algorithm with an approximation factor of* $\mathcal{O}(\ell k^2)$ *on the packing objective. Moreover, for instances with sufficiently large packing size,* FAIRELIMINATE *guarantees SRF with* $\gamma = \mathcal{O}(1)$*.*

Proof. The theorem follows from Theorem 4.5.23. □

4.6 Experiments

Our experiments run on commodity hardware using Python 3.6 with NumPy [130] as well as IP and LP solvers in Gurobi 9.1.2. The empirical evaluation of our algorithms uses both purely synthetic and realistic kidney exchange FAIRSP instances. To benchmark our algorithms, we include i) ALG-BLIND, the LP-rounding (fairness agnostic) k-set packing randomized algorithm from [119] as well as ii) IP-FSP and iii) IP-BLIND, which solve the integer restrictions of LP (4.3) with and without fairness constraints, respectively. The integer restrictions of IP-FSP and IP-BLIND make the fairness guarantee be satisfied deterministically as opposed to in expectation like RF.

4.6.1 Evaluation of randomized algorithms

FAIRSAMPLE and ALG-BLIND provide objective guarantees in expectation thus their reported objective values are the mean objectives of 300 independent LP solution roundings. FAIRSAMPLE uses $L = 300$ to estimate q'_i values. Since FAIRELIMINATE provides ex-post fairness guarantees the reported objective values correspond to a *single* LP rounding per FAIRSP instance.

4.6.2 Kidney exchange datasets

We use realistic kidney exchange instances drawn from the US-based United Network for Organ Sharing (UNOS) fielded exchange program; our synthetic instances are generated from this data in the standard way [131]. In real programs,[8] both *blood type* and *patient sensitization*, a measure that is high when a patient's body has built up many antibodies to potential donor organs, are taken into account

[8]For example, the OPTN/UNOS exchange transparently reports its prioritization points in Table 13-2 of [132]; high values for the sensitization metric CPRA are prioritized extensively, and O-type candidates are prioritized as well. For summaries of countries' exchanges in the European Union and the United Kingdom, we refer the reader to [116].

when deciding what type of participant to prioritize. Thus, in our experiments, we treat sensitization as a binary sensitive attribute (taking value 1 if the patient is sensitized and 0 otherwise), and separately treat patient blood type as a different binary sensitive attribute (taking value 1 if the patient is a hard-to-match O-type blood, and 0 otherwise). These binary attributes give rise to two datasets, each with $\ell = 2$, which we call the *blood group* and *sensitization* datasets. We generate graphs of size 64, 128, 256, and 512 from that real data, noting that the largest exchanges in the world hover at or slightly below 256.[9]

Recall as mentioned in the introduction, the FAIRSP instances have $\mathcal{S}$ consisting of all 2- and 3-cycles in said graphs. As a result, the largest FAIRSP instances have $n = 512$ and $m \approx 28000$. We omit IP solutions for these largest instances as solving them takes too long and they only serve to motivate probabilistic fairness. Each graph size has 32 instances for which we create fair instances with proportionality vector $\vec{\alpha} \in \{(0.25, 0.75), (0.5, 0.5), (0.75, 0.25)\}$. $w_i = |S_i|$ for all $S_i \in \mathcal{S}$, and $k = 3$.

4.6.3 Experiments on kidney exchange datasets

Altogether, the experiments solve a total of 864 IPs with fairness constraints, 960 LPs with fairness constraints, 96 IPs without fairness constraints and 128 LPs without fairness constraints. Both FAIRSAMPLE and FAIRELIMINATE produce solutions swiftly, taking up to 7 minutes for the largest instances. In both cases the bottleneck is solving the LP. Given there are multiple FAIRSP instances per graph size, the plots report the mean over all instances and the shaded regions capture 95% of the

[9] As examples, we direct the reader to [116], a recent survey of 17 European countries' national exchange programs, stating that the UK "has become the largest operating [kidney exchange program] in Europe, with 250 recipient-donor pairs registered per matching run," followed by the Netherlands and then Spain, with 110 recipient-donor pairs registered per run, and a long tail of countries afterward. In the US, programs tend to match at a faster cadence which can keep pool sizes low; for example, arguably the most successful program in the US, which is run by the National Kidney Registry (NKR), had a pool size hovering around 150 recipient-donor pairs in Q1 2022 [133].

instances.

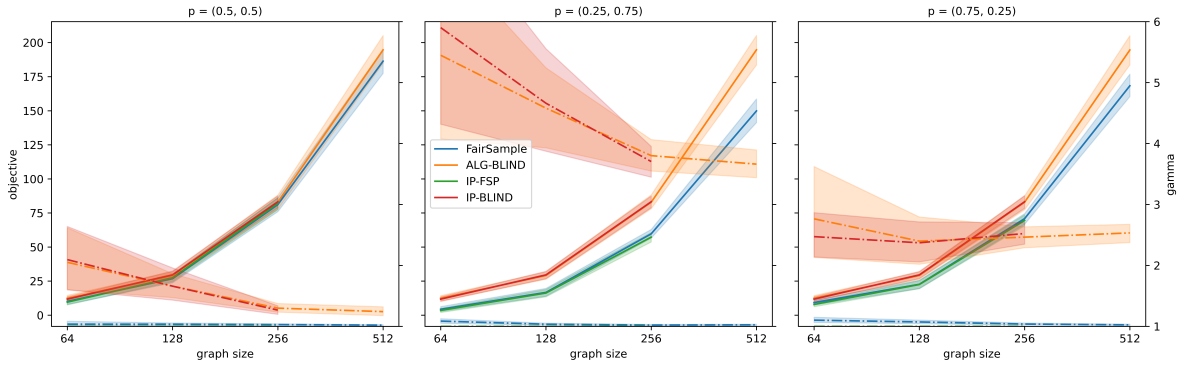


Figure 4.2: Plots for the *sensitization* dataset. Left (right) y-axes and solid (dashed) lines correspond to packing objectives (RF γ values).

Figure 4.1 uses all 32 instances with $n = 256$ from the *blood group* dataset over each of 40 choices of $\vec{\alpha}$; hence running FAIRSAMPLE 1280 times in 9 hours. Figure 4.2 shows FAIRSAMPLE achieves remarkably improved fairness at little cost to the objective compared to its fairness agnostic counterparts ALG-BLIND and IP-BLIND. IP-FSP also ran into several infeasible instances while this was never an issue for FAIRSAMPLE, further making the case for probabilistic fairness.

Experiments in Figure 4.3 use FAIRELIMINATE to round the 960 LPs with fairness constraints with 25 evenly spaced choices of the parameter $a \in [0.1, 0.99]$. Both the approximation and fairness factors improve as a increases. Although the worst-case guarantees in Theorem 4.2.3 are valid for $a < (1 - \delta)\alpha^*/6k$ (for any $\delta > 0$), in this real-world example the fairness guarantees continue to improve as a approaches 1. One explanation is that as a increases, the number of initially sampled sets increases but so does the number of sets dropped in the alterations step; however, since our instances are sparse the number of sets dropped is not too high. For comparison, we include the approximation and fairness factors from a *single* rounding of FAIRSAMPLE. Interestingly, here FAIRSAMPLE outperforms FAIRELIMINATE in ex-post fairness. One possible explanation is that the realistic instances are

far from worst-case.

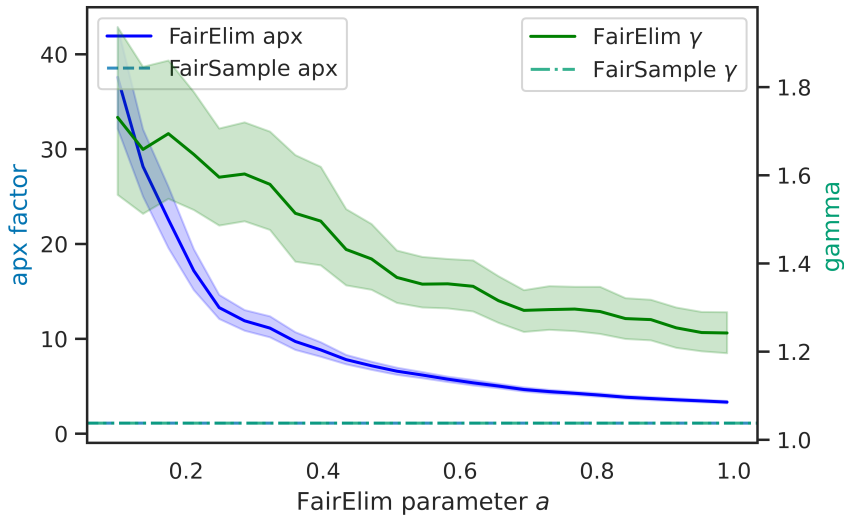


Figure 4.3: Blue (green) lines show approximation (fairness) factors and correspond to the left (right) y-axes. For comparison, we plot the average approximation factor ($= 1.12$) and γ ($= 1.04$) achieved by FAIRSAMPLE. Approximation factors are the LP objective divided by the rounded solution's objective.

4.6.4 DatasetMaxK:

We generate 200 purely synthetic FAIRSP instances with $n = 50$, $m = 1000$, $\ell \in \{2, 3\}$, and $\vec{\alpha}$ generated uniformly at random. Each instance has a parameter $\text{max\_k} \in \{3, 8, 13, \ldots, 49\}$. For each parameter configuration above, we independently generate 10 instances. Then this parameter max_k is used to sample sets. For each $i \in [m]$, $S_i \in \mathcal{S}$ is a random subset of $\mathcal{U}$ with cardinality $k_i \sim \mathcal{N}(\text{max\_k}, 2)$ (while ensuring k_i stays in between 2 and 50). For each $S_i \in \mathcal{S}$, w_i is an integer drawn uniformly between 5 and 35. Element colors are chosen uniformly at random. As expected, the objectives decrease as k increases, thus the approximation factor of FAIRSAMPLE increases alongside k.

We verify the objectives achievable decrease as max_k increases. Additionally, regardless of

k, FAIRSP continues to achieve better fairness than its fairness-agnostic counterparts. Intuitively, fairness disparity increases with the number of colors c. Since our fairness definitions consider the most "over-allocated" and "under-allocated" colors, it is reasonable that as the number of groups increases attaining fairness must become more deliberate.

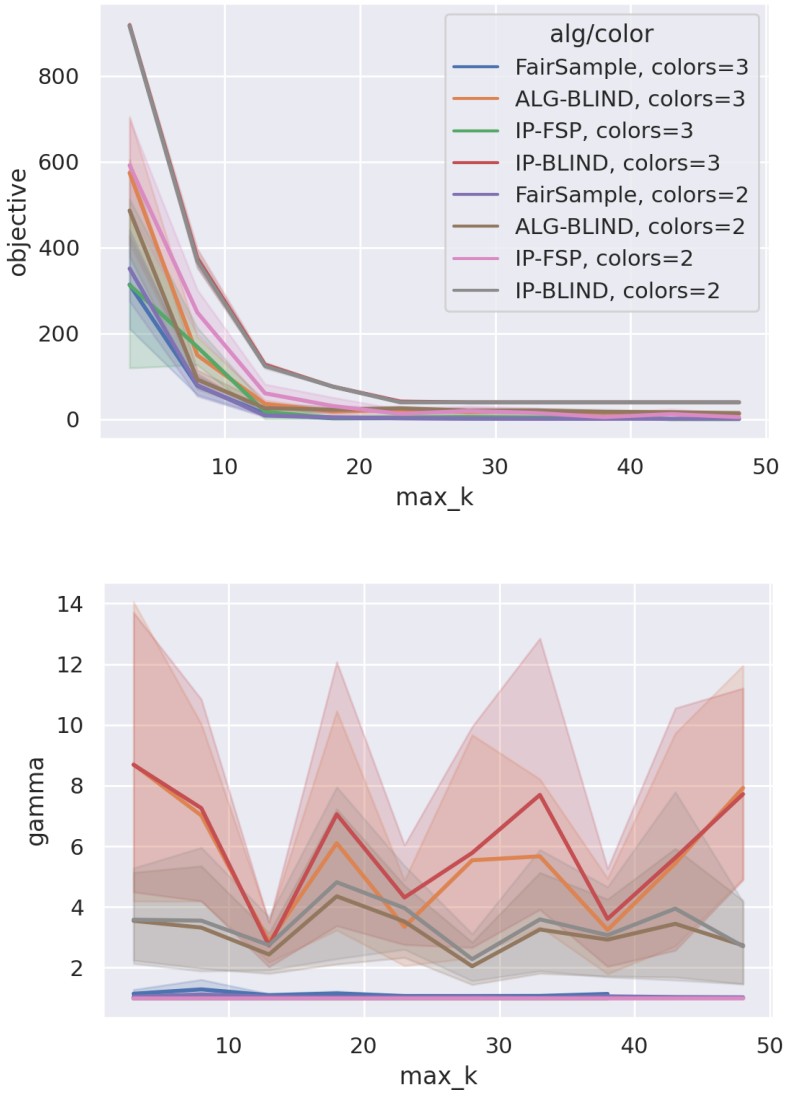


Figure 4.4: Objective and γ values on DatasetMaxK.

Chapter 5: Online Bipartite Matching under Uniform Metric

This chapter is largely based on our paper [50].

5.1 Introduction

In the past decade, the online smartphone computing has increased tremendously with the applications like assignment of cabs to customer (like Uber, Lyft), assignment of client requests to servers, traditionally called the k-server problem, real-time crowdsourcing where workers are to be assigned to jobs etc, can be modelled as Online Minimum Bipartite Matching problem. Formally, the OMBM problem can be defined as finding a matching that maximizes the size of the matching with an objective of minimizing the overall cost (total distance for metric spaces) of the matching where the requests arrive dynamically and are to be matched immediately to an unmatched (unassigned) service.

For OMBM (Online Minimum Bitpartite Matching Problem), the first results were given independently by Khuller et.al in [134] and Kalyanasundaram and Pruhs in [135] for the deterministic lower bound. They showed that for non-metric spaces the competitive ratio of any algorithm is unbounded. For metric spaces, they gave an algorithm with competitive ratio of $2n - 1$ which is best possible for deterministic algorithms. Later, [136] has given the first randomized algorithm based on a greedy approach, which has a sublinear competitive ratio, which is then improved to $\mathcal{O}\left(\log^2 n\right)$ by [137] and also gave a lower bound of $\Omega\left(\log n\right)$. All the above results are when the arrivals are in

adversarial order.

The first results for the random order of arrivals is given by Gairing and Klimm in [138]. They proved that for random order arrivals, the greedy algorithm gives a competitive ratio of atmost n. They complemented this result with a lower bound for the greedy algorithm, which has a competitive ratio of $\Omega\left(n^{0.292}\right)$ (when the metric space is line) for the random order of arrivals. For random order arrivals, the optimal algorithm is given by Raghavendra in [139] which has a competitive ratio of $\mathcal{O}\left(\log n\right)$. The same algorithm is also optimal for the adversarial model of the OMBM problem for general metric space.

The online metric matching problem has been studied for the case of special metrics as well. The online metric matching problem is similar to the wellstudied k-server problem introduced by Manasse et.al in [49] as it can be viewed as a restricted case of the k-server problem where the servers do not move. The methods used to give lower and upper bounds for the online matching problem, particularly on the uniform metric, are closely related to the results for k-server. Bansal et.al [137] gives a $\Theta(\log k)$ lower and upper bounds for the uniform metric under adversarial order arrival of requests. We recall that uniform metric consists of set of points where distance between any two points is the same, without loss of generality, we can assume that this distance is one unit. However, the problem of OMBM is open for uniform metric under random order of arrival model.

5.2 Problem formulation

OMBM (online bipartite matching problem) can be defined as finding a matching that maximizes the cardinality of the matching with an objective of minimizing the overall cost (total distance for metric spaces) of the matching where the requests arrive dynamically and are to be matched immediately

to a unmatched service.

Definition 5.2.1 (Online Minimum Matching Problem)**.** Given a set of service providers with their locations in a metric space $S = \{(s_1, p_1), (s_2, p_2), \ldots (s_m, p_m)\}$ and a set of user requests $R = \{(r_1, q_1), (r_2, q_2), \ldots (r_n, q_n)\}$ which appear one after the other whose location is also in the same metric space with k points and a distance metric as $d : [k] \to [k]$. Each user request has to be immediately matched with a service provider and the cost of the matching r_i to s_j is given by $d(p_i, q_j)$. Therefore the overall cost of a matching $M \in \mathcal{M} : [n] \to [m]$ is given by $Cost(M) = \sum_{(r_i, s_j) \in M} d(p_i, q_j)$. In a Online Minimum Matching Problem, our goal is to find a matching $M^* = \arg\min_{M \in \mathcal{M}} \text{Cost}(M)$.

Online algorithms can be evaluated using an evaluation standard called "Competitive Ratio", which is the ratio of the solution of an online algorithm to the optimal solution that can be obtained in the offline scenario overall all the instances of the problem. The performance evaluation here also depends on the input sequence of the requests. There are two different models of evaluation namely, the adversarial order of arrivals(worst case analysis) in which the input sequence can be considered to be coming from an adversary. The other model is called random order arrival model (average case analysis) where the input sequence is randomly generated.

Definition 5.2.2 (Adversarial Model - Competitive Ratio)**.** For a given online algorithm AO, the competitive ratio for an adversarial model is given as the maximum ratio of the cost of matching M_{AO} obtained by AO to the cost of optimal matching M_{opt} obtained when we know the entire input sequence beforehand (i.e, offline scenario) over all input instances where π is an arbitary arrival order of the requests.

$$\text{CR}_{\text{ao}} = \max_{(S,R),\pi of R} \frac{Cost(M_{AO})}{Cost(M_{opt})}$$

Definition 5.2.3 (Oblivious Adversarial Model - Competitive Ratio)**.** For a given online algorithm AO, the competitive ratio for an adversarial model is given as the maximum ratio of the cost of matching M_{AO} obtained by AO to the cost of optimal matching M_{opt} obtained when we know the entire input sequence beforehand (i.e, offline scenario) over all input instances where π is an arbitary arrival order of the requests.

$$\text{CR}_{\text{oa}} = \max_{(S,R),\pi of R} \frac{Cost(M_{OA})}{Cost(M_{opt})}$$

Definition 5.2.4 (Random Order Model- Competitive Ratio)**.** For a given online algorithm RO, the competitive ratio for a random order arrival model is given as the maximum ratio of the expected cost of the matching M_{RO} obtained by RO to the cost of optimal matching M_{opt} obtained when we know the entire input sequence beforehand (i.e, offline scenario) over all input instances.

$$\text{CR}_{\text{ro}} = \max_{(S,R)} \frac{\mathbb{E}_{\pi}[Cost(M_{RO})]}{\mathbb{E}_{\pi}[COST(M_{opt})]}$$

5.3 Deterministic Greedy algorithm

5.3.1 Average case analysis of greedy algorithm

In a deterministic greedy algorithm, each request is assigned to the nearest service provider that hasn't been matched yet. It is to be noted that the number of service provides, $m \geq n$ where n is the number of requests. Our main results is to address the open problem mentioned in [18] about the competitive ratio of the online minimum bipartite matching problem under random order arrival model (i.e, the requests arrive in random order) for uniform metric space. Let us assume a uniform metric space with k points, with S as the set of m service providers. A set of n requests arrive dynamically

in a random order, which is often referred to as Random Order Arrival model. For average case analysis of the greedy algorithm for a given configuration of the servers, we use a urn model as the underlying metric and the service provider configuration wouldn't effect the analysis. The following lemma proves this.

Lemma 5.3.1. *A greedy algorithm for online matching on a uniform metric with random order of arrivals doesn't depend on the location configuration of the service providers i.e,* $CR_{DG} = \frac{\mathbb{E}_\pi[Cost(M_{DG})]}{\mathbb{E}_\pi[COST(M_{opt})]}$.

Proof. On a uniform metric space with k points with a set of service providers S, the expected cost of DG (deterministic greedy algorithm) under ROA (Random Order Arrival) model is the same. Let us assume two different configurations of the server locations C and C', Let the expected cost of greedy algorithm for configuration C $\mathbb{E}_\pi[Cost(M_{DG})]$ be $f(k, m, n, C)$ and we have the following recursion,

$$f(k, m, n, C) = \frac{m}{k} f(k-1, m-1, n-1, C)$$

$$+\frac{k-m}{k}(1 + f(k-2, m-1, n-1, C))$$

Similarly for the configuration C', the expected cost is

$$f(k, m, n, C') = \frac{m}{k} f(k-1, m-1, n-1, C')$$

$$+\frac{k-m}{k}(1 + f(k-2, m, n-1, C'))$$

It is clear from the above equations that the expected cost doesn't depend on the underlying configuration but only on the number of servers and the number of requests.

□

5.3.2 Alternative model for analysis of deterministic greedy algorithm

The above lemma hence reduces the online matching of requests for ROA model under uniform metric to a simple urn model. In this model, there is a urn with k balls of read and green colour. There are m green balls in the urn. Now, we define a random process that clearly represents our original matching problem and the cost of this random process has a clear mapping to the greedy algorithm.

Definition 5.3.2. We draw n balls uniformly at random from the urn without replacement. So, each of the k balls can be picked with probability $\frac{1}{k}$. There are two cases : (a) If a green ball is picked, there wouldn't be any cost. (b) If a red ball is picked, it would incur a unit cost. And we remove one green ball from the urn, making the number of balls in the urn from k to $k-2$.

Let $g(k, m, n)$ be the expected cost of drawing n balls in the above random process where k is the total number of balls and m is the number of greens balls. Therefore the expected cost is

$$g(k, m, n) == \frac{m}{k} g(k-1, m-1, n-1) + \frac{k-m}{k}(1 + g(k-2, m, n-1))$$

For the soundness of the analysis we use $k = 2n$, subsequently $m = n$.

Theorem 5.3.3. *The random process 5.3.2 incurs an expected cost of drawing n balls is $\mathcal{O}(n^2)$ for a urn with k balls out of which m balls are green.*

Proof. Expected number of red balls picked will give us the expected cost of the random process 5.3.2. Let X_i be the indicator random variable that the i^{th} ball picked is *red*. The overall cost is $\mathbb{E}_\pi[X_1 + X_2 + X_2 + \ldots X_n]$. Assuming that $k = 2n$, WLOG , let the total number of red balls picked till the i^{th} turn be $Z_i = \sum_{j=1}^{j=i} X_i$. The expected number of balls picked in n turns is

$$
\begin{aligned}
\mathbb{E}[Z_n] &= \mathbb{E}[\sum_{i=1}^{i=n} X_i] = \sum_{i=1}^{i=n} \mathbb{E}[X_i] \\
&= \sum_{i=1}^{i=n} \sum_{j=1}^{i-1} \mathbb{E}[X_i | Z_{i-1} = j] \\
&= \sum_{i=1}^{i=n} \sum_{j=1}^{i-1} Pr[X_i = 1 | Z_{i-1} = j] \\
&= \sum_{i=1}^{i=n} \sum_{j=1}^{i-1} \frac{k-(m-i+1)}{k-(i-1+j)} \\
&= \sum_{i=1}^{n} (k-m+i-1)\Big(\frac{1}{k-i} + \cdots + \frac{1}{k-2(i-1)}\Big) \\
&= \sum_{i=1}^{n} k-m+i-1\Big(H_{k-i} - H_{k-2(i-1)}\Big) \\
&= (k-m)\sum_{i=1}^{n} \Big(H_{k-i} - H_{k-2(i-1)}) + \sum_{i=1}^{n} (i-1)(H_{k-i} - H_{k-2(i-1)}\Big) \\
&\leq (k-m+n)\Big(\sum_{i=1}^{n} H_{k-2i} - \sum_{i=n+1}^{2n-1} H_{k-i}\Big) \\
&= (k-m+n)\Big(\sum_{i=1}^{i=2n-1} H_i - 2\sum_{i=1}^{i=n} H_i\Big) \\
&= (n)(2nH_{2n} - 2nH_n) = n(2n\log 2 - \frac{1}{2} + \frac{1}{8n}) \text{ as } n \to \infty \\
&= \mathcal{O}\left(n^2\right)
\end{aligned}
$$

□

We now see that the expected cost of offline optimal algorithm for the OMBM problem for ROA model under uniform metric is $\mathbb{E}_{\pi \approx [k]}[Cost(M_{opt})] = \frac{(k-m)n}{k}$. Therefore, the competitive ratio for the above problem is $\mathcal{O}\left(\frac{(n^2)k}{(k-m)n}\right) = \mathcal{O}\left(\frac{(n^2)(2n)}{(n)n}\right) = \mathcal{O}\left(n\right)$ with $k = 2n, m = n$.

5.3.3 Lower bound for deterministic algorithms

In this section we prove the lowerbound that can be achieved on the competetitive ratio of deterministic algorithms for the OMBM problem that is being discussed. Clearly, the lower bound matches the upper bound for any deterministic algorithm. For any deterministic algorithm, there is no change in the expected cost that can be attained. Hence, the competitive ratio remains the same, thus matching the greedy algorithm's upper bound.

Theorem 5.3.4. *There is no deterministic algorithm with a competitive ratio bettwe than $\Omega\left(n^2\right)$ for an OMBM problem on a uniform metric under random order arrival model where k is the total number of points in the metric space, m is the number of service providers and n is the number of requests.*

Proof. We know that each request that arrives at a location without a unmatched service provider incurs a cost of atleast 1 (otherwise a cost of 0). Let us say that the request is a mishit if it appears at a location without a unmatched service provider (the request with a cost of atleast 1). Now, the expected cost of any deterministic assignment for the i^th request be $\mathbb{E}[X_i]$ where X_i is the indicator random variable that the i^{th} request is a mishit. The total expected cost is $\mathbb{E}_\pi[X_1 + X_2 + X_2 + \ldots X_n]$. Let the total number of mishits till i^{th} request be $Z_i = \sum_{j=1}^{j=i} X_i$. The expected cost of any deterministic

algorithms for n requests is

$$\begin{aligned}
\mathbb{E}[Cost(DA)] \geq \mathbb{E}[Z_n] = \mathbb{E}[\sum_{i=1}^{i=n} X_i] &= \sum_{i=1}^{i=n} \mathbb{E}[X_i] \\
&= \sum_{i=1}^{i=n} \sum_{j=1}^{i-1} \mathbb{E}[X_i | Z_{i-1} = j] \\
&= \sum_{i=1}^{i=n} \sum_{j=1}^{i-1} Pr[X_i = 1 | Z_{i-1} = j] \\
&= \sum_{i=1}^{i=n} \sum_{j=1}^{i-1} \frac{k-(m-i+1)}{k-(i-1+j)} \\
&= \sum_{i=1}^{n} k - m + i - 1 \Big(H_{k-i} - H_{k-2(i-1)} \Big) \\
&\approx (k-m+n) \cdot c \\
&\geq (n) \Big(\sum_{i=1}^{n} H_{k-2i} - \sum_{i=n+1}^{2n-1} H_{k-i} \Big) \\
&= n(2n \log 2 - \frac{1}{2} + \frac{1}{8n}) \geq n^2 = \Omega\left(n^2\right)
\end{aligned}$$

□

From Theorems 5.3.3 and 5.3.4 we can infer that the Deterministic greedy algorithm meets the lower bound proving that the solution obtained is tight. The below theorem formalizes this result.

Theorem 5.3.5. *Deterministic Greedy achieves a competitive ratio of* $\Theta(n)$ *for OMBM on the uniform metric in the random order arrival model where* $k(= 2n)$ *is the total number of points in the metric space,* $m(= n)$ *is the number of service providers and* n *is the number of requests.*

In the following section, we consider an oblivious adversary which results in improved competitive ratio. We further show that randomized greedy algorithm is optimal for the problem.

5.4 Randomized greedy algorithm

5.4.1 Average-case Analyses of Randomized Greedy

Randomized Greedy (RG) was first introduced in [136]. It can be viewed as a natural generalization of (deterministic) Greedy. Consider the scenario when an online request r arrives, and let S_r be the set of nearest unmatched service providers with respect to r. If $|S_r| \leq 1$, then RG is reduced to Greedy, i.e., choose the element in S_r if there is. However, in the case $|S_r| \geq 2$, RG will sample one choice from S_r uniformly instead of arbitrarily choosing one from S_r. Generally speaking, RG can outperform Greedy in most theoretical worst cases. In this section, we offer rigorous average-case analysis (i.e., random order) for RG over two special metrics, namely uniform metric[1] and line metric[2], and try to compare its performance with that of Greedy.Throughout this paper, we use $H_n \doteq \sum_{i=1}^{n} 1/i$ to denote the nth Harmonic number for each given integer n. Recall that for two unbounded increasing functions $f(n), g(n)$, $f \sim g$ iff $\lim_{n\to\infty} f/g = 1$.

5.4.2 RG on the Uniform Metric

[136] analyzed Greedy and RG on the uniform metric but in the *adversarial* model. The main results are: (1) Greedy achieves an online competitive ratio of $k = \min(|S|, |R|)$, which is optimal among all deterministic online algorithms. In other words, no deterministic online algorithm can attain a ratio better than k; (2) RG achieves an online ratio of $H_k \sim \log k$, which is optimal among all possible randomized algorithms. From (2), we have the below lemma.

Lemma 5.4.1. *Randomized Greedy achieves an online ratio at most $H_k \sim \log k$ for the OMBM on the*

[1] All edge costs are either 0 or 1 satisfying triangle inequality.
[2] All vertices can be arranged in a line.

uniform metric in the random order model.

Proof. Notice that the model of random order is surely *as easy as* that of adversarial, if not strictly harder. Thus for any algorithm, any upper bound result in the adversarial model can directly extend to the case of random order. Therefore, we get the above claim directly from [136]. □

In this section, we prove the following main hardness result.

Theorem 5.4.2. *No randomized algorithm can achieve an online ratio strictly better than (i.e., less than)* $(1 + \frac{1}{k})(H_{k+1} - 1) \sim \log k$ *for OMBM on the uniform metric in the random order model.*

The following claim directly follows by Lemma 5.4.1 and Theorem 5.4.2.

Corollary 5.4.3. *Randomized Greedy is optimal among all randomized algorithms for the OMBM problem on the uniform metric under the random order model. It achieves an online ratio of* $(1 + o(1)) \log k$*, where* $k = \min(|R|, |S|)$ *and* $o(1)$ *is a vanishing term when* $k \to \infty$.

By comparing results in the above corollary and those in [136], we see that for OMBM with uniform metric, RG achieves the same online ratio in the model of random order as that of adversarial. That means when applying RG to OMBM with uniform metric, the assumption of random order does not offer any extra algorithmic power compared to that of adversarial. Now we mainly apply Yao's Principle to prove the main hardness theorem. We try to construct a family of instances $\mathcal{I}$ and a distribution $\mathcal{D}$ over it such that the expected performance of the best deterministic algorithm is at least $(1 + \frac{1}{k})(H_{k+1} - 1)$. We generate a family of random instances as follows. Let $S = (s_1, s_2, \ldots, s_k)$ and $R = (r_1, r_2, \ldots, r_k)$. Let $\Sigma = \{\sigma\}$ be the set of all possible random permutations over $[k] = \{1, 2, \ldots, k\}$. For each $\sigma \in \Sigma$, let I_σ be such an instance $G = (S, R)$, where edges $(r_i, s_{\sigma(i)})$ have zero cost with $1 \leq i \leq k - 1$ and all the rest have cost one. Set $\mathcal{I} = \{I_\sigma | \sigma \in \Sigma\}$ and $\mathcal{D}$ as the

uniform distribution, i.e, each time an instance is uniformed sampled from $\mathcal{I}$. Observe that (1) each instance $I \in \mathcal{I}$ has an offline optimal cost of 1 and (2) each user $i \leq n-1$ has at exactly one zero-cost neighbor (service provider). For a given deterministic algorithm DA, let $\mathbb{E}_{\sigma\sim\Sigma}[\mathrm{DA}(I_\sigma)]$ be the expected performance of DA over a random instance I_σ with σ being uniformly sampled from Σ. Let $F(k) = \min_{\mathrm{DA}} \mathbb{E}_{\sigma\sim\Sigma}[\mathrm{DA}(I_\sigma)]$, which denotes the performance of the best deterministic algorithm over the random instances specified by $(\mathcal{I}, \mathcal{D})$. Now we try to show $F(k) \geq (1+\frac{1}{k})(H_{k+1}-1)$ by induction over k.

Lemma 5.4.4.

$$F(k) \geq (1+\frac{1}{k})(H_{k+1}-1), \forall k = 1, 2, \ldots \tag{5.1}$$

Notice that Lemma 5.4.4 together with Yao's Principle immediately yields Theorem 5.4.2. Now we present a formal proof for the above lemma.

Proof. Consider the basic case $k = 1$. We see that $\mathcal{I}$ has only one single instance $I = (R, S)$, where there is one single edge (r, s) with cost one. Thus, we claim that for any DA, $\mathrm{DA(I)} = 1$, implying $F(1) = 1$.

Now assume Inequality (5.1) is valid for all integers from 1 to $k-1$. We prove the case of k. Consider any given deterministic algorithm DA. Observe that for each given instance I, $\mathrm{DA}(I) = \mathbb{E}_{\pi\sim\Pi}[\mathrm{DA}(I, \pi)]$, where the expectation is taken over all possible permutations π of T. Thus, the expected performance of DA over a uniformed sampled instance of $\mathcal{I}$ can be rewritten as

$$\mathbb{E}_{\sigma\sim\Sigma,\pi\sim\Pi}[\mathrm{DA}(I_\sigma, \pi)] = \mathbb{E}_{\pi\sim\Pi,\sigma\sim\Sigma}[\mathrm{DA}(I_\sigma, \pi)]$$

Thus, we can interpret the expected performance of DA over a uniformly sampled instance from

$\mathcal{I}$ in the following equivalent way: first we sample a random permutation π over $[k] \doteq \{1, 2, \ldots, k\}$ and assume $\{t_{\pi(\ell)}, \ell = 1, 2, \ldots\}$ is the arrival sequence; then choose a random permutation σ over $[k]$ and designate $w_{\sigma(i)}$ as the unique zero-cost neighbor for the user t_i with $1 \leq i \leq n-1$. Now without losing of generality, assume that for each online user t, DA will first check the availability of its zero-cost neighbor and then follow another specific order to check the rest.

Now consider a given π and let $\pi(\ell) = k$, i.e, t_k appears at the ℓ position in π. We are sure that the matching cost of DA for all previous arrival users are zeros and the matching cost for $t_{\pi(\ell)} = t_k$ is one. Consider a fixed σ. Observe that when $t_{\pi(\ell')}$ arrives with $\ell' < \ell$, DA will match $t_{\pi(\ell')}$ with $w_{\sigma(\pi(\ell'))}$. Thus, when $t_{\pi(\ell)} = t_k$ arrives, DA will observe such a list of all one-cost neighbors[3] of t_k: $W - \{w_{\sigma(\pi(\ell'))}, 1 \leq \ell' < \ell\} \doteq W^\ell$. Conditioning on the observation of the list of W^ℓ, DA will *deterministically* select one choice from W^ℓ, say $\hat{w}$, and match it to t_k. Note that the *ideal* match for t_k in σ should be $w_{\sigma(k)}$, which is *unknown* to DA; meanwhile we are sure any given choice in W^ℓ will have same chance to be $w_{\sigma(k)}$. Since $|W^\ell| = k - \ell + 1$, we have that DA made the exact right choice for t_k with probability equal to $1/(n - t + 1)$. Essentially, we claim that

$$\Pr[\hat{\mathrm{w}} = \mathrm{w}_{\sigma(\mathrm{k})}] = \frac{1}{\mathrm{k} - \ell + 1} \tag{5.2}$$

Notice that (1) if $\hat{w} = w_{\sigma(k)}$, then $\mathrm{DA}(I_\sigma, \pi) = 1$; (2) if $\hat{w} \neq w_{\sigma(k)}$, then we can view the remaining process as applying another deterministic algorithm to another instance of length $k - t$, with t_k being replaced by t_i such that $w_{\sigma(i)} = \hat{w}$. Let $\mathrm{DA}(k)$ be the expected performance of DA over the uniformly sampled random instance I_σ when each side has k elements. Summarizing all above

[3] A one-cost neighbor of t refers to some w whose distance to t is one.

analysis, we have the following

$$\mathrm{DA}(k) \geq \frac{1}{k}\sum_{\ell=1}^{k-1}\Big(1+\big(1-\frac{1}{k-\ell+1}\big)F(k-\ell)\Big).$$

The inequality is due to our induction assumption. Solving the above recursive inequality, we get that $\mathrm{DA}(k) \geq (1+\frac{1}{k})(H_{k+1}-1)$. Since DA is an arbitrarily given deterministic algorithm, we claim that the best algorithm should also have a performance at least $(1+\frac{1}{k})(H_{k+1}-1)$. Thus, we complete the proof for the case of k. □

5.5 Conclusion

We study the deterministic and randomized algorithms for the OMBM problem under uniform metric and random order arrival model. We especially show the average case analysis of the deterministic and randomized greedy algorithms where the both of them are optimal in their respective families of algorithms.

Chapter 6: Fair Labeled Clustering

This chapter is largely based on our paper [51].

6.1 Introduction

Machine learning applications have seen widespread use across diverse areas from criminal justice to hiring to healthcare. These applications significantly affect human lives and risk contributing to discrimination [3, 140]. As a result, research has been directed toward the creation of fair machine learning algorithms [67]. Much existing work has focused on the supervised setting. However, significant attention has recently been given to clustering—a fundamental problem in unsupervised learning and operations research. While many important notions of fair clustering have been proposed, the most relevant to our work is group (demographic) fairness [7, 8, 12, 58, 141, 142, 143, 144, 145]. In many of those works, fairness is maintained at the cluster level by imposing constraints on the proportions of groups present in each cluster. For example, we may require the racial demographics of each cluster to be close to the dataset as a whole (demographic/statistical parity) or that no group is over-represented in any cluster.

While constraining the demographics of each cluster is appropriate in some settings, it may be unnecessary or impractical in others. In decision making applications, each cluster eventually has a specific label (outcome) associated with it which may be more positive or negative than others. If

the same label is applied to multiple clusters, we may only wish to bound the demographics of points associated with a given label as opposed to bounding the demographics of each cluster.

To be more concrete, consider the application of clustering for market segmentation in order to generate better targeted advertising [146, 147, 148, 149]. In this setting, we select or engineer features which are informative for targeted advertising and apply clustering (e.g., k-means) to the dataset. Then, we analyze the resulting centers (prototypical examples) and make decisions for targeted advertising in the form of recommending specific products or offering certain deals. These products or deals may have different levels of quality, i.e., we may assign labels such as: *mediocre*, *good*, or *excellent* to each cluster based on the quality of its advertisements. For the clusters of a given label (treated as one), it is possible that a certain demographic would be under-represented in the *excellent* label or that another could be over-represented in the *mediocre* label. In fact, the reports in [150, 151, 152] indicate that targeted advertising may under-represent certain demographics for some advertisements. An algorithm that ensures each group is represented proportionally in each label could remedy this issue. While applying group fair clustering algorithms would also ensure demographic representation in the clusters and thus the labels, it could come at the price of a higher deformation in the clustering since points would have to be routed to possibly faraway centers just to satisfy the representation proportions. On the other hand, ensuring fair representation across the labels, but not necessarily the centers is less restrictive and likely to cause less deformation to the clustering.

Another similar example is clustering for job screening [153] in which we have a dataset of candidates,[1] and each candidate is represented as a point in a metric space. Clustering could be applied over this set to obtain k many clusters. Then, the center of each cluster is given a more costly examina-

[1]In some countries, such as India, the number of candidates can be in the millions for government jobs: `https://www.bbc.com/news/world-asia-india-43551719`.

tion (e.g., a human carefully screening a job application). Accordingly, the centers would be assigned labels from the set: *hire*, *short-list*, *scrutinize further*, or *reject*. Naturally, more than one cluster could be assigned the same label. Clearly, the greater concern here is demographic parity across the labels, but not necessarily the individual clusters. Thus, group fair clustering would yield unnecessarily sub-optimal solutions.

While in the above examples the label of the center was decided according to its position in the metric space. One can envision applications in Operations Research where the label assignment of the center is not dependent on its position [154, 155]. Rather, we would have a set of centers (facilities) of different service types (or quality) and we would have a budget for each service type. Further, to ensure group fairness we would satisfy the demographic representation over the service types offered. In this setting, we would have to choose the labels so as to minimize the clustering cost subject to further constraints such as budget and fair demographic representation.

The above examples illustrate the need for a group fairness definition at the label level when clustering is applied in decision-making settings or when the different centers (facilities) provide different types of services. In addition to being sufficient, evaluating fairness at the label level rather than cluster level can also be necessary. When the metric space is correlated with group membership it may be costly, counterproductive, or impossible to get meaningful clusters that each preserve the demographics of the dataset. For example, if the metric space is geographic as in many facility location problems, a person's location can be correlated with their racial group membership due to housing segregation. The same is true in machine learning when common features like location redundantly encode sensitive features such as race. In this case, the more strict approach of group fairness in each cluster could cause a large enough degradation in clustering quality that the entity in charge chooses a classical "unfair" clustering algorithm instead. In legal terms, this unfair clustering approach may

exhibit *disparate impact*—members of a protected class may be adversely affected without provable intent on the part of the algorithm. However, disparate impact is allowed if the unfair clustering can be justified by *business necessity* (e.g., the fair clustering alternative is too costly)[156].

Thus, our work can be seen as a less stringent, less costly, and fundamentally different approach which still satisfies some similar fairness criteria to existing group fair clustering formulations. In addition, the decision-maker may not be concerned with the demographic representation in all labels, but rather only a specific set of label(s) such as *hire* and *short-list*. It may also be desired to enforce different lower and upper representation bounds for different labels.

6.2 Our Contributions

We introduce the problem of fairness in labeled clustering in which group fairness is ensured within the labels as opposed to each cluster. Specifically, we are given a set of centers found by a clustering algorithm, then having found the centers, we have to satisfy group fairness over the labels. We consider two settings: (1) **labeled clustering with assigned labels (**LCAL**)** where the center labels are decided based on their position as would be expected in machine learning applications and (2) **labeled clustering with unassigned labels (**LCUL**)** where we are free to select the center labels subject to some constraints. We note that throughout we consider the set of centers to be given and fixed (although in the unassigned setting their labels are unknown), therefore the problem is essentially a routing (assignment) problem where points are assigned to centers rather than a clustering problem. We however, refer to it as clustering since we minimize the clustering cost throughout and since our motivation is clustering based. Moreover, many of the application cases of the assigned labels setting would not alter the centers as that would not change the assigned labels which are given manually

through further inspection [146, 148, 153] or in the case of the unassigned labels we would have a fixed set of centers. Further, the work of [145] in fair clustering follows a similar setting where the centers are fixed.

For the LCAL (assigned labels) setting, we show that if the number of labels is constant, then we can obtain an optimal clustering cost subject to satisfying fairness within labels in polynomial time. This is in contrast to the equivalent *fair assignment* problem in fair clustering which is NP-hard[12, 157].[2] Furthermore, for the important special case of two labels, we obtain a faster algorithm with running time $O(n(\log n + k))$.

For the LCUL (unassigned labels) setting, we give a detailed characterization of the hardness under different constraints and show that the problem could be NP-hard or solvable in polynomial time. Furthermore, for a natural specific form of constraints we show a randomized algorithm that always achieves an optimal clustering and satisfies the fairness constraints in expectation.

We conduct experiments on real world datasets that show the effectiveness of our algorithms. In particular, we show that our algorithms provide fairness at a lower cost than fair clustering and that they indeed scale to large datasets.

6.3 Related Work

Much of the investigation into fairness in machine learning and automated systems was sparked by the seminal work of [67]. That work and others [158, 159] respond to the reality that points which should receive similar classifications, but belong to different demographic groups may not be near each other in the feature space. Our approach accounts for this phenomenon as well by allowing points from

[2] In this equivalent problem, the set of centers is given. We seek an assignment of points to these centers that minimizes a clustering objective and bounds the group proportions assigned to each center.

different groups to be distant in the metric space and assigned to different clusters, but receive the same label.

The most closely related work in the clustering space addresses group (demographic) fairness among the members of each cluster [7, 8, 12, 58, 141, 142, 143, 145, 157]. However, as noted earlier, these approaches can diverge quite a bit from the problem we consider and are not directly comparable. Some work also considers the less related fair data summarization problem of bounding group proportions among the set of centers/exemplars [144]. In addition, several other notions of fair clustering and summarization exist to capture the diverse settings and objectives for which fairness is desirable. These include service guarantees bounding the distance of points to centers [160], preserving nearby pairs or communities of points in the metric space [161], equitable group representation [162, 163], and fair candidate selection[164].

In particular, the setting of [145] is very similar to ours in that the set of centers is fixed, and the problem amounts to routing points to centers so as to minimize the clustering cost function. However, unlike our work, the constraint is to satisfy conventional group fairness in the clusters; whereas in our setting, we are concerned with group fairness only within the labels.

6.4 Preliminaries and Problem Formulation

We are given a complete metric graph with a set of vertices (points) $\mathcal{P}$ where $|\,\mathcal{P}\,| = n$. Further, each point has a color assigned to it according to the function $\chi : \mathcal{P} \to \mathcal{G}$ where $\mathcal{G}$ is the set of possible colors, with cardinality ℓ, i.e. $|\,\mathcal{G}\,| = \ell$. We refer to the set of points with color $\mathrm{g} \in \mathcal{G}$ by $\mathcal{P}_h$. We further have a distance function $d : \mathcal{P} \times \mathcal{P} \to \mathbb{R}_{\geq 0}$ which defines a metric. We are given a set S of centers that have been selected, S contains at most k many centers, i.e. $|S| \leq k$. Furthermore,

we have the set of labels $\mathcal{L}$ where $\mathcal{L}$ has a total of m many possible labels, i.e. $|\mathcal{L}| = m$. The function $\mathcal{X} : S \to \mathcal{L}$ assigns centers to labels. Our problem always involves finding an assignment from points to centers, $\phi : \mathcal{P} \to S$ such that it is the optimal solution to a constrained optimization problem where the objective is a clustering objective. Specifically, we always have to minimize the objectives: $\Big(\sum_{j\in\mathcal{P}} d^p(j,\phi(j))\Big)^{1/p}$, where $p = \infty, 1$, and 2 for the k-center, k-median, and k-means objectives, respectively. We note that for the k-center with $p = \infty$, the objective reduces to a simpler form $\Big(\sum_{j\in\mathcal{P}} d^p(j,\phi(j))\Big)^{1/p} = \max_{j\in\mathcal{P}} d(j,\phi(j))$ which is the maximum distance between a point j and its assigned center $\phi(j)$. We consider the number of colors ℓ to be a constant throughout. This is justified by the fact that in most applications demographic groups tend to be limited in number.

As mentioned earlier, we have two settings and accordingly two variants of this optimization: (1) labeled clustering with assigned labels (LCAL) where the centers have already been assigned labels and (2) labeled clustering with unassigned labels (LCUL) where the centers have not been assigned any labels and can be assigned any arbitrary labels from the set $\mathcal{L}$ subject to (possible) additional constraints.

We pay special attention to the two label case where $\mathcal{L} = \{P, N\}$ with P being a positive outcome label and N being a negative outcome label, although many of our results can be extended to the general case where $|\mathcal{L}| = m > 2$.

6.4.1 Labeled Clustering with Assigned Labels (LCAL)

In this problem the labels of the centers have been assigned, i.e. the function $\mathcal{X}$ is fully known and fixed. We look for an assignment ϕ which is the optimal solution to the following problem:

$$\min_{\phi} \Big(\sum_{j \in \mathcal{P}} d^p(j, \phi(j)) \Big)^{1/p} \tag{6.1}$$

$$\forall L \in \mathcal{L}, \forall g \in \mathcal{G} : \alpha_{L,\mathrm{g}} \sum_{\substack{i \in S \\ \mathcal{X}(i)=L}} |\, \mathcal{P}_i \,| \leq \sum_{\substack{i \in S \\ \mathcal{X}(i)=L}} |\, \mathcal{P}_{i,\mathrm{g}} \,| \leq \beta_{L,\mathrm{g}} \sum_{\substack{i \in S \\ \mathcal{X}(i)=L}} |\, \mathcal{P}_i \,| \tag{6.2}$$

$$\forall L \in \mathcal{L} : (UB)_L \leq \sum_{i \in S : \mathcal{X}(i)=L} |\, \mathcal{P}_i \,| \leq (UB)_L \tag{6.3}$$

where $\mathcal{P}_i$ refers to the points ϕ assigns to the center i, i.e. $\mathcal{P}_i = \{j \in \mathcal{P} \,|\phi(j) = i\}$. $\mathcal{P}_i^{\mathrm{g}} = \mathcal{P}_i \cap \mathcal{P}_h$, i.e. the subset of $\mathcal{P}_i$ with color g. $\alpha_{L,\mathrm{g}}$ and $\beta_{L,\mathrm{g}}$ are lower and upper proportional bounds for color g. Clearly, $\alpha_{L,\mathrm{g}}, \beta_{L,\mathrm{g}} \in [0, 1]$. Constraints (6.2) are the proportionality (fairness) constraints that are to be satisfied in fair labeled clustering. Notice how we have a superscript L in $\alpha_{L,\mathrm{g}}$ and $\beta_{L,\mathrm{g}}$, this is to indicate that we may desire different proportional representations in different labels. For example, for the case of two labels $\mathcal{L} = \{P, N\}$, we may not want to enforce proportional representation in the negative label so we set $\alpha_{N,\mathrm{g}} = 0$ and $\beta_{N,\mathrm{g}} = 1$ but we may want to enforce lower representation bounds in the positive label and therefore set $\alpha_{P,\mathrm{g}}$ to some non-trivial value. Note that these constraints generalize those of fair clustering, in fact we can obtain the constraints of fair clustering by letting each center have its own label ($m = k$) and enforcing the proportional representation bounds to be the same throughout all labels. However, in our problem we focus on the case where the number of labels m is constant since in most applications we expect a small number of labels (outcomes). In fact, a large number could cause a problem in terms of decision making and result interpretability.

In constraints (6.3), $(LB)_L$ and $(UB)_L$ are pre-set upper and lower bounds on the number of points assigned to a given label, clearly $(LB)_L, (UB)_L \in \{0, 1, \ldots, n\}$. They are additional constraints we introduce to the problem that have not been previously considered in fair clustering. Our

motivation comes from the fact that since positive or negative outcomes could be associated with different labels, it is reasonable to set an upper bound on the total number of points assigned to a positive label, since a positive assignment may incur a cost and there is a bound on the budget. Similarly, we may set a lower bound to avoid trivial solutions where most points are assigned to negative outcomes and no or very few agents enjoy the positive outcome.

6.4.2 Labeled Clustering with Unassigned Labels (LCUL):

In labeled clustering with unassigned labels LCUL, the labels of the centers have not been assigned. As noted, this captures certain OR applications in which the label of a center is not related to its position in the metric space.

Similar to the case with assigned labels LCAL, we would also wish to minimize the clustering objective. In general we have the following optimization problem:

$$\min_{\phi,\mathcal{X}} \Big(\sum_{j \in \mathcal{P}} d^p(j, \phi(j)) \Big)^{1/p} \tag{6.4}$$

$$\forall L \in \mathcal{L}, \forall h \in \mathcal{G} : \alpha_{L,\mathrm{g}} \sum_{\substack{i \in S \\ \mathcal{X}(i)=L}} |\, \mathcal{P}_i \,| \leq \sum_{\substack{i \in S \\ \mathcal{X}(i)=L}} |\, \mathcal{P}_{i,g} \,| \leq \beta_{L,\mathrm{g}} \sum_{\substack{i \in S \\ \mathcal{X}(i)=L}} |\, \mathcal{P}_i \,| \tag{6.5}$$

$$\forall L \in \mathcal{L} : (LB)_L \leq \sum_{i \in S : \mathcal{X}(i)=L} |\, \mathcal{P}_i \,| \leq (UB)_L \tag{6.6}$$

$$\forall L \in \mathcal{L} : (CL)_L \leq |S^L| \leq (CU)_L \tag{6.7}$$

Note how in the above objective $\mathcal{X}$ has been added as an optimization variable unlike the objective in (6.1) for LCAL. Further, we have added constraint (6.7) where S^L refers to the subset

of centers that have been assigned label L by the function $\mathcal{X}$, i.e. $S^L = \{i \in S | \mathcal{X}(i) = L\}$. This constraint simply lower bounds S^L by $(CL)_L$ and upper bounds it by $(CU)_L$. This constraint models minimal service guarantees (lower bound) and budget (upper bound) guarantees. Clearly, $(CL)_L, (CU)_L \in \{0, 1, \ldots, k\}$. Further, setting $(CL)_L = 0$ and $(CU)_L = k \; \forall L \in \mathcal{L}$ allows any label to have any number of centers, effectively nullifying the constraint. We show in a subsequent section that forcing certain constraints on the problem can make it NP-hard and that relaxing some constraints would make the problem permit polynomial time solutions.

6.5 Algorithms and Theoretical Guarantees for LCAL

6.5.1 LCAL is Polynomial Time Solvable:

LCAL is problem (6.1) where we have a collection of centers and we wish to minimize a clustering objective subject to proportionality constraints (6.2) and possible constraints on the number of points each label is assigned (6.3). Fair assignment[3] is a problem which has a very similar form to our problem; the centers have already been decided and we wish to satisfy the same proportionality constraints in every cluster, specifically the optimization problem is:

$$\min_{\phi} \Big(\sum_{j \in \mathcal{P}} d^p(j, \phi(j)) \Big)^{1/p} \tag{6.8}$$

$$\forall i \in S, \forall g \in \mathcal{G} : \alpha_{\mathrm{g}} |\, \mathcal{P}_i \,| \leq |\, \mathcal{P}_{i,\mathrm{g}} \,| \leq \beta_{\mathrm{g}} |\, \mathcal{P}_i \,| \tag{6.9}$$

It may be thought that the above optimization is simpler than that of LCAL (6.1), since all clusters have to satisfy the same proportionality bounds and there is no bound on the total number of

[3]Fair assignment [8, 12, 58] is a sub-problem solved in fair clustering to finally yield a full algorithm for fair clustering.

points assigned to a any specific cluster. However, [12, 157] show that the problem is in fact NP-hard for all clustering objectives. We show in the theorem below that LCAL can be solved in polynomial time for all clustering objectives.

Theorem 6.5.1. *Labeled clustering with assigned labels* LCAL *is solvable in polynomial time for the all clustering objectives (k-center, k-median, and k-means).*

Proof. The key observation is that any assignment function ϕ, will assign a specific number of points n_L to the centers with label L. Further, we have that $\sum_{L \in \mathcal{L}} n_L = n$ since all points must be covered. Now, since $|\mathcal{L}| = m$ is a constant, this means that there is a polynomial number of ways to vary the total number of points distributed across the labels. More specifically, the total number of ways to distribute points across the given labels is upper bounded by $\underbrace{n \times n \times \cdots \times n}_{m-1} = n^{m-1}$. Note that once we decide the number of points assigned to the first $(m-1)$ labels, the last label must be assigned the remaining amount to cover all n points, so we have a total of n^{m-1} possibilities. Since we have established, that there is a polynomial number of possibilities for distributing the number of points across the labels, if we can solve LCAL optimally for each possibility and simply take the minimum across all possibilities then we would obtain the optimal solution.

Now that we are given a specific distribution of number of points across labels, i.e. $(n_1, \ldots, n_m)$ where $\sum_{L \in \mathcal{L}} n_L = n$, we have to solve LCAL optimally for that distribution. The problem amounts to routing points to appropriate centers such that we minimize the clustering objective and satisfy the distribution of number of points across the labels along with the color proportionality. To do that we construct a network flow graph and solve the resulting minimum cost max flow problem. The network flow graph is constructed as follows:

- **Vertices:** the set of vertices is $V = \{s\} \cup \mathcal{P} \cup (\cup_{g \in \mathcal{G}} S^g) \cup (\cup_{g \in \mathcal{G}} \mathcal{L}^g) \cup \mathcal{L} \cup \{t\}$. Vertex s is the source,

further we have a vertex for each point, hence the set of vertices $\mathcal{P}$. For each color $\mathrm{g} \in \mathcal{G}$ we create a vertex for each center in S and for each label in $\mathcal{L}$, these vertices constitute the sets $\cup_{\mathrm{g}\in\mathcal{G}} S^{\mathrm{g}}$ and $\cup_{\mathrm{g}\in\mathcal{G}} \mathcal{L}^{\mathrm{g}}$, respectively. We also have a vertex for each label in $\mathcal{L}$ and finally the sink t.

- **Edges:** the set of edges is $E = E_{s\to\mathcal{P}} \cup E_{\mathcal{P}\to S^{\mathrm{g}}} \cup E_{S^{\mathrm{g}}\to\mathcal{L}^{\mathrm{g}}} \cup E_{\mathcal{L}^{\mathrm{g}}\to\mathcal{L}} \cup E_{\mathcal{L}\to t}$. $E_{s\to\mathcal{P}}$ consists of edges from the source s to every point $j \in \mathcal{P}$, $E_{\mathcal{P}\to S^{\mathrm{g}}}$ consists of edges from every point $j \in \mathcal{P}$ to the center of vertices of the same color in S^{g}, $E_{S^{\mathrm{g}}\to\mathcal{L}^{\mathrm{g}}}$ consists of edges from the colored centers to their corresponding label of the same color, $E_{\mathcal{L}^{\mathrm{g}}\to\mathcal{L}}$ consists of edges from the colored labels to their corresponding label, finally $E_{\mathcal{L}\to t}$ consists of edges from every label in $\mathcal{L}$ to the sink t.

- **Capacities:** the edges of $E_{s\to\mathcal{P}}$ have a capacity of 1, the edges of $E_{\mathcal{L}^{\mathrm{g}}\to\mathcal{L}}$ have a capacity of $\lfloor \beta_{L,g} n_L \rfloor$, the edges of $E_{\mathcal{L}\to t}$ have a capacity of n_L.

- **Demands:** the vertices of $\mathcal{L}^{\mathrm{g}}$ have a demand of $\lceil \alpha_{L,\mathrm{g}} n_L \rceil$, the vertices of $\mathcal{L}$ have a demand of n_L.

- **Costs:** all edges have a cost of zero except the edges of $E_{\mathcal{P}\to S^{\mathrm{g}}}$ where the cost of the edge between the point and the center is set according to the distance and the clustering objective (k-median or k-means). As noted earlier a vertex j will only be connected to the same color vertex that represents center i in the network flow graph, we refer to that vertex by $i^{\chi(j)}$ and clearly $i^{\chi(j)} \in S^{\chi(j)}$. Specifically, $\forall (j, i^{\chi(j)}) \in E_{\mathcal{P}\to S^{\mathrm{g}}}, \mathrm{cost}(j, i^{\chi(j)}) = d^p(j, i)$ where $p = 1$ for the k-median and $p = 2$ for the k-means.

We write the cost for a constructed flow graph as $\sum_{j\in\mathcal{P}, i\in S} d^p(j, i) x_{ij}$ where x_{ij} is the amount of flow between vertex j and center $i^{\chi(j)}$. Since all capacities, demands, and costs are set to integer values. Therefore we can obtain an optimal solution (maximum flow at a minimum cost) in polynomial time where all flow values are integers. Therefore, we can solve LCAL optimally for a given distribution

of points.

The above construction are for the k-median and k-means. For the k-center we slightly modify the graph. First, we point out that unlike the k-median and k-means, for the k-center the objective value has only a polynomial set of possibilities (kn many exactly) since it is the distance between a center and a vertex. So our network flow diagram is identical but instead of setting a cost value for the edges in edges of $E_{\mathcal{P}\to S^{\mathrm{g}}}$, we instead pick a value d from the set of possible distances $d(j,i)$ where $j \in \mathcal{P}, i \in S$ and draw an edge between a point j and a center $i^{\chi(j)}$ only if $d(j,i) \leq d$. Also we do not need to solve the minimum cost max flow problem, instead the max flow problem is sufficient. □

6.5.2 Efficient Algorithms for LCAL for the Two Label Case

For the k-median and k-means and the two label case we present an algorithm with $O(n(\log(n)+k))$ running-time. The intuition behind our algorithm is best understood for the case with "exact population proportions" for both the positive and negative labels[4]. First, we note that each color $\mathrm{g} \in \mathcal{G}$ exists in proportion $r_{\mathrm{g}} = \frac{|\mathcal{P}_h|}{|\mathcal{P}|}$ where we refer to r_{g} as the population proportion. The case of exact population proportions for the positive and negative labels, is the one where $\forall\, \mathrm{g} \in \mathcal{G}, \forall L \in \{P, N\}$: $\alpha_{L,g} = \beta_{L,g} = r_{\mathrm{g}} = \frac{|\mathcal{P}_h|}{|\mathcal{P}|}$

That is, the upper and lower proportion bounds coincide and are equal to the proportion of the color in the entire set. This forces only a limited set of possibilities for the total number of points (and their colors) which we can assign to either P or N. For example, if we have two colors and $r_1 = r_2 = \frac{1}{2}$, then we can only assign an equal number of red and blue points to P and likewise to N. For the case of three colors with $r_1 = \frac{1}{3}, r_2 = \frac{1}{2}, r_3 = \frac{1}{6}$, then we can only assign points of the following form across the different labels: points for the first color $= 2c$, points for the second color $=$

[4]The general case is shown in the appendix.

$3c$, points for the third color $= c$ where c is a non-negative integer. We refer to this smallest "atomic" number of points by n_{atomic} and the number of color h of its subset by n^h_{atomic}.

Now we define some notation $P(j) = \min_{i \in P} d(j,i)$ and $N(j) = \min_{i \in N} d(j,i)$, i.e. the distance of the closest centers to j in P and N, respectively. Further, $\phi^{-1}(P)$ and $\phi^{-1}(N)$ are the set of points assigned to the positive and negative centers by the assignment ϕ, respectively. We can now define the drop of a point j as $drop(j) = N(j) - P(j)$, clearly the larger $drop(j)$ the higher the cost goes down as we move it from the negative to the positive set. We can obtain a sorted values of $drop$ for each color in $O\big(n(\log n + k)\big)$ run-time.

The algorithm is shown (algorithm block (7)). In the first step we start with all points in N, then in step 2 we move the minimum number of n^h_{atomic} for each color h to satisfy the size bounds for each label (constraint (6.3)). Finally in the loop starting at step 3, we move more points to the positive label (in an "atomic" manner) if it lowers the cost and is within the size bounds.

Algorithm 7 Exact Preservation for k-median / k-means

1: Find an assignment ϕ_0 that assigns all points to their nearest center in N, this means that $|\phi_0^{-1}(N)| = n$ and $|\phi_0^{-1}(P)| = 0$. Set $\phi^* = \phi_0$.
2: Move $q_h = r_h \max\{(LB)_P, n - (UB)_N\}$ many points of color h with the highest values in $drop$ from the negative label to the positive label
3: **for** $i = \left(\frac{n}{\sum_{g \in \mathcal{G}} q_g}\right)$ to $\frac{n}{n_{\text{atomic}}}$ **do**
4: Take n^h_{atomic} many points from each color h with the highest values in $drop$, call the new assignment ϕ'.
5: **if** $\phi'^{-1}(P)$ and $\phi'^{-1}(N)$ are within bounds **and** $cost(\phi') < cost(\phi^*)$ **then**
6: update the assignment to $\phi^* = \phi'$
7: **else**
8: break
9: **end if**
10: **end for**

Theorem 6.5.2. *Algorithm (7) finds the optimal solution and runs in* $O\Big(n(\log n + k)\Big)$ *time.*

Proof. First we prove that the solution is feasible. Constraint (6.2) for the color proportionality holds,

this can is clearly the case before the start of the loop since the centers with negative labels cover the entire set which is color proportional and the the centers with positive labels cover cover nothing which is also color proportional. In each iteration, we move an atomic number of each color from the negative to the positive label and hence both the negative and the positive set of centers satisfy color proportionality in the points they cover.

For constraint (6.2) because of exact preservation of the color proportions, we can always tighten the bounds $(LB)_L$ and $(UB)_L$ for each label L such that there multiples of n_{atomic} without modification to the problem, so we assume that $(LB)_N = a\, n_{\text{atomic}}, (LB)_P = b\, n_{\text{atomic}}, (UB)_N = a'\, n_{\text{atomic}}, (UB)_P = b'\, n_{\text{atomic}}$ where a, a', b, b' are non-negative integers and clearly $a \leq b$ and $a' \leq b'$. Step 2 satisfies the lower bound on the number of points in the positive label and the upper bound for the negative set. Note that if this step fails then the problem has infeasible constraints. Further, since we have moved the minimum number of points from the negative set to the positive set, it follows that the upper bounds on the positive are also satisfied since $(LB)_P \leq (UB)_P$, also the lower bound on the negative set is also satisfied since $(LB)_N \leq (UB)_N$. Finally in step 5, the size bounds are always checked fair therefore both labels are balanced.

Optimally follows since we move the points with the highest $drop$ value to the positive set (these are also the points closest to the positive set). Further, in step 5 we stop moving any points to the positive if there isn't a reduction in the clustering cost. Note that since the values in $drop$ are sorted, another iteration would not reduce the cost.

Finding the closest center of each label for every point takes $O(nk)$ time. Finding and sorting the values in $drop$ clearly takes $O(n \log n)$ time. The algorithm does constant work in each iteration for at most n many iterations. Thus, the run time is $O\big(n(\log n + k)\big)$. □

With more elaborate conditional statements, the above algorithms can be generalized to give all solution values for arbitrary choices of label size bounds (constraint(6.3)) with the same asymptotic run-time. Such a solution would be useful as it would enable the decision maker to see the complete trade-off between the label sizes and the clustering cost (quality).

6.6 Algorithms and Theoretical Guarantees for LCUL

6.6.1 Computational Hardness of LCUL

We start by discussing the hardness of LCUL. In contrast to LCAL, the LCUL problem it not solvable in polynomial time. In the fact, the following theorem shows that even if we were to drop one constraint for the LCUL (problem (6.4)) we would still have an NP-hard problem.

We note that all of our hardness results use the k-center problem for simplicity. Before we introduce the hardness result, we note all of our reductions are from exact cover by 3-sets (X3C) [165] where we have universe $\mathcal{U} = \{u_1, u_2, \ldots, u_{3q}\}$ and subsets $\mathcal{W}_1, \ldots, \mathcal{W}_t$ where $t = q + r$ and for non-trivial instances $r > 0$. We form an instance of LCUL by representing each one the subsets $\mathcal{W}_1, \ldots, \mathcal{W}_t$ by a vertex and each element in $\mathcal{U} = \{u_1, u_2, \ldots, u_{3q}\}$ by a vertex. The centers are the sets $\mathcal{W}_1, \ldots, \mathcal{W}_t$ and they are given a blue color whereas the rest of the points (in $\mathcal{U}$) are red. Further, each point u_i is connected by a edge to a center $\mathcal{W}_i$ if and only if $u_i \in \mathcal{W}_j$. The distances between any two points is the length of the shortest path between them. This clearly leads to a metric. This is essentially a reduction we follow in all proofs, sometimes changes are introduced and mentioned explicitly in the proofs.

Now we introduce the following theorem:

Theorem 6.6.1. *Even if the color-proportionality constraint (6.5) are ignored*[5] LCUL *is NP-hard.*

Proof. As mentioned we consider an instance of exact cover by 3-sets (X3C) with universe $\mathcal{U} = \{u_1, u_2, \ldots, u_{3q}\}$ and subsets $\mathcal{W}_1, \ldots, \mathcal{W}_t$. We construct an instance of LCUL where the proportionality constraints are ignored. Further, we only have two labels $\mathcal{L} = \{N, P\}$, we set $(CL)_P = 0, (CU)_P = q$,$(CL)_N = 0, (CU)_N = t$ and $(LB)_P = 4q, (UB)_P = 3q + t$, $(LB)_N = 0, (UB)_N = 3q + t$.

A solution for X3C leads to a solution for LCUL at cost 1: Take the collection of q many subsets that solve X3C and give their corresponding centers in LCAL a positive label. Then it is clear that $|S^P| = q$ and that the number of points covered by the positive centers is $4q$ and that this done at a cost of 1. The centers that do not correspond to the solution of X3C will be given a negative label and assigned no points.

A solution for LCUL at cost 1 leads to a solution X3C: A solution for LCUL cannot assign more than $(CU)_P = q$ many centers a positive label and it has to cover $3q$ more points to have a total of $4q$ points and this has to be done at a distance of 1. By construction, since each center is connected to 3 points, the LCUL solution cannot have less than q centers. Further, to have $4q$ points, then each center would have to cover a unique set of 3 points at a distance of 1. Since points are connected to centers at a distance of 1 only if they are corresponding values are contained in the subsets corresponding to those centers, it follows that the q subsets in the LCUL solution are indeed an exact cover for X3C.

□

Here we instead we ignore the constraints on the number of points a label should receive, i.e. constraints (6.6 and keep the proportionality constraints. We show that this also results in an NP-hard

[5]We can simply remove the constraint or set $\alpha_{L,\mathrm{g}} = 0, \beta_{L,\mathrm{g}} = 1, \forall\, \mathrm{g} \in \mathcal{G}, L \in \mathcal{L}$.

problem as demonstrated in the theorem below:

Theorem 6.6.2. *Even if we do not specify the number of points a label should receive (constraint(6.6)),* LCUL *is NP-hard.*

Proof. Similar to the proof of theorem (6.6.1) we follow the reduction from X3C with two labels for LCUL, i.e. $\mathcal{L} = \{N, P\}$, but now we consider the color of the vertices. Vertices of the subsets $\mathcal{W}_1, \ldots, \mathcal{W}_t$ are blue and all of the vertices of the elements of $\mathcal{U}$ are red. For the LCUL instance, we set $(CL)_P = q, (CU)_P = t$,$(CL)_N = 0, (CU)_N = t$. The representation for the negative set is ignored, i.e. $\alpha_{N,\text{red}} = \alpha_{N,\text{blue}} = 0$ and $\beta_{N,\text{red}} = \beta_{N,\text{blue}} = 1$. For the positive set, we only have set a bound on the lower proportion for the red color, specifically $\alpha_{P,\text{red}} = \frac{3}{4}, \beta_{P,\text{red}} = 1$ and $\alpha_{P,\text{blue}} = 0, \beta_{P,\text{blue}} = 1$. As the reduction of theorem (6.6.1) the optimal value of the k-center objective cannot be less than 1.

A solution for X3C leads to a solution for LCUL at cost 1: Take the q subsets in the solution of X3C and assign their corresponding centers a positive labels, then $|S^P| = q \geq (CU)_P$. Further since elements of $\mathcal{U}$ are represented by red vertices, you will have $3q$ red vertices covered at a distance of 1, the red proportion of the positive label would be $\frac{3q}{4q} = \frac{3}{4} \geq \alpha_{P,\text{red}}$. To complete the solution assign the rest of the centers a negative label.

A solution for LCUL at cost 1 leads to a solution X3C: A solution for LCUL would have to choose at least $(CL)_P = q$ many centers. Since all centers are blue and because there are only $3q$ many red points in the graph, we would have to choose exactly q centers and cover all of the $3q$ many red points to satisfy the color proportionality constraints of $\alpha_{P,\text{red}}$. Since this is being done at a cost of 1, these points must be representing elements in $\mathcal{U}$ that are contained in the subsets corresponding to the selected centers. Further, since every center is connected to exactly 3 points at radius 1, we have found an exact cover. □

Theorem 6.6.3. *Even if we do not specify the number of centers of each label (ignoring constraints (6.7)),* LCUL *is NP-hard.*

Proof. Similar to theorems (6.6.1,6.6.2) we follow the same reduction from X3C. This time we ignore constraint (6.7) on the number of centers, i.e. $0 \leq |S^N|, |S^P| \leq k$. We set $(LB)_P = (UB)_P = 4q$ and $(LB)_N = 0, (LB)_N = n$. Further for the color proportionality constraints, we have for the positive set we set $\alpha_{P,\text{red}} = \beta_{P,\text{red}} = \frac{3}{4}$, $\alpha_{P,\text{blue}} = \beta_{P,\text{blue}} = \frac{1}{4}$ and for the negative set we have $\alpha_{N,\text{red}} = \alpha_{N,\text{blue}} = 0$. $\beta_{N,\text{red}} = \beta_{N,\text{blue}} = 1$.

A solution for X3C leads to a solution for LCUL at cost 1: Simply let the subsets (centers) in the solution if X3C have a positive label and assign all of the points in $\mathcal{U}$ to them. Clearly, we have $(LB)_P = (UB)_P = 4q$ and the red color has a representation of $\frac{3}{4}$ and the blue has a representation of $\frac{1}{4}$. Furthe, this is done at an optimal cost of 1.

A solution for LCUL at cost 1 leads to a solution X3C: Since $(LB)_P = (UB)_P = 4q$, $\alpha_{P,\text{red}} = \beta_{P,\text{red}} = \frac{3}{4}$, and $\alpha_{P,\text{blue}} = \beta_{P,\text{blue}} = \frac{1}{4}$, it follows that the positive set should cover $\frac{3}{4}4q = 3q$ many red points and that it must also cover $\frac{1}{4}4q = q$ many blue points. Since all blue points are centers and all red points are from $\mathcal{U}$,it follows that we have to choose q many centers to cover $3q$ many points at an optimal cost of 1. This leads to a solution for X3C. □

Theorem 6.6.4. *For the* LCUL *problem with two labels and two colors, dropping one of the constraints(6.5), (6.6), or (6.7) still leads to an NP-hard problem.*

Proof. This follows immediately from theorems (6.6.1,6.6.2,6.6.3) above. □

Having established the hardness of LCUL for different sets of constraints, we show that it is fixed-parameter tractable[6] for a constant number of labels. This immediately follows since a given

[6]An algorithm is called fixed-parameter tractable if its run-time is $O(f(k)n^c)$ where $f(k)$ can be exponential in k, see

choice of labels for the centers leads to an instance of LCAL which is solvable in polynomial time and there are at most m^k many possible choice labels.

Theorem 6.6.5. *The* LCUL *problem is fixed-parameter tractable with respect to* k *for a constant number of labels.*

Proof. This follows simply by noting that if the labels are assigned, then we have an LCAL instance which solvable in time that is polynomial in n and k, since $k \leq n$, it follows that the run time for solving LCAL is $O(n^c)$ for some constant c. Now, since there are at most m^k many label choices for the centers, it follows that the run time is for LCUL is $O(m^k n^c)$. □

It is also worth wondering if the problem remains hard if we were to drop two constraints and have only one instead. Interestingly, we show that even for the case where the number of labels m is super-constant ($m = \Omega(1)$) , if we only had the color-proportionality constraint (6.5) or the constraint on the number of labels (6.6), then the problem is solvable in polynomial time. However, if we only had constraint (6.7) for the number of centers a label has, the problem is still NP-hard.

Theorem 6.6.6. *Even if number of labels* $m = \Omega(n)$*, the* LCUL *problem is solvable in polynomial time under constraint (6.5) alone or constraint (6.7) alone. However, it is NP-hard under constraint (6.6) alone.*

Proof. Let us consider the color proportionality constraint (6.5) alone. To solve the problem optimally and satisfy the constraint, simply assign all points to their closest center and let all centers take one label from the set $\mathcal{L}$.

Now, we consider only the constraints on the number of centers for each label (6.7). Again we assign each point to its closest center for an optimal cost. To satisfy constraints (6.7), assuming the

[166] for more details.

constraint parameters of (6.7) lead to a feasible problem, then each label $L \in \mathcal{L}$, assign it $(CL)_L$ many centers arbitrarily. If some centers have not been assigned any labels, then simply go to label L which has not reached its upper bound $(CU)_L$ and assign more labels from it. We simply keep assigning labels from label values that have not reached their upper bound on the number of centers until all centers have a label.

Now, we consider only the constraints on the number of points a label receives (6.6). We simply follow the same reduction from theorems (6.6.1,6.6.2,6.6.3), see also the beginning of this subsection for the details of the reduction from X3C. We have $t = q + r$ many subsets, we let the number of labels of the LCUL instance be $m = t = q + r$. Further, we partition the set of labels into two, i.e. $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$ where $|\mathcal{L}_1| = q$ and $|\mathcal{L}_2| = r$, and we set the lower and upper bounds for the labels according to these sets. Specifically, $\forall L \in \mathcal{L}_1 : (LB)_L = (UB)_L = 4q$ and $\forall L \in \mathcal{L}_2 : (LB)_L = (UB)_L = 1$. Now, clearly a solution for X3C leads to a solution for the LCUL instance, we simply let the subsets (centers) in the solution of X3C be the centers for the label set $\mathcal{L}_1$. Each center is assigned a label from $\mathcal{L}_1$ and covers itself and 3 points from $\mathcal{U}$, this leads to $4q$ many points which clearly satisfies the upper and lower bounds. Further, the centers not the solution are assigned a label from $\mathcal{L}_2$ and cover themselves, which is just 1 point and therefore satisfies the constraints. Now for the reverse direction, consider the set $\mathcal{L}_2$ where we have r many labels each covering 1 point. It clear, the smallest cost would be for a center to be assigned to itself, it follows that we are looking for r many centers and that each center should only be assigned to itself. This then leaves us with q many centers, since no center can cover more than $4q$ many points at a distance of 1, and since we have q many labels with each having to cover $4q$ many points, we clearly have a set cover, i.e. a solution for X3C. □

6.6.2 A Randomized Algorithm for label proportional LCUL

Here we consider a natural special case of the LCUL problem which we call color and label proportional case (CLP) where the constraints are restricted to a specific form. In CLP each label must have color proportions "around" that of the population, i.e. color h has proportion r_h in each label $L \in \mathcal{L}$. Further, each label has a proportion $\alpha_L \in [0,1]$ and $\sum_{L\in\mathcal{L}} \alpha_L = 1$, this proportion decides the number of points the label covers and the number of centers it has i.e., label L covers around $\alpha_L n$ many points and has around $\alpha_L k$ many centers. Therefore, the optimization takes on the following form below where we have included the ϵ values to relax the constraints (note that for every value of ϵ, we have that $\epsilon \geq 0$):

$$\min_{\phi,\mathcal{X}} \Big(\sum_{j\in\mathcal{P}} d^p(j,\phi(j)) \Big)^{1/p} \tag{6.10}$$

$$\forall L \in \mathcal{L}, \forall g \in \mathcal{G} : (r_h - \epsilon^A_{g,L}) \sum_{\substack{i\in S:\\ \mathcal{X}(i)=L}} |\,\mathcal{P}_i\,| \leq \sum_{\substack{i\in S:\\ \mathcal{X}(i)=L}} |\,\mathcal{P}_{i,\mathrm{g}}\,| \leq (r_h + \epsilon'^A_{g,L}) \sum_{\substack{i\in S:\\ \mathcal{X}(i)=L}} |\,\mathcal{P}_i\,| \tag{6.11}$$

$$\forall L \in \mathcal{L} : (\alpha_L - \epsilon^B_L)n \leq \sum_{i\in S:\mathcal{X}(i)=L} |\,\mathcal{P}_i\,| \leq (\alpha_L + \epsilon'^B_L)n \tag{6.12}$$

$$\forall L \in \mathcal{L} : (\alpha_L - \epsilon^C_L)k \leq |S^L| \leq (\alpha_L + \epsilon'^C_L)k \tag{6.13}$$

We note that even when the constraints take on this specific form the problem is still NP-hard as shown in the theorem below:

Theorem 6.6.7. *The* CLP *problem is NP-hard even for the two color and two label case.*

Proof. We follow a reduction for X3C (see the beginning of the appendix). We consider the two

label case, $\mathcal{L} = \{N, P\}$. Similiar to the previous reductions we will have t many blue centers for the subsets $\mathcal{W}_1, \ldots, \mathcal{W}_t$ each being connected to its elements in $\mathcal{U}$ at a distance of 1 with all elements in $\mathcal{U}$ being red. Note that $|\mathcal{U}| = q$ and that $t = q + r$. Now we also add $2q$ many blue centers which are not connected to anything by an edge, expect for one center which is connected by an edge to a new $3(r+2q)$ many red points, this means that any one of these red points is at a distance of 1 from this new center. Note that the increase in the problem size is still polynomial in the original X3C problem. We set the color proportionality constraint so that each label should have exactly 3:1 ratio of red points to blue points. Now the total number of points in the problem is $n = 4q+r+2q+3(r+2q) = 4(3q+r)$. The number of centers $k = q+r+2q = 3q+r$. Further, we set $\alpha_P = \frac{q}{(3q+r)}$ and $\alpha_N = 1-\alpha_P = \frac{2q+r}{3q+r}$. We set the lower and upper size bounds according to α_P and α_N, this leads to $(LB)_P = (UB)_P = \alpha_P n = \frac{q}{(3q+r)}n = \frac{q}{(3q+r)}4(3q+r) = 4q$ and $(LB)_N = (UB)_N = \frac{2q+r}{3q+r}n = \frac{2q+r}{3q+r}4(3q+r) = 4(2q+r)$. Further, the number of centers for each label are $(CL)_P = (CU)_P = \alpha_P k = \frac{q}{(3q+r)}k = \frac{q}{(3q+r)}3q+r = q$ and $(CL)_N = (CU)_N = \alpha_N k = \frac{2q+r}{3q+r}3q + r = 2q + r$.

A solution for X3C leads to a solution for LCUL at cost 1: Simply let the q many centers representing the solution set in $\mathcal{W}_1, \ldots, \mathcal{W}_t$ be the positive labeled centers and assign them the points that belong to them and let all other centers be negative and assign the last new center all of the $3(r + 2q)$ many red children points. We then q many positive centers covering $4q$ many points with the color proportionality being 3:1 red points to blue points. Similarly, for the negative set we have $2q + r$ many centers covering $4(2q + r)$ many points at a color proportionality of 3:1 red to blue. This is done at cost of 1, so clearly optimal.

A solution for LCUL at cost 1 leads to a solution X3C: Suppose the new blue center with $3(r+2q)$ many red children is assigned a positive label, this to achieve an optimal cost all of its children have to be assigned to it. This means that the positive set would have at least $3(r + 2q) = 6q + 3r$

many points, but $(LB)_P = (UB)_P = \alpha_P n = 4q < 6q < 6q + 3r$ which causes a contradiction. Therefore that center can never be positive. Therefore, we are looking for $\alpha_P k = q$ many centers to cover $\alpha_P n = 4q$ many points and because of the color proportionality constraint $3q$ many of them are red and q are blue. Finding this set at an optimal cost is a solution for X3C. □

We show a randomized algorithm (algorithm block (8)) which always gives an optimal cost to the clustering and satisfies all constraints in expectation and further satisfies constraint (6.13) deterministically with a violation of at most 1. Our algorithm is follows three steps. In step 1 we find the assignment ϕ^* by assigning each point to its nearest center, thereby guaranteeing an optimal clustering cost. In step 2, we set the center-to-label probabilistic assignments $p_{i,L} = \alpha_L$. Then in step 3, we apply dependent rounding, due to Gandhi et al. [103], to the probabilistic assignments to find the deterministic assignments. This leads to the following theorem:

Theorem 6.6.8. *Algorithm 8 gives an optimal clustering and satisfies constraints (6.11,6.12,6.13) in expectation with (6.13) being satisfied deterministically at a violation at most 1.*

Proof. The optimality of the clustering cost follows immediately since each point is assigned to its closest center. Now, we show that the assignment satisfies all of the constraints. We have $p_{i,L} = \alpha_L$ for each center i. Now we prove that constraints (6.5,6.6,6.7) hold in expectation over the assignments $P_{i,L}$. Note that $P_{i,L}$ is also an indicator random variable for center i, taking label L. Then we can show that using property **(A)** of dependent rounding (marginal probability) that:

$$\mathbb{E}[\sum_{i \in S: \ell(i) = L} |\,\mathcal{P}_i\,|] = \mathbb{E}[\sum_{i \in S} |\,\mathcal{P}_i\,|\,P_{i,L}] = \sum_{i \in S} |\,\mathcal{P}_i\,|\,\mathbb{E}[P_{i,L}]$$

$$= \sum_{i \in S} |\,\mathcal{P}_i\,|\,p_{i,L} = \alpha_L \sum_{i \in S} |\,\mathcal{P}_i\,| = \alpha_L n$$

Clearly, constraint (6.12) is satisfied. Through a similar argument we can show that the rest of the constraints also hold in expectation.

We have that $\forall L \in \mathcal{L} : |S^L| = \sum_{i \in S} P_{i,L} = \sum_{i \in S} \alpha_L = \alpha_L k$. By property **(B)** of dependent rounding (degree preservation) we have $\forall L \in \mathcal{L} : |S^L| \in \{\lfloor \alpha_L k \rfloor, \lceil \alpha_L k \rceil\}$. Therefore constraint (6.13) is satisfied in every run of the algorithm at a violation of at most 1. □

Algorithm 8 Randomized LCUL Algorithm

1: Find the assignment ϕ^* by assigning each point to its nearest center in S.
2: For each center i, set its probabilistic assignment for label L to $p_{i,L} = \alpha_L$.
3: Apply dependent rounding [103] to probabilistic assignments $p_{i,L}$ to get the deterministic assignments $P_{i,L}$

We note that dependent rounding enjoys the **Marginal Probability** property which means that $\Pr[P_{i,L} = 1] = p_{i,L}$. This enables us to satisfy the constraints in expectation. While we note that letting each center i take label L with probability α_L would also satisfy the constraints in expectation. Dependent rounding also has the **Degree Preservation** property which implies that $\forall L \in \mathcal{L} : \sum_{i \in S} P_{i,L} \in \{\lfloor \sum_{i \in S} p_{i,L} \rfloor, \lceil \sum_{i \in S} p_{i,L} \rceil\}$ which leads us to satisfy constraint (6.13) deterministically (in every run of the algorithm) with a violation of at most 1. Further, dependent rounding has the **Negative Correlation** property which under some conditions leads to a concentration around the expected value. Although, we cannot theoretically guarantee that we have a concentration around the expected value, we observe empirically (section 6.8.2) that dependent rounding is much better concentrated around the expected value, especially for constraint (6.12) for the number of points in each label.

6.7 LCAL Algorithm for Two Labels and General Proportions

Our algorithm for general proportions is similar to the exact preservation algorithm. The two main differences lie in the fact that we use feasibility checks for a given value of n_P where n_P is

the number of points to be assigned to the positive label and that the way we move points from the negative-labelled centers to the positive-labelled labels is more elaborate. In particular algorithm block 9 shows our algorithm. Note that $n_{\mathrm{g}} = |\,\mathcal{P}^{\mathrm{g}}\,|$.

Algorithm 9 Non-Exact Preservation for k-median/k-means

1: Define the optimal assignment as ϕ^* and its cost as $cost^*$.
2: **Step 1:**
3: Find an assignment ϕ_0 that assigns all points to their nearest center in N, this means that $|\phi_0^{-1}(N)| = n$ and $|\phi_0^{-1}(P)| = 0$. Update ϕ^* and $cost^*$ according to the values for this solution ϕ_0.
4: **Step 2:**
5: **for** $n_P = 0$ to n **do**
6: $\quad \forall\, \mathrm{g} \in \mathcal{G}$ find $n^P_{\mathrm{g},l} = \max\left(\lceil \alpha_{P,\mathrm{g}} n_P \rceil\,, n_{\mathrm{g}} - \lfloor \beta_{N,\mathrm{g}}(n - n_P) \rfloor \right)$
7: $\quad \forall\, \mathrm{g} \in \mathcal{G}$ find $n^P_{\mathrm{g},u} = \min\left(\lfloor \beta_{P,\mathrm{g}} n_P \rfloor\,, n_{\mathrm{g}} - \lceil \alpha_{N,\mathrm{g}}(n - n_P) \rceil \right)$
8: $\quad$ Find $n_{P,l} = \sum_{\mathrm{g}\in\mathcal{G}} n^P_{\mathrm{g},l}$ and $n_{P,u} = \sum_{\mathrm{g}\in\mathcal{G}} n^P_{\mathrm{g},u}$
9: $\quad$ **if** $(\forall\, \mathrm{g} \in \mathcal{G} : n^P_{\mathrm{g},u} \geq n^P_{\mathrm{g},l})$ **and** $(n_{P,u} \geq n_P \geq n_{P,l})$ **then**
10: $\quad\quad \forall\, \mathrm{g} \in \mathcal{G}$ move as many points with the maximum drop such that there at least $n^P_{\mathrm{g},l}$ points of color g in the positive set
11: $\quad\quad$ **if** $n_P > n_{P,l}$ **then**
12: $\quad\quad\quad$ Move $\left(n_P - n_{P,l}\right)$ many points to the positive set, each time selecting the point with the maximum drop provided the total number in the positive set of its color g does not exceed $n^P_{\mathrm{g},u}$.
13: $\quad\quad\quad$ For each point j moved to the positive set, record $N(j)$, $P(j)$, and the iteration of movement $t_j = i$.
14: $\quad\quad$ **end if**
15: $\quad\quad$ Record the cost in entry $cost(i) = cost_i$.
16: $\quad\quad$ **if** $cost_i < cost^*$ **then**
17: $\quad\quad\quad$ Let $\phi^* = \phi_i$ and $cost^* = cost_i$
18: $\quad\quad$ **end if**
19: $\quad$ **else**
20: $\quad\quad$ Move to line 5.
21: $\quad$ **end if**
22: **end for**
23: **return** $\phi^*, cost^*$

Notice how we calculate $n^P_{\mathrm{g},l}$ and $n^P_{\mathrm{g},u}$ in each iteration, these are the lower and upper bounds for the number of points of color g that should be in the positive set for a given value of n_P. It is not difficult to see that a violation of these bounds would cause a problem in the proportion representation

in either the positive of negative set and that $n_{P,u} \geq n_P \geq n_{P,l}$. This is proved in the next lemma:

Lemma 6.7.1. *If the value of n_P satisfies: (1) $\forall\, \mathrm{g} \in \mathcal{G} : n^P_{\mathrm{g},u} \geq n^P_{\mathrm{g},l}$. (2) $n_{P,u} \geq n_P \geq n_{P,l}$. Then n_P is feasible.*

Proof. For (1): if we have $\forall\, \mathrm{g} \in \mathcal{G} : n^P_{\mathrm{g},u} \geq n^P_{\mathrm{g},l}$, then we can have $n_{P,h}$ points such that $n^P_{\mathrm{g},u} \geq n_{P,h} \geq n^P_{\mathrm{g},l}$. It follows by the values of $n^P_{\mathrm{g},l}$ and $n^P_{\mathrm{g},u}$ that $\forall\, \mathrm{g} \in \mathcal{G} : \lfloor \beta_{P,\mathrm{g}} n_P \rfloor \geq n_{P,h} \geq \lceil \alpha_{P,\mathrm{g}} n_P \rceil$ which means that the positive set is indeed feasible in terms of proportions.

For the negative set, we would have $n_N = n - n_P$ many points in the negative set as well as $n_{N,h} = n_{\mathrm{g}} - n_{P,h}$ many points of color g in the negative set. It follows, that $n_{N,h} = n_{\mathrm{g}} - n_{P,h} \geq n_{\mathrm{g}} - \min\Big(\lfloor \beta_{P,\mathrm{g}} n_P \rfloor, n_{\mathrm{g}} - \lceil \alpha_{N,\mathrm{g}}(n - n_P) \rceil \Big) \geq n_{\mathrm{g}} - \big(n_h - \lceil \alpha_{N,\mathrm{g}}(n - n_P) \rceil \big) \geq \lceil \alpha_{N,\mathrm{g}}(n - n_P) \rceil \geq \lceil \alpha_{N,\mathrm{g}} n_N \rceil$. By similar arguments we can show that $n_{N,h} \leq \lfloor \beta_{N,\mathrm{g}} n_N \rfloor$ which means that the color proportions are balanced for the negative set as well.

For (2): It is immediate to see that this is feasible if $n_{P,u} = n_P = n_{P,l}$, if on the other hand we have an inequality on one side, then we can also have a total of n_P by moving points for each color within its upper and lower bounds ($n^P_{\mathrm{g},l}$ and $n_{P,u}$) until we have a total of n_P many points. □

Theorem 6.7.2. *Algorithm (9) returns an optimal solution.*

Proof. It is clear that when all points are assigned to the negative label the solution is optimal for that value of n_P (although possibly not feasible). As we iterate through the values of n_P until we find a feasible n_P our method results in feasible solution as proved in lemma (6.7.1) and it also leads to an optimal solution for that value of n_P. To prove the second statement follow a similar argument to the proof of theorem (6.5.2).

Specifically, suppose that we are iteration n_P and that the last time the solution was updated [7] was

[7] Note that the solution at step $(n_P - 1)$ may not be updated since the number of points n_P assigned to the positive set of centers may not be feasible.

at iteration $n'_P < n_P$. Assuming our solution at step n'_P is optimal we wish to prove that the solution for step n_P is also optimal. Let ϕ' and ϕ denote the assignments for steps n'_P and n_P, respectively. As the assignment changes from ϕ' to ϕ, we can put the points in 4 sets: $\mathcal{P}_{N\to N}, \mathcal{P}_{P\to P}, \mathcal{P}_{N\to P}, \mathcal{P}_{P\to N}$. The first two sets remain assigned to the same label, whereas the last two change labels. Since the first two set of points do not change labels, they are assigned to the same centers as they were in ϕ' since that is their closest center in N or P. Since at iteration n_P, we should have n_P many points assigned to the positive set and at iteration n'_P we had n'_P many points assigned, then $|\,\mathcal{P}_{N\to P}\,| = |\,\mathcal{P}_{P\to N}\,| + (n_P - n'_P)$. Further, let n'_h denote the number of points of color h assigned to the positive label by the assignment ϕ'.

Sort the set of points in $\mathcal{P}_{N\to P}$ descendingly according to their drop value, take from each color $n^P_{h,l} - n'_h$ many points with the maximum drop. Further, if this set does not have a size of n_P, then choose more points from each color provided their upper bound has not be reached, each time picking the ones with the maximum drop, let $\mathcal{P}^*_{N\to P}$ denote that resulting set of points, and $\mathcal{P}^{*\,\mathrm{g}}_{N\to P}$ the subset of $\mathcal{P}^*_{N\to P}$ with color g. It follows that the change in the cost is:

$$\underbrace{\sum_{j\in\mathcal{P}^{*\,\mathrm{g}_1}_{N\to P}}\Big(N(j)-P(j)\Big)+\cdots+\sum_{j\in\mathcal{P}^{*\,\mathrm{g}_\ell}_{N\to P}}\Big(N(j)-P(j)\Big)+}_{A} \tag{6.14}$$

$$\underbrace{\sum_{j\in\mathcal{P}_{N\to P}-\mathcal{P}^*_{N\to P}}\Big(N(j)-P(j)\Big)+\sum_{j\in\mathcal{P}_{P\to N}}\Big(N(j)-P(j)\Big)}_{B} \tag{6.15}$$

if having $|\,\mathcal{P}_{N\to P}\,| = (n_P - n'_P)$ and $|\,\mathcal{P}_{P\to N}\,| = 0$ does not achieve the optimal solution and instead we need $|\,\mathcal{P}_{P\to N}\,| = t > 0$ and $|\,\mathcal{P}_{N\to P}\,| = t + (n_P - n'_P)$, then it must be the case that $B < 0$ but that would imply that we can achieve a better solution by interchanging $|\,\mathcal{P}_{P\to N}\,|$ many points[8] from

[8]Note that $|\,\mathcal{P}_{N\to P} - \mathcal{P}^*_{N\to P}\,| = |\,\mathcal{P}_{P\to N}\,|$

the positive set to the negative, this implies that the assignment of ϕ' is not optimal which contradicts the inductive hypothesis.

Now since we find the optimal value for each feasible n_P, we indeed can find the optimal value by the finding the minimum of those values. □

We note that although the algorithm would find the optimal solution if it exists, the pre-set proportion bounds may lead to an infeasible problem. In that case, our algorithm would terminate without finding a solution. Further, it is not difficult to generalize the algorithm to the case of the k-center.

Theorem 6.7.3. *For the two label case $\mathcal{L} = \{N, P\}$ and n many points. Using algorithm (9) we can obtain the optimal cost and solution for all possible distribution of points among the positive P and negative N labels in $O(n(\log n + k))$ and the memory required to save the costs and solutions is $O(n \log n)$.*

Proof. Follows the same argument as that of theorem (6.5.2). There are clearly n many possible solution values and each point may be assigned to $k \leq n$ many possible centers so we need at most $O(n \log n)$ memory. □

6.8 Experiments

We run our algorithms using commodity hardware with our code written in Python 3.6 using the `NumPy` library and functions from the `Scikit-learn` library [167]. We evaluate the performance of our algorithms over a collection of datasets from the UCI repository [168]. For all datasets, we choose specific attributes for group membership and use numeric attributes as coordinates with the Euclidean distance measure. Through all experiments for a color $\mathrm{g} \in \mathcal{G}$ with population proportion

$r_{\mathrm{g}} = \frac{|\mathcal{P}_{\mathrm{g}}|}{|\mathcal{P}|}$ we set the the upper and lower proportion bounds to $\alpha_{\mathrm{g}} = (1-\delta)r_{\mathrm{g}}$ and $\beta_{\mathrm{g}} = (1+\delta)r_{\mathrm{g}}$, respectively. Note that the upper and lower proportion bounds are the same for both labels. Further, we have $\delta \in [0, 1]$, and smaller values correspond to more stringent constraints. In our experiments, we set δ to 0.1. For both the LCAL and LCUL we measure the price of fairness as follows:

$$\mathrm{PoF} = \frac{\text{fair solution cost}}{\text{color-blind solution cost}}$$

where fair solution cost is the cost of the fair variant and color-blind solution cost is the cost of the *unfair* algorithm which would assign each point to its closest center.

We note that since all constraints are proportionality constraints, we calculate the proportional violation. To be precise, for the color proportionality constraint (6.5), we consider a label L and define $\Delta_{L,g} \in [0, 1]$ where $\Delta_{L,g}$ is the smallest relaxation of the constraint for which the constraint is satisfied, i.e. the minimum value for which the following constraint is feasible given the solution:

$$(\alpha_{L,\mathrm{g}} - \Delta_{L,g}) \sum_{i \in S : \mathcal{X}(i) = L} |\,\mathcal{P}_i\,| \leq \sum_{i \in S : \mathcal{X}(i) = L} |\,\mathcal{P}_{i,\mathrm{g}}\,| \leq (\beta_{L,\mathrm{g}} + \Delta_{L,g}) \sum_{i \in S : \mathcal{X}(i) = L} |\,\mathcal{P}_i\,|$$

having found $\Delta_{L,g}$ we report Δ_{color} where $\Delta_{\text{color}} = \max_{\{g \in \mathcal{G}, L \in \mathcal{L}\}} \Delta_{L,g}$. Similarly, we define the proportional violation for the number of points $\Delta^{L}_{\text{points/label}}$ assigned to a label as the minimal relaxation of the constraint for it to be satisfied. We set $\Delta_{\text{points/label}}$ to the maximum across the two labels. In a similar manner, we define $\Delta_{\text{center/label}}$ for the number of centers a label receives.

We use the k-means++ algorithm [169] to open a set of k centers. These centers are inspected and assigned a label. Further, this set of centers and its assigned labels are fixed when comparing to baselines other than our algorithm.

Clustering Baseline: In the labeled setting and in the absence of our algorithm, the only alternative that would result in. a fair outcome is a fair clustering algorithm. Therefore we compare against fair clustering algorithms. The literature in fair clustering is vast, we choose the work of [8] as it can be tailored easily to this setting in which the centers are open. Further, it allows both lower and upper proportion bounds in arbitrary metric spaces and results in fair solutions at relatively small values of PoF compared to larger PoF (as high as 7) reported in [7]. Our primary concern here is not to compare to all fair clustering work, but gauge the performance of these algorithms in this setting. We also compare against the "unfair" solution that would simply assign each point to its closest center which we call the nearest center baseline. Though this in general would violate the fairness constraints it would result in the minimum cost.

Datasets: We use two datasets from the UCI repository: The **Adult** dataset consisting of $32{,}561$ points and the **Credit** dataset consisting of $30{,}000$ points. For the group membership attribute we use race for **Adult** which takes on 5 possible values (5 colors) and marriage for **Credit** which takes on 4 possible values (4 colors). For the **Adult** dataset we use the numeric entries of the dataset (age, final-weight, education, capital gain, and hours worked per week) as coordinates in the space. Whereas for the **Credit** dataset we use age and 12 other financial entries as coordinates.

6.8.1 LCAL Experiments

Adult Dataset: After obtaining k centers using the k-means++ algorithm, we inspect the resulting centers. In an advertising setting, it is reasonable to think that advertisements for expensive items could be targeting individuals who obtained a high capital gain. Therefore, we choose centers high in the capital gain coordinate to be positive (assign an advertisement for an expensive item). Specifically,

centers whose capital gain coordinate is greater than $1{,}100$ receive a positive label and the remaining centers are assigned a negative one. Such a choice is somewhat arbitrary, but suffices to demonstrate the effectiveness of our algorithm. In real world scenarios, we expect the process to be significantly more elaborate with more representative features available. We run our algorithm for LCAL as well as the fair clustering algorithm as a baseline. Figure 6.1 shows the results. It is clear that our algorithm leads to a much smaller PoF and the PoF is more robust to variations in the number of clusters. In fact, our algorithm can lead to a PoF as small as 1.0059 (0.59%) and very close to the unfair nearest center baseline whereas fair clustering would have a PoF as large as 1.7 (70%). Further, we also see that the unlike the nearest center baseline, fair labeled clustering has no proportional violations just like fair clustering.

Here for the LCAL setting, we compare to the optimal (fairness-agnostic) solution where each point is simply routed to its closest center regardless of color or label. We use the same setting at that from section 6.8. We set $\delta = 0.1$ and measure the PoF. Since the (fairness-agnostic) solution does not consider the fairness constraint we also measure its proportional violations. Figures 6 and 7 show the results over the **Adult** and **Credit** datasets. We can clearly see that although the (fairness-agnostic) solution has the smallest cost it has large color violation. We also see that our algorithm unlike fair clustering achieves fairness but at a much lower PoF.

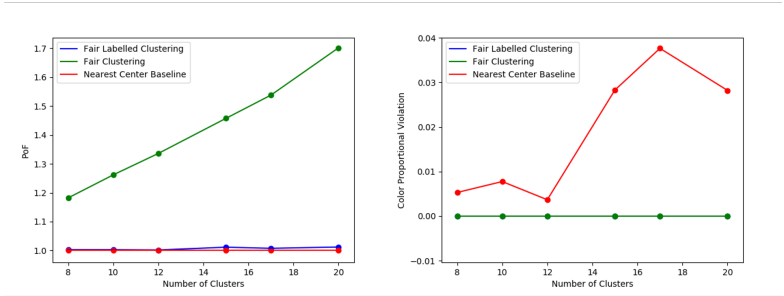
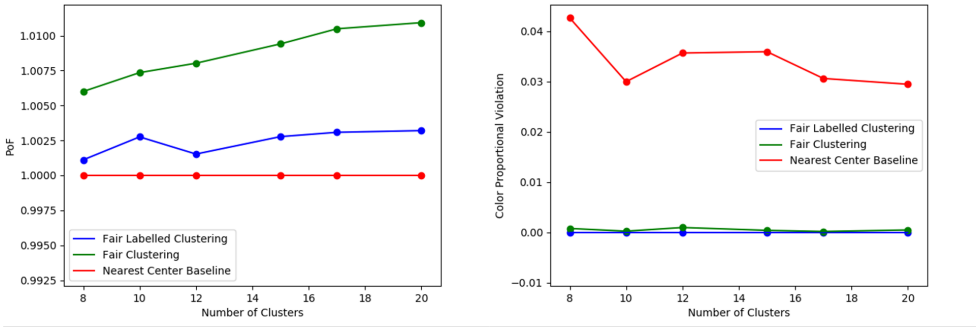


Figure 6.1: **Adult** dataset results (a):PoF, (b):Δ_{color}

Credit Dataset: Similar to the **Adult** dataset experiment, after finding the centers using k-means++, we assign them positive and negative labels. For similar motivations, if the center has a coordinate corresponding to the amount of balance that is greater than $300{,}000$ we assign the center a positive label and a negative one otherwise. Figure 6.2 shows the results of the experiments. We see again that our algorithm leads to a lower price of fairness than fair clustering, but not to the same extent as in the **Adult** dataset but it still has no proportional violation just like fair clustering.



(a): PoF (b): Color Proportional Violation

Figure 6.2: **Credit** dataset results (a):PoF, (b):Δ_{color}

As mentioned in section 6.5.2, algorithm (7) can allow the user to obtain the solutions for different values of $|\phi^{-1}(P)|$ (the number of points assigned to the positive set) without an asymptotic increase in the running time. In figure 6.3 we show a plot of $|\phi^{-1}(P)|$ vs the clustering cost. Interestingly, requiring more points to be assigned to the positive label comes at the expense of a larger cost for some instances (**Adult** with $k = 15$) whereas for others it has a non-monotonic behaviour (**Adult** with $k = 10$). This can perhaps be explained by the different choices of centers as k varies. There are 5 centers with positive labels for $k = 10$ (50% of the total), but only 4 for $k = 15$ (less than 30%) making it difficult to route points to positive centers.

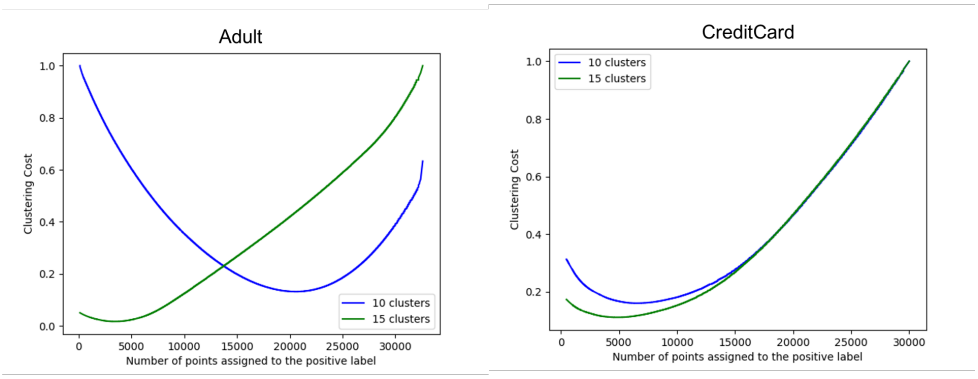


Figure 6.3: A plot of $|\phi^{-1}(P)|$ vs the clustering cost (normalized by the maximum cost obtained).

6.8.2 LCUL Experiments

Similar to the LCAL setting for LCUL we get the centers by running k-means++. However, we do not have the labels. We compare our algorithm (algorithm 8) to two baselines: (1) Nearest Center with Random Assignment (**NCRA**) and (2) Fair Clustering (**FC**). We refer to our algorithm (block 8) as **LFC** (labeled fair clustering). In **NCRA** we assign each point to its closest center which leads to an optimal clustering cost, whereas for fair clustering (**FC**) we solve the fair clustering problem. For both **NCRA** and **FC** we assign each center label L with probability α_L.

We use two labels with $\alpha_1 = \frac{1}{4}$ and $\alpha_2 = \frac{3}{4}$. For all colors and labels we set $\epsilon^A_{h,L} = \epsilon'^A_{h,L} = 0.2$ and for all labels we set $\epsilon^B_L = \epsilon'^B_L = \epsilon^C_L = \epsilon'^C_L = 0.1$. Further, all algorithms satisfy the constraints in expectation, therefore we seek a measure of centrality around the expectation like the variance. Each algorithm is ran 50 times and we report the average values of Δ_{color},$\Delta_{\text{points/label}}$, and $\Delta_{\text{center/label}}$.

Figures 6.4 and 6.5 show the results for **Adult** and **Credit**. For PoF, our algorithm achieves an optimal clustering and hence coincides with **NCRA** whereas fair clustering achieves a much higher PoF as large as 1.5. For the color proportionality (Δ_{color}), we see that fair clustering has almost no violation whereas the **NCRA** and labeled clustering have small but noticeable violations. For the

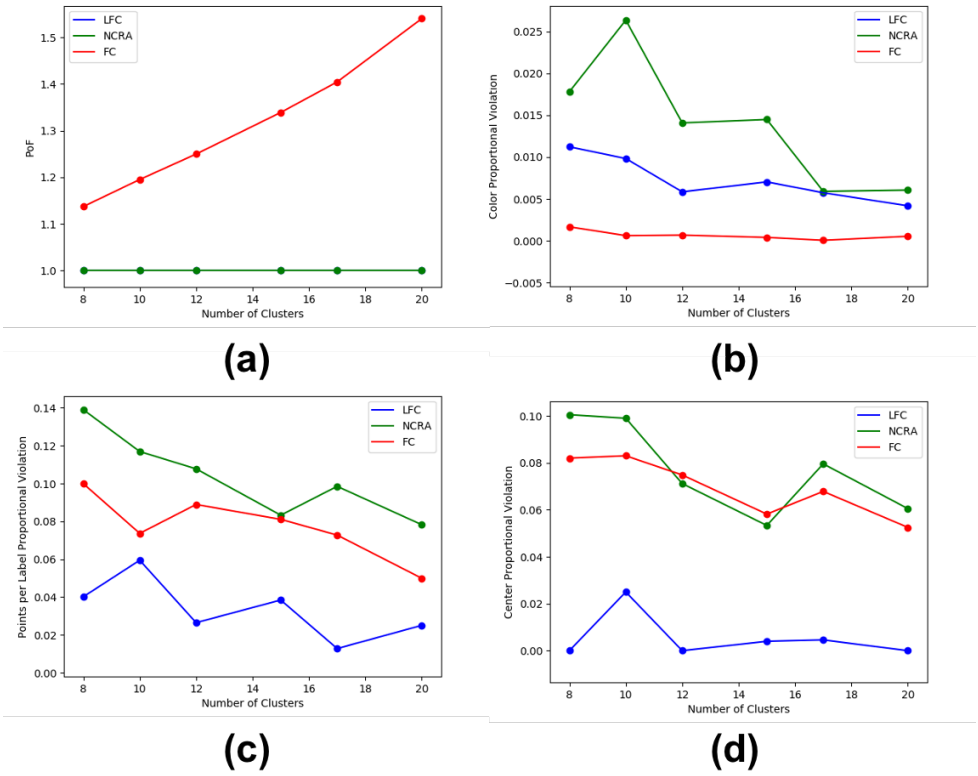


Figure 6.4: LCUL results on the **Adult** dataset. (a):PoF, (b):Δ_{color}, (c):$\Delta_{\text{points/label}}$,(d):$\Delta_{\text{center/label}}$.

number of points a label receives ($\Delta_{\text{points/label}}$) we notice that all algorithms have a violation although labeled clustering has a smaller violation mostly. As noted earlier, we suspect that this is a result of dependent rounding's negative correlation property leading to some concentration around the expectation. Finally, for the number of centers a label receives ($\Delta_{\text{center/label}}$), clearly **LFC** has a much lower violation.

6.8.3 Algorithm Scalability

Here we investigate the scalability of our algorithms. In particular, we take the **Census1990** dataset which consists of 2,458,285 points and sub-sample it to a specific number, each time we find

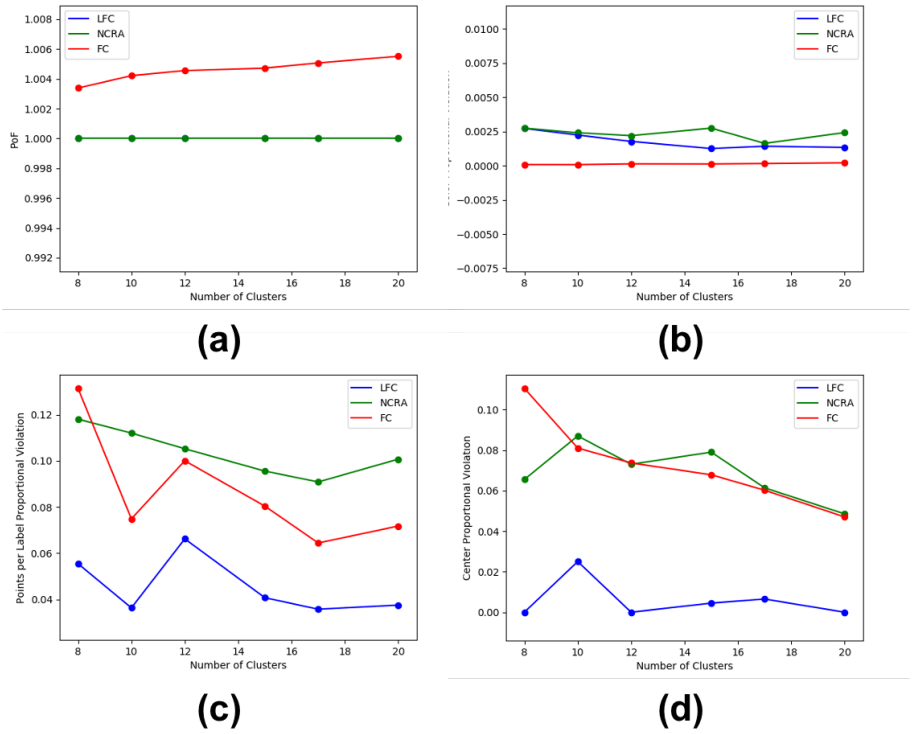


Figure 6.5: LCUL results on the **Credit** dataset.
(a):PoF, (b):Δ_{color}, (c):$\Delta_{\text{points/label}}$, and (d):$\Delta_{\text{center/label}}$.

the centers with the k-means algorithm[9], assign them random labels, and solve the LCAL and LCUL problems. Note since we care only about the run-time a random assignment of labels should suffice. Our group membership attribute is gender which has two values (two colors). We find our algorithm are indeed highly scalable (figure 6.6) and that even for 500,000 points it takes less than 90 seconds. We note in contrast that the fair clustering algorithm of [8] would takes around 30 minutes to solve a similar size on the same dataset. In fact, scalability is an issue in fair clustering and it has instigated a collection of work such as [142, 143]. The fact that our algorithm performs relatively well run-time wise is worthy of noting.

[9]We choose $k = 5$ for all different dataset sizes.

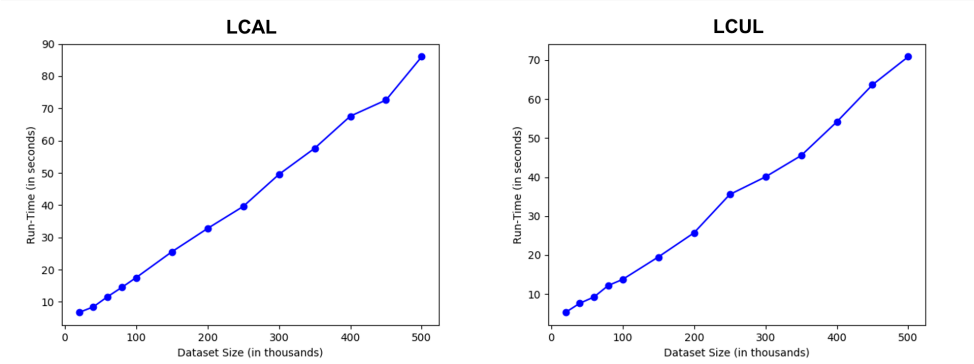


Figure 6.6: Dataset size vs algorithm Run-Time: (left) LCAL, (right) LCUL.

Chapter 7: Robust Fair Clustering with Group Mmebership Uncertainty Sets

This chapter is largely based on our paper [60].

7.1 Introduction

Machine learning and algorithmic-based decision-making systems have seen a remarkable proliferation in the last few decades. These systems are used in financial crime detection [170, 171], loan approval [172, 173], automated hiring systems [174, 175], and recidivism prediction [176, 177]. The clear effect of these applications on the welfare of individuals and groups coupled with recorded instances of algorithmic bias and harm [178, 179] has made fairness—in its many forms under different interpretations, with its many definitions—a prominent consideration in algorithm design.

Thus, it is unsurprising that fair *unsupervised learning* has received great interest in the AI/ML, statistics, operations research, and optimization communities—including *fair clustering*. Clustering is a central problem in AI/ML and operations research and arguably the most fundamental problem in unsupervised learning writ large. The literature in fair clustering has produced a significant number of publications spanning a wide range of fairness notions [see, e.g., 180, for an overview]. However, the most prominent of the fairness notions that were introduced is the *group fairness* notion due to Chierichetti et al. [7], Bercea et al. [12], and Bera et al. [8]. Since our work is concerned with this notion in particular, for ease of exposition we will simply refer to it as fair clustering. In fair clustering,

each point belongs to a demographic group and therefore each demographic group has some percentage representation in the entire dataset.[1] Like in agnostic (ordinary or "unfair") clustering the dataset is partitioned into a collection of clusters. However, unlike agnostic clustering each cluster must have a proportional representation of each group that is close to the representation in the entire dataset. For example, if the dataset consists of groups A and B at 30% and 70% representation, respectively. Then each cluster in the fair clustering should have a $(30 \pm \epsilon)\%$ and $(70 \pm \epsilon)\%$ representation of groups A and B, respectively.

One can see a significant advantage behind group fairness. Each cluster has close to dataset-level representation[2] of each group, so any outcome associated with any cluster will affect all groups proportionally, satisfying the disparate impact doctrine [159], as discussed by Chierichetti et al. [7]. Despite the attractive properties of group fairness, it requires complete knowledge of each point's membership in a group. In practice—say, in an advertising setting where membership is estimated via a machine learning model, or in a lending scenario where membership may be illegal to estimate at train time—knowledge of group membership may range from noisy, to adversarially corrupted, to completely unknown. This salient problem has received significant attention in fair *classification* [see, e.g., 52, 53, 54, 55, 56, 57]. However, this important consideration has not received significant attention in fair *clustering* with the exception of the theoretical work of Esmaeili et al. [58] that introduced uncertain group membership and the empirical work of Chhabra et al. [59], who provide a data-driven approach to achieve robustness against adversarial perturbations on fair clustering systems.

Our work addresses the practical modeling shortcomings of both Esmaeili et al. [58] and

[1] As a simple example, the demographic groups could be based on income. Therefore, each point would belong to an income bracket and each income bracket would have some percentage representation (e.g., 20% belong to group "$\leq$ USD$30k," 10% belong to group "$\geq$ USD$250k," and so on) of the dataset.

[2] And, ideally, *population-level* representation—yet this may not hold in common machine learning applications, where proportionally sampling an underlying population to form a training dataset requires deep nuance. For a discussion of biased sampling and its implications in fair machine learning, we direct the reader to Barocas et al. [66, Chapters 4 & 6].

Chhabra et al. [59] and gives new theoretical worst-case guarantees. In short, the model of Esmaeili et al. [58] makes the strong assumption of having probabilistic information about the group membership of each point in the dataset and the weak guarantee of having proportional representation of each group in every cluster but only in expectation. Further, Chhabra et al. [59] looks into *black-box adversarial perturbations* on fair clustering. However, in their model it is assumed that only a fixed subset of points in the dataset will have their memberships perturbed and it is not clarified how this fixed subset is exactly decided. In Section 7.3.2 and Appendix 7.8 we give a more detailed comparison to these prior works and demonstrate their weaknesses.

7.1.1 Outline and Contributions:

In Section 7.2, we briefly go over some prior work in fair clustering and other works in fair classification with emphasis on papers that tackle the incomplete/noisy group membership case. Then in Section 7.3, we formally describe the basic clustering setting and introduce our notation then we give an overview of the prior noise models of [58] and [59]. In Section 2.2, we present our noise model. Our model requires a small number of parameters as input instead of full probabilistic information for each point. In fact, as a special case it can be given only one parameter that represents the bound on the maximum number of incorrectly assigned group memberships in the dataset. Based on the framework of robust optimization, we then define the *robust fair clustering* problem for the k-center objective. In Section 7.5, we present our theoretically grounded algorithm to solve the robust fair k-center problem. Our algorithms require making careful observations about the structure of a robust fair solution and represents a novel addition to the existing fair clustering algorithms. Finally, in Section 2.5 we validate the performance of our algorithm on real world datasets and show that it has superior performance in

comparison to the existing methods.

7.2 Additional Related Work

We will focus on the fairness notion most relevant to us in fair clustering, specifically where the solution is constrained to have proportional group representation in each cluster [e.g., 7, 8, 12, 13, 14, 15, and others]. Under the assumption that group memberships are perfectly known, this notion is well-investigated. For example, Backurs et al. [142] gives faster scalable algorithms for this problem to handle large datasets. Bera et al. [8] have considered a variant of this problem when each point is allowed to belong to more than one group simultaneously. Further, variants of this notion in non-centroid based clustering have also been considered. Ahmadian et al. [181] address the same fairness notion in correlation clustering whereas Kleindessner et al. [97] address it in spectral clustering, and Knittel et al. [182, 183] address it in hierarchical clustering.

The problem of incomplete and imperfect knowledge of group memberships has received significant attention in fair classification. Awasthi et al. [52] study the effects on the equalized odds notion of Hardt et al. [184] when the group memberships are perturbed. Awasthi et al. [53] study the effects of using a classifier to predict the group membership of a point on the bias of downstream ML tasks. Kallus et al. [56] study a similar problem but focusing mainly on assessing the disparate impact in various applications when the group memberships are not unavailable and have to be predicted instead. Robust optimization methods were used to obtain fair classifiers under the setting of noisy group memberships by Wang et al. [e.g., 54] and unavailable group memberships by Hashimoto et al. [e.g., 55]. While our problem falls under the robust optimization framework, our techniques are very different.

7.3 Preliminaries and Previous Noise Models

In this section we go through preliminary background, notation, and previously introduced noise models in clustering.

7.3.1 Preliminaries

Let $\mathcal{P}$ be a set of n points in a metric space with distance function $d : \mathcal{P} \times \mathcal{P} \to \mathbb{R}_{\geq 0}$. In k-center clustering, the goal is to select a set of centers S from $\mathcal{P}$ of at most k points and an assignment $\phi : \mathcal{P} \to S$ minimizing the clustering cost which is $\text{cost}(S, \phi) := \max_{j \in \mathcal{P}} d(j, \phi(j))$, i.e., the cost is the maximum distance between a point and its assigned center. Since the clustering cost is the maximum distance between a point and its center, we also refer to that cost as the clustering radius or just radius. Clearly, in the ordinary k-center problem, ϕ will assign each point $j \in \mathcal{P}$ to its closest center in S, i.e., $\phi(j) = \arg\min_{i \in S} d(j, i)$. On the other hand, finding ϕ is non-trivial in more general k-center variants when constraints are imposed, as ϕ may assign points to centers that are further away to satisfy the imposed constraint.

We index the set of ℓ many demographic groups that exist in the dataset by $\mathcal{G} = \{1, 2, \ldots, \ell\}$. Following the fair clustering literature we associate a specific color with each group [7, 8, 12]. Therefore, we use the words group and color interchangeably. Let $\mathcal{P}_g$ denote the subset of points in $\mathcal{P}$ that are assigned color g. Each point $j \in \mathcal{P}$ belongs to exactly one color from the set of colors $\mathcal{G}$. We can equivalently describe this assignment using the function $\chi : \mathcal{P} \to \mathcal{G}$, such that for any $j \in \mathcal{P}_g$, the color assignment for j would be $\chi(j) = g$. We denote the total number of points of color g by $n_g = |\mathcal{P}_g|$, it follows that $\sum_{g \in \mathcal{G}} n_g = n$.

Further, given a solution (S, ϕ), for each $i \in S$, C_i denotes the set of points assigned to center

i (i.e., cluster i) and $C_{i,g}$ denotes the subset of points in that cluster belonging to group g. The FAIR-k-CENTER problem [e.g., 7, 8, 12, 58] adds the following fairness constraint to the k-center objective, formally the optimization problem is:

$$\min_{S:|S|\leq k,\phi} \max_{j\in\mathcal{P}} d(j,\phi(j)) \tag{7.1a}$$

$$\forall i \in S, \forall g \in \mathcal{G} : \alpha_g \leq \frac{|C_{i,g}|}{|C_i|} \leq \beta_g \tag{7.1b}$$

where α_g and β_g are proportion bounds that satisfy $0 < \alpha_g \leq r_g \leq \beta_g < 1$ with r_g being the ratio (proportion) of group g in the entire set of points, i.e., $r_g := \frac{n_g}{n}$. Therefore, an instance of FAIR-k-CENTER is parametrized by the tuple $(\mathcal{P}, \chi, k, \mathcal{G}, \vec{\alpha}, \vec{\beta})$.

7.3.2 Previous Noise Models in Fair Clustering

In this section we give more details about [58] and [59], the two prior works which have considered robustness in fair clustering. [58] introduced a probabilistic noise model where each point $j \in \mathcal{P}$ has a probability $p_{j,g} \in [0,1]$ of belonging to group g, with $\sum_{g\in\mathcal{G}} p_{j,g} = 1$. While their algorithms satisfy proportional fairness constraints in expectation[3], the worst-case realization can significantly violate these constraints as noted earlier. In fact, in Section 7.8.1 we show an example where a clustering of the given points satisfies fairness in expectation, but violates it completely in realization. This highlights a core deficiency in this model.

Chhabra et al. [59] introduced an adversarial model where the adversary has access to a subset of points whose group memberships can be modified. However, they do not specify how this subset is

[3]Since each point has some probability of belonging to each specific group, one can calculate the expected number of points belonging to a specific group in a clustering by simply adding the points' probabilities in that cluster.

selected. In their experiments, they independently sample points with equal probability and add them to this subset. We can construct instances where with high probability certain point combinations are never sampled in the subset, thereby heavily restricting the model's capability. We give a concrete discussion of this in Section 7.8.2. Moreover, their algorithm does not have theoretical guarantees.

7.4 Our Noise Model and Problem Statement

In our model we assume that there exists a number of points whose group memberships (colors) have been incorrectly assigned to other groups. The two main considerations in our model are that: (1) in general these incorrect assignments exhibit a heterogeneity across the groups and (2) that the incorrect assignments can be arbitrarily allocated across the dataset.

The first consideration is based on the fact that in many settings there exist group memberships that are more desirable than others and therefore individuals may misreport their group memberships as other more favorable groups [185]. Further, the mechanism through which the group memberships were assigned may exhibit higher error rates for particular groups. For example, the method used to elicit group memberships may fail with higher rates on some particular groups. Therefore, noise exhibits heterogeneity across the groups and an effective noise model should capture that.

The second consideration is based on the fact that incorrect group assignments could arise from a set of possibilities and therefore unlike [59] we should not assume knowledge of these particular noisy points. The (noise) perturbations in the group assignments could have resulted from a process similar to iid noise as done in [186] and [187]. At another extreme, the group memberships could have been assigned using a machine learning classifier which predicts the group memberships, in that case if the classifier's errors are localized to a specific region in the feature space[4] then clearly the group

[4]This could be the case, if the training dataset happens to be particularly scare in that region.

membership perturbations do not act similar to random noise. Further, note that both scenarios are empirically well-motivated and could possibly occur in the same dataset simultaneously. Therefore, an effective noise model should not assume knowledge of the spatial noise distribution and allow noise to be arbitrarily allocated across the dataset.

Now, we delve into the formal description of the model. For a given color $g \in \mathcal{G}$ we associate two parameters m_g^+ and m_g^-, the first is the maximum number of points that were mistakenly assigned group memberships other than g and the second is the maximum number of points that were mistakenly assigned to group g.

Further, for a given set of value m_g^+ and m_g^-, by definition the consistency of the values requires that the following inequalities should be satisfied:

$$\forall g \in \mathcal{G} : m_g^+ \leq \sum_{g \in \mathcal{G}, g \neq h} m_h^- \tag{7.2}$$

$$\forall g \in \mathcal{G} : m_g^- \leq \sum_{g \in \mathcal{G}, g \neq h} m_h^+ \tag{7.3}$$

In words, the first inequality (7.2) simply states that no color should "gain" more points than the total number of points "lost" by the other groups. Similarly, the second inequality (7.3) states that if a color loses some number of points than the rest of the colors must gain at least the same amount in total. Note that since no color can lose more points than it has, an additional set of inequalities is also implied, namely $\forall g \in \mathcal{G} : m_g^- \leq n_g$. We did not list it as we assume that any given set values of m_g^- always satisfy it.

The values of m_g^+ and m_g^- lead to new possible group membership assignments (colorings) $\hat{\chi}$ other than the original coloring χ. Following the language of robust optimization [188], the set of all

possible colorings $\hat{\chi} : \mathcal{P} \to \mathcal{G}$ that result from a given set of values $\{m_g^+, m_g^-\}_{g\in\mathcal{G}}$ is referred to as the *uncertainty set* $\mathcal{U}$. We use $\{\mathcal{P}_g\}_{g\in\mathcal{G}}$ to denote the color partition of $\mathcal{P}$ where $j \in \mathcal{P}_g$ has color g and we define $\chi^{-1}(g) = \mathcal{P}_g$. We can analogously define the partition $\{\hat{\mathcal{P}}_g\}_{g\in\mathcal{G}}$ for the assignment $\hat{\chi}$ where $\hat{\chi}^{-1}(g) = \hat{\mathcal{P}}_g$. More formally, given a valid set of noise parameters $\{m_g^+, m_g^-\}_{g\in\mathcal{G}}$ satisfying inequalities (7.2) and (7.3) the uncertainty set $\mathcal{U}$ is

$$\mathcal{U} = \{\hat{\chi} \mid \hat{\chi} \text{ satisfies (7.5a)} - \text{(7.5b)}\} \tag{7.4}$$

$$\forall g \in \mathcal{G} : |\mathcal{P}_g \setminus \hat{\mathcal{P}}_g| \leq m_g^- \tag{7.5a}$$

$$\forall g \in \mathcal{G} : |\hat{\mathcal{P}}_g \setminus \mathcal{P}_g| \leq m_g^+ \tag{7.5b}$$

Proposition 7.4.1. *For any instance with noise parameters* $\{m_g^+, m_g^-\}_{g\in\mathcal{G}}$*, the group uncertainty set* $\mathcal{U}$ *is defined as the set of all group assignments* $\hat{\chi} : \mathcal{P} \to \mathcal{G}$ *that satisfy the constraints in* (7.5a) *and* (7.5b).

Proof. Let $\hat{\chi}$ be a feasible assignment in the uncertainty set and let $\hat{\mathcal{P}}_g = \hat{\chi}^{-1}(g)$ be the collection of points assigned color g by $\hat{\chi}$. For any color g, $\mathcal{P}_g \setminus \hat{\mathcal{P}}_g$ denotes the set of points that are assigned color g by χ and a different color $h \neq g$ by $\hat{\chi}$. Clearly, by definition of m_g^- the first constraint should hold, i.e.

$$|\mathcal{P}_g \setminus \hat{\mathcal{P}}_g| \leq m_g^- \tag{7.6}$$

Furthermore, $\hat{\mathcal{P}}_g \setminus \mathcal{P}_g$ denotes the set of points that are assigned color g by $\hat{\chi}$ but were assigned a

different color $h \neq g$ by χ. By definition of m_g^+ the second constraint holds, i.e.

$$|\hat{\mathcal{P}}_g \setminus \mathcal{P}_g| \leq m_g^- \tag{7.7}$$

□

The above simply states that any $\hat{\chi}$ coloring (element) in the uncertainty set should result in an assignment where (i) the number of points that are assigned a color g by χ but actually belong to a color $h \neq g$ should not exceed m_g^- and (ii) the number of points that are assigned a color $h \neq g$ by χ but actually have color g is no more than m_g^+.

For the case of two colors (denote them by red and blue), we naturally have $m_{\text{red}}^+ = m_{\text{blue}}^-$ and $m_{\text{red}}^- = m_{\text{blue}}^+$. To see that, note that using inequalities (7.2) and (7.3) it follows that $m_{\text{red}}^+ \leq m_{\text{blue}}^- \leq m_{\text{red}}^+$ and therefore $m_{\text{red}}^+ = m_{\text{blue}}^-$. Similarly, one can show that $m_{\text{red}}^- = m_{\text{blue}}^+$. The new implied equalities are natural as they simply state that what one color loses is gained by the other and vice versa.

To give a sense of the resulting group assignments implied by a given set of values m_g^+ and m_g^- consider the two color toy example shown in Figure 7.1. In this example, we have $m_{\text{red}}^- = 2$ whereas $m_{\text{blue}}^- = 1$. Further, from the previous discussion since we have two colors then we immediately have $m_{\text{red}}^+ = 1$ and $m_{\text{blue}}^+ = 2$. The figure shows the many possible colorings that can result from the given noise values, note that even for this simple example there are a total of 12 possibilities which is larger than the number of given points 4. A robust fair clustering has to achieve fairness over all possible colorings (all colorings in the uncertainty set).

We further note that a simple possible assignment of the color parameters would set them all to the same value, i.e., $\forall g \in \mathcal{G} : m_g^+ = m_g^- = m$. This essentially states that any color can increase or decrease by m points. Figure 7.2 shows the same previous example where all noise parameters have

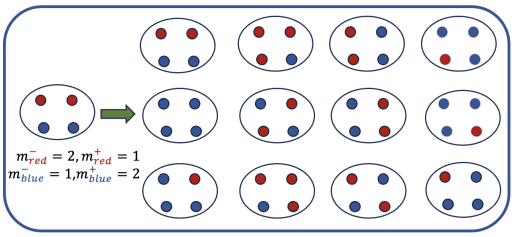


Figure 7.1: All colorings in the uncertainty set of a toy example with 4 points and $m^-_{\text{red}} = 2, m^-_{\text{blue}} = 1$ are shown.

been set to 2. Clearly, the number of possible colorings (size of the uncertainty set) has increased from 12 to 16. We are now ready to state our problem. Given an instance of *fair clustering* $(\mathcal{P}, \chi, k, \mathcal{G}, \vec{\alpha}, \vec{\beta})$

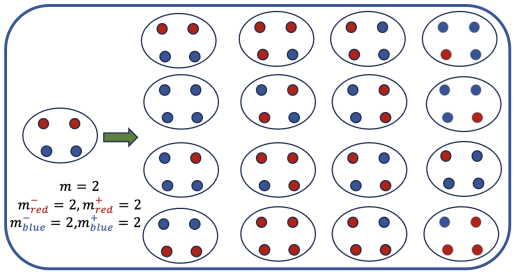


Figure 7.2: All colorings in the uncertainty set for the same toy example, now with $m = 2$. Note the new color assignments in the bottom row where the two (originally) blue points become red.

along with noise parameters $\{m^+_g, m^+_g\}_{g \in \mathcal{G}}$. The objective of the RobustFair-k-Center problem is to find a clustering that minimizes the k-center objective while ensuring that the fairness constraints are satisfied for every possible coloring in the uncertainty set $\mathcal{U}$. Formally, the optimization problem of RobustFair-k-Center is

$$\min_{S:|S|\leq k,\phi} \max_{j \in \mathcal{P}} d(j, \phi(j)) \tag{7.8a}$$

$$\forall \hat{\chi} \in \mathcal{U}, \forall i \in S, g \in \mathcal{G} : \alpha_g \leq \frac{|C_{i,g}(\hat{\chi})|}{|C_i|} \leq \beta_g \tag{7.8b}$$

where $C_{i,g}(\hat{\chi}) := \hat{\chi}^{-1}(g) \cap C_i$ denotes the subset of points in C_i which have been assigned to group g by a coloring $\hat{\chi} \in \mathcal{U}$. We also define a λ-violating solution as one where the fairness constraints of

(7.8b) are violated by at most λ, formally a λ-violating solution satisfies

$$\forall \hat{\chi} \in \mathcal{U}, \forall i \in S, g \in \mathcal{G} : \alpha_g - \lambda \leq \frac{|C_{i,g}(\hat{\chi})|}{|C_i|} \leq \beta_g + \lambda \tag{7.9}$$

Clearly, smaller λ implies a smaller violation of the fairness constraints and any value of $\lambda \geq 1$ is vacuous.

Finally, we note that our model requires the decision maker to specify a total of at most 2ℓ many parameters unlike [58] which needs a total of at least $(\ell - 1) \cdot n$ many parameters (necessarily growing with the size of the dataset). The values in our model can be simply set using prior knowledge and statistics. Furthermore, if the decision maker does not posses fine-grained knowledge about the noise parameters m_g^+, m_g^- for each group then deciding one value m and setting $m_g^+ = m_g^- = m$ to upper bound the total change in any group would be sufficient. While this would increase the size of the uncertainty set, a clustering that is robust fair under an uncertainty set remains robust fair under a more restricted one. More formally, given a problem instance and two uncertainty sets $\mathcal{U}'$ and $\mathcal{U}$, if $\mathcal{U}' \subset \mathcal{U}$ then it follows immediately that a λ-violating solution under $\mathcal{U}$ is also a λ-violating solution under $\mathcal{U}'$. However, we note that while one may always expand the uncertainty set to ensure fairness for higher noise values it would come at an expense. Specifically, a robust solution would have to ensure fairness over a larger uncertainty set and that would in general lead the optimization objective (which is the clustering cost/quality) to be degraded.

7.5 Algorithm and Theoretical Analysis

We start this section by making a collection of mathematical observations that are essential for our algorithm. First, note that the size of the uncertainty set $|\mathcal{U}|$ can grow exponentially in the size of

the dataset n. For example, for the two color case with $m_g^+ = m_g^- = m$ for both colors, the size of $\mathcal{U}$ is $\sum_{0 \le i,j \le m} \binom{n_1}{i}\binom{n_2}{j} \ge \left(\frac{n}{2m}\right)^m$ where n_1 and n_2 are the number of points for the first and second color. For a reasonable choice of $m = \alpha n$ where α is a fractional constant in $(0, 1)$, it is straight forward to see that the size of $|\mathcal{U}| = \Omega(c^n)$ where c is a constant strictly greater than 1. This implies that we can have an exponential set of constraints in (7.8b). We begin proving a simple observation.

Observation 7.5.1. *Suppose that (S, ϕ) is robust fair solution to a given input instance and $\beta_g < 1$ and $\alpha_g > 0$. Then the clustering $\{C_1, C_2, \cdots, C_{k'}\}$ of points induced by (S, ϕ) must satisfy the following,*

- *For any $i \in S$, $g \in \mathcal{G}$, $|C_{i,g}(\chi)| > m_g^-$.*
- *For any $i \in S$, $g \in \mathcal{G}$, $\sum_{h \in \mathcal{G}, h \neq g} |C_{i,h}(\chi)| > m_g^+$.*

Proof. We know that since the clustering is robust fair it must satisfy the constraints in Equation (7.8b) i.e.,

$$\forall i \in S, \forall g \in \mathcal{G} : \min \frac{|C_{i,g}(\hat{\chi})|}{|C_i|} \ge \alpha_g \implies \min |C_{i,g}(\hat{\chi})| > 0$$

However, we know that there can be as much as m_g^- points that have incorrect group memberships from each color g. Therefore we have

$$\forall i \in S, \forall g \in \mathcal{G} : \min |C_{i,g}(\hat{\chi})| > 0 \implies |C_{i,g}(\chi)| - m_g^- > 0$$

Thus, we show the first part of the claim.

For each $i \in S$ and color $g \in \mathcal{G}$, using the lower bound on $|C_{i,g}(\chi)|$ obtained in the first part, we can derive a lower bound on $\sum_{h \in \mathcal{G}, h \neq g} |C_{i,h}(\chi)|$ by summing over all the groups $h \in \mathcal{G}$ and $h \neq g$ as

follows,

$$\sum_{h\in\mathcal{G},h\neq g} |C_{i,h}(\chi)| > \sum_{h\in\mathcal{G},h\neq g} m_h^- \geq m_g^+$$

The last inequality is from the inequality (7.2) which says that the number of points gained by any group g is at most the total number of points lost by the remaining groups $h \neq g$ i.e., $\sum_{h\in\mathcal{G},h\neq g} m_h^- \geq m_g^+$. Therefore, we get the desired bound. □

Our first critical observation is that we can replace this exponential set by an equivalent polynomial sized set of constraints.

Lemma 7.5.1. *For any instance of* RobustFair-k-Center *the constraints* (7.8b) *are equivalent to* (7.10a) *and* (7.10b).

$$\forall i \in S, g \in \mathcal{G} : \frac{|C_{i,g}(\chi)| + m_g^+}{|C_i|} \leq \beta_g \tag{7.10a}$$

$$\forall i \in S, g \in \mathcal{G} : \frac{|C_{i,g}(\chi)| - m_g^-}{|C_i|} \geq \alpha_g \tag{7.10b}$$

Proof. To show that the constraints (7.8b) are equivalent to (7.10a) and (7.10b), we have to show that for any feasible clustering $\{C_1, C_2, \ldots, C_{k'}\}$ with $k' \leq k$ satisfying (7.8b) must also satisfy (7.10a) and (7.10b) and vice versa.

First we show the forward direction, i.e., if $\{C_1, C_2, \ldots, C_{k'}\}$ satisfies (7.8b) then it also satisfies (7.10a) and (7.10b). Since the fairness constraints in (7.8b) must hold for all points in the uncertainty set $\mathcal{U}$, they are also valid for the worst-case $\hat{\chi}$ in $\mathcal{U}$ i.e.,

$$\forall \hat{\chi} \in \mathcal{U}, \quad \frac{|C_{i,g}(\hat{\chi})|}{|C_i|} \leq \beta_g \implies \max_{\hat{\chi}\in\mathcal{U}} \frac{|C_{i,g}(\hat{\chi})|}{|C_i|} \leq \beta_g.$$

However, in each cluster C_i, we know that the maximum number of points from color g that are incorrectly labeled as g but actually belong to a different group h given by $\min\{m_g^+, \sum_{h\in\mathcal{G},h\neq g} |C_{i,h}(\hat{\chi})|\}$. Therefore, for each $1 \leq i \leq k'$ and $g \in \mathcal{G}$, we have

$$\max_{\hat{\chi}\in\mathcal{U}} \frac{|C_{i,g}(\hat{\chi})|}{|C_i|} = \frac{|C_{i,g}(\hat{\chi})| + \min\{m_g^+, \sum_{g\in\mathcal{G},g\neq h} |C_{i,g}(\hat{\chi})|\}}{|C_i|} = \frac{|C_{i,g}(\hat{\chi})| + m_g^+}{|C_i|} \leq \beta_g.$$

The last equality is from the fact in Observation 7.5.1 that $\sum_{h\in\mathcal{G},h\neq g} |C_{i,h}(\hat{\chi})| > m_g^+$

Next we prove that if a clustering satisfies constraints in (7.8b) then it also satisfies (7.10b). Since the clustering is feasible we know that,

$$\forall \hat{\chi} \in \mathcal{U}, \quad \frac{|C_{i,g}(\hat{\chi})|}{|C_i|} \geq \alpha_g \implies \min_{\hat{\chi}\in\mathcal{U}} \frac{|C_{i,g}(\hat{\chi})|}{|C_i|} \geq \alpha_g.$$

We know that in each C_i, the maximum number of points mistakenly assigned a color g but actually belonging to group $h \neq g$ is $\min\{m_g^-, |C_{i,g}(\chi)|\}$. However from the first part of Observation 7.5.1 it is clear that, for any cluster C_i, we have $|C_{i,g}(\chi)| > m_g^-$ which implies that $\min\{m_g^-, |C_{i,g}(\chi)|\} = m_g^-$. Therefore, we have

$$\min_{\hat{\chi}\in\mathcal{U}} \frac{|C_{i,g}(\hat{\chi})|}{|C_i|} = \frac{|C_{i,g}(\chi)| - \min\{m_g^-, |C_{i,g}(\chi)|\}}{|C_i|} = \frac{|C_{i,g}(\chi)| - m_g^-}{|C_i|} \geq \alpha_g.$$

This concludes the forward direction of our proof. Next, we show that if (7.10b) and (7.10a) holds, then Equation (7.8b) also holds. We know that for any $1 \leq i \leq k'$, C_i satisfies the upper bound constraints in (7.10a). This implies that any feasible assignment $\hat{\chi}$ contains at most $|C_{i,g}(\chi)| + m_g^+$ points of color g in C_i. From (7.5b) in Proposition 7.4.1 we know that, for any assignment in the

uncertainty set at most m_g^+ additional points are assigned to color g. Therefore, we have

$$\forall g \in \mathcal{G} : \frac{|C_{i,g}(\chi)| + m_g^+}{|C_i|} \leq \beta_g \implies \forall \hat{\chi} \in \mathcal{U} : \frac{|C_{i,g}(\hat{\chi})|}{|C_i|} \leq \frac{|C_{i,g}(\chi)| + m_g^+}{|C_i|} \leq \beta_g.$$

For the case of lower bound constraints we know that C_i satisfies the constraints in (7.10b). Therefore, we can say that any feasible assignment cannot have less than $|C_{i,g}(\chi)| - m_g^-$ points of color g in C_i. Therefore, for any $1 \leq i \leq k'$,

$$\forall g \in \mathcal{G} : \frac{|C_{i,g}(\chi)| - m_g^-}{|C_i|} \geq \alpha_g \implies \forall \hat{\chi} \in \mathcal{U}, \quad \frac{|C_{i,g}(\hat{\chi})|}{|C_i|} \geq \frac{|C_{i,g}(\chi)| - m_g^-}{|C_i|} \geq \alpha_g$$

as desired. Therefore, the constraints in (7.8b) are equivalent to (7.10a) and (7.10b). □

We call the constraints (7.10a) and (7.10b) the *robust fairness* constraints. To see what the lemma means, consider a color g and its upper proportion bound β_g. The lemma essentially states that instead of ensuring that the solution satisfies $\frac{|C_{i,g}(\hat{\chi})|}{|C_i|} \leq \beta_g$ for all colorings $\hat{\chi} \in \mathcal{U}$ as done in (7.8b), we may instead take $C_{i,g}(\chi)$ (which uses the given coloring χ) and add the highest (worst-case) increase in the number of points that can be gained by color g which is m_g^+ and satisfy a single constraint of $\frac{|C_{i,g}(\chi)|+m_g^+}{|C_i|} \leq \beta_g$ in (7.10a) instead. A similar statement can be made about the lower bound α_g in (7.10b).

As a result of Lemma 7.5.1, it follows that a λ-violating solution of constraint (7.9) is only required to satisfy the following reduced constraints.

$$\forall i \in S, g \in \mathcal{G} : \frac{|C_{i,g}(\chi)| + m_g^+}{|C_i|} \leq \beta_g + \lambda \tag{7.11a}$$

$$\forall i \in S, g \in \mathcal{G} : \frac{|C_{i,g}(\chi)| - m_g^-}{|C_i|} \geq \alpha_g - \lambda \tag{7.11b}$$

Another critical observation is that upper and lower proportion bounds that would have a non-empty set of feasible solutions in ordinary fair clustering might lead to an infeasible RobustFair-k-Center instance unless the bounds are relaxed by a sufficient margin. To see that, consider an instance of RobustFair-k-Center with two red and two blue points and noise parameters $m^+_{\text{red}} = m^+_{\text{blue}} = m = 1$. If we set the proportion bounds to $\alpha_{\text{red}} = \alpha_{\text{blue}} = 1/2$ and $\beta_{\text{red}} = \beta_{\text{blue}} = 1/2$, then clearly we would have an ordinary (non-robust) fair clustering solution. However, one can see through Lemma 7.5.1 that no feasible robust fair solution exists. Now, if we relax the bounds to $\alpha_{\text{red}} = \alpha_{\text{blue}} = 1/4$ and $\beta_{\text{red}} = \beta_{\text{blue}} = 3/4$, then a single cluster containing all four points becomes a feasible solution. More formally, the proportions bounds have to be relaxed exactly as shown in the following observation:

Observation 7.5.2. *For any instance of the* RobustFair-k-Center *problem, a feasible solution exists if and only if for every group $g \in \mathcal{G}$, $\beta_g \geq \frac{n_g + m_g^+}{n}$ and $\alpha_g \leq \frac{n_g - m_g^-}{n}$.*

Proof. Suppose we have a clustering $\{C_1, \ldots, C_{k'}\}$) with $k' \leq k$. As before $C_{i,g}(\chi)$ denotes the set of points belonging to group $\mathcal{G}$ in the original coloring. Applying the Fact A.0.1, we get the following,

$$\min_{i \in [k']} \frac{|C_{i,g}(\chi)|}{|C_i|} \leq \frac{\sum_{i \in [k']} C_{i,g}(\chi)}{\sum_{i \in [k']} C_i} = \frac{|\mathcal{P}_g|}{|\mathcal{P}|} \leq \max_{i \in [k']} \frac{|C_{i,g}(\chi)|}{|C_i|} \tag{7.12}$$

Suppose that $\beta_g < \frac{n_g + m_g^+}{n}$ for some color, then by the definition of the robust optimization problem, it is possible to have $n_g + m_g^+$ many points of color g in the dataset. Accordingly, we it must be that for some color assignment $\frac{|\mathcal{P}_g|}{|\mathcal{P}|} = \frac{n_g + m_g^+}{n}$ but (7.12) implies that there exists some value $i \in [k']$ such that $\frac{|C_{i,g}(\chi)|}{|C_i|} \geq \frac{|\mathcal{P}_g|}{|\mathcal{P}|} = \frac{n_g + m_g^+}{n}$. Therefore, $|C_{i,g}(\chi)| > \beta_g |C_i|$ and therefore the solution is infeasible. The same argument can be made for the lower bound as well.

We will now show that if $\forall g \in \mathcal{G} : \beta_g \geq \frac{n_g+m_g^+}{n}, \alpha_g \leq \frac{n_g-m_g^-}{n}$, then the problem must be feasible. To show feasibility we simply show one feasible solution. Specifically, a solution which is always feasible is a one cluster solution that includes all of the points, i.e. $\{C_1\} = \{\mathcal{P}\}$. Clearly, we have for any color g : $\frac{n_g-m_g^-}{n} \leq \frac{|\mathcal{P}_g|}{|\mathcal{P}|} \leq \frac{n_g+m_g^+}{n}$. Since the $\beta_g \geq \frac{n_g+m_g^+}{n}, \alpha_g \leq \frac{n_g-m_g^-}{n}$, then the solution is feasible. □

7.5.1 Our Algorithm: RobustAlg

To solve our problem we employ a two-stage approach where the initial stage selects the centers and the second stage assigns the points to the centers. While various prior papers in fair clustering Bera et al. [8], Bercea et al. [12], Esmaeili et al. [58, 157] use a similar two-stage approach, the centers used in the first stage are selected by any vanilla (ordinary) k-center algorithm. In our case, it is actually critical that the centers are selected carefully by our algorithm (subroutine) GetCenters. In fact, in Section 7.5.2 we show how using another algorithm would break a critical step in our proof.

Since we are dealing with a k-center objective, the optimal radius (the maximum distance from any point to its assigned center) for RobustFair-k-Center belongs to a finite set of possible values, i.e., the set of $\binom{n}{2}$ distance values. Therefore, our subroutine GetCenters (Algorithm 10) receives as input a "guessed" radius value R along with the entire set of points $\mathcal{P}$. GetCenters outputs a set of centers S. The procedure begins with all the points being unmarked, then in each iteration we add an arbitrary unmarked point j to S, and mark the all the points at a distance of at most $2R$ from j (this includes point j as well). We repeat this step until all the points are marked.

We will show that the centers returned by GetCenters when run at a sufficiently large value of R satisfy good properties that enable us to post-process them to obtain a *robust fair* clustering.

Algorithm 10 GETCENTERS

Input: Set of points $\mathcal{P}$, and a radius R
Output: Cluster centers S

$S \leftarrow \varnothing$
while $\mathcal{P} \neq \varnothing$ **do**
    Pick an arbitrary $j \in \mathcal{P}$ and $S \leftarrow S \cup \{j\}$
    $\mathcal{P} \leftarrow \mathcal{P} \setminus \text{Ball}(j, 2R)$
end while
return S

Lemma 7.5.2. *For any $R \geq R^*$ and set of centers $\hat{S}$ returned by the* GETCENTERS *subroutine, (i) there exists an assignment $\phi' : \mathcal{P} \to \hat{S}$ at a clustering cost of at most $3R$, i.e., cost$(\hat{S}, \phi') \leq 3R$ and (ii) the assignment leads each center $i \in \hat{S}$ to have a cluster that is robust fair.*

Proof. Suppose that we have an optimal solution (S^*, ϕ^*) with cost R^*. Each point j is assigned to some center $i^* \in S^*$ by the optimal assignment ϕ^*. Let C_{i^*} denote the set of points assigned to i^*, i.e., $C_{i^*} = \phi^{*-1}(i^*)$. We show the existence of an assignment $\phi' : \mathcal{P} \to \hat{S}$ from the set of points $\mathcal{P}$ to the set of centers $\hat{S}$ returned by Algorithm 10 such that (i) the assignment ϕ' has cost$(\hat{S}, \phi') \leq 3R$ and (ii) each cluster corresponding to a center $i \in \hat{S}$ is *robust fair*. Note that the proof is non-constructive since it assumes knowledge of the optimal solution.

We construct the assignment ϕ' as follows: ϕ' assigns all the points in each cluster C_{i^*} to the center $i' \in \hat{S}$ if i' belongs to the cluster C_{i^*}, i.e., if $\phi^*(i') = i^*$. It is possible that there are clusters C_{i^*} with no point $i' \in \hat{S}$. Therefore, we assign such clusters to a center i' in $\hat{S}$ where $d(i^*, i') \leq 2R$, i.e., i' is at a distance of at most $2R$ from the cluster's center. Since every point has at least one center in $\hat{S}$ at a distance of at most $2R$. This concludes the description of our assignment ϕ'.

Notice that each $i' \in \hat{S}$ gets assigned all the points associated with at least one center $i^* \in S^*$. This is because any two centers in $\hat{S}$ are separated by a distance strictly greater than $2R$ and therefore no two centers in $\hat{S}$ are selected from the same optimal cluster. Further, each center $i^* \in S^*$ gets

assigned to some $i' \in \hat{S}$ at a distance of at most $2R$. We now prove that this new assignment has a cost of at most $3R$, i.e., $\text{cost}(\hat{S}, \phi') \leq 3R$. This holds because for any point $j \in \mathcal{P}$ we have:

$$d(j, \phi'(j)) \leq d(j, \phi^*(j)) + d(\phi^*(j), \phi'(j)) \tag{7.13}$$

$$\leq R^* + d(\phi^*(j), \phi'(j)) \tag{7.14}$$

$$\leq R^* + 2R \tag{7.15}$$

$$\leq 3R. \tag{7.16}$$

Inequalities (7.13) follows from triangle inequality. Inequality (7.14) follows from the fact that ϕ^* is an optimal *robust fair* assignment. Finally, inequality (7.15) is from the fact that $d(\phi^*(j), \phi'(j)) \leq 2R$ since each center $i^* \in S^*$ is assigned to a center $i' \in \hat{S}$ at a distance of at most $2R$. It remains to show that for each $i' \in \hat{S}$ the corresponding cluster $\phi'^{-1}(i')$ is *robust fair*, i.e., satisfying the constraints (7.10b) and (7.10a). Claim 7.5.1 concludes the proof of this lemma. □

Claim 7.5.1. *The clustering induced by $(\hat{S}, \phi')$ leads each center $i' \in \hat{S}$ to be robust fair.*

Proof. Recall that C_{i^*} denotes the set of points that are assigned to center $i^* \in S^*$ in the optimal clustering (S^*, ϕ^*). Since (S^*, ϕ^*) is an optimal clustering to our problem, it must satisfy the constraints in (7.10a) and (7.10b). Therefore, we have the following:

$$\forall i^* \in S^*, \forall g \in \mathcal{G} : \frac{|C_{i^*,g}(\chi)| + m_g^+}{|C_{i^*}|} \leq \beta_g, \text{ and } \frac{|C_{i^*,g}(\chi)| - m_g^-}{|C_{i^*}|} \geq \alpha_g.$$

For each $i' \in \hat{S}$, let $N(i')$ denote the set of centers $i^* \in S^*$ that are assigned to i' by ϕ'. For each $i' \in \hat{S}$

we can upper bound the proportion of any color as follows:

$$\begin{aligned}\frac{|C_{i',g}| + m_g^+}{|C_{i'}|} &= \frac{\left(\sum_{i^* \in N(i')} |C_{i^*,g}|\right) + m_g^+}{\sum_{i^* \in N(i')} |C_{i^*}|} \\ &\leq \frac{\sum_{i^* \in N(i')} \left(|C_{i^*,g}| + m_g^+\right)}{\sum_{i^* \in N(i')} |C_{i^*}|} \\ &\leq \max_{i^* \in N(i')} \frac{|C_{i^*,g}| + m_g^+}{|C_{i^*}|} \leq \beta_g\end{aligned}$$

Similarly, we can also lower bound the proportions:

$$\begin{aligned}\frac{|C_{i',g}| - m_g^-}{|C_{i'}|} &= \frac{\left(\sum_{i^* \in N(i')} |C_{i^*,g}|\right) - m_g^-}{\sum_{i^* \in N(i')} |C_{i^*}|} \\ &\geq \frac{\sum_{i^* \in N(i')} \left(|C_{i^*,g}| - m_g^-\right)}{\sum_{i^* \in N(i')} |C_{i^*}|} \\ &\geq \min_{i^* \in N(i')} \frac{|C_{i^*,g}| - m_g^-}{|C_{i^*}|} \geq \alpha_g\end{aligned}$$

Note that the above inequalities follow from the Fact A.0.1 since for any center $i^* \in N(i')$, $\frac{|C_{i^*,g}| - m_g^-}{|C_{i^*}|} \geq \alpha_g$ and $\frac{|C_{i^*,g}| + m_g^+}{|C_{i^*}|} \leq \beta_g$.

Furthermore, note that from the definition of ϕ' we know that $|N(i')| \geq 1$ for any i' in $\hat{S}$, i.e., there exists at least one cluster C_{i^*} in the optimal clustering that is assigned to i'. This concludes that each center $i' \in S$ is indeed *robust fair*, i.e., satisfies the constraints (7.10a) and (7.10b). □

We introduce the feasibility linear program (LP) which takes a radius value R and a collection of centers S and assigns the points in $\mathcal{P}$ to centers in S. Since in a given fixed instance the set of centers

S and radius value R can vary as inputs, we call it LP(S, R), its full details are shown below.

$$\text{LP}(S, R):$$

$$\forall j \in \mathcal{P} : \sum_{i \in S} x_{i,j} = 1 \tag{7.17}$$

$$\forall i \in S, g \in \mathcal{G} : \sum_{j \in \mathcal{P}_g} x_{i,j} + m_g^+ \leq \beta_g \sum_{j \in \mathcal{P}} x_{i,j} \tag{7.18}$$

$$\forall i \in S, g \in \mathcal{G} : \sum_{j \in \mathcal{P}_g} x_{i,j} - m_g^- \geq \alpha_g \sum_{j \in \mathcal{P}} x_{i,j} \tag{7.19}$$

$$\forall i \in S, j \in \mathcal{P} : \text{ if } d(i,j) > 3R\text{: } x_{i,j} = 0, \text{else:} x_{i,j} \geq 0 \tag{7.20}$$

For each point $j \in \mathcal{P}$ and center $i \in S$, LP(S, R) has a decision variable $x_{i,j} \in [0, 1]$ which denotes the fractional assignment of point j to center i in S. LP(S, R) is more easily interpreted by considering the integral values of $x_{i,j} \in \{0, 1\}$ instead of $[0, 1]$. Therefore, constraint (7.17) simply states that each point should be assigned to exactly one center. Further, it follows that $\sum_{j \in \mathcal{P}_g} x_{i,j} = |C_{i,g}(\chi)|$ and $\sum_{j \in \mathcal{P}} x_{i,j} = |C_i|$, hence constraint (7.18) is simply imposing constraint (7.10a) of Lemma 7.5.1 to ensure that the upper proportion bounds are not violated. A similar reasoning follows for constraint (7.19). The last constraint (7.20) simply forbids assigning points j to centers i that are at a distance greater than $3R$, this is done by setting the assignment variables $x_{i,j} = 0$ if the distance $d(i, j) > 3R$ and otherwise allowing it to be in $[0, 1]$. Note that the LP(S, R) receives R as an input parameter but uses $3R$ in constraint (7.20).

LP(S, R) is elaborate and in fact it is not difficult to see that it might not be feasible for an arbitrary set of centers S and an arbitrary radius value R. Interestingly, we show that if GETCENTERS is run at a value of $R \geq R^*$ where R^* is the optimal radius (clustering cost) value then the set of centers

$\hat{S}$ returned by GETCENTERS satisfies this LP at radius R, i.e., LP$(\hat{S}, R)$ is *feasible*. This is is shown in the following lemma. In fact, the lemma additionally shows that the number of centers in $\hat{S}$ is at most k, i.e., guaranteeing that we would not have more than k centers. The main idea in the proof is to show that an optimal robust fair solution (S^*, ϕ^*) of cost R^* can instead use the centers $\hat{S}$ at the expense of degrading the clustering cost to $3R^*$. This is done by moving clusters in (S^*, ϕ^*) to carefully chosen centers in $\hat{S}$. Note that the proof is non-constructive as it assumes knowledge of the optimal solution.

Lemma 7.5.3. *Let the optimal clustering cost (radius) be R^*, then if we set $R \geq R^*$ then* GETCENTERS *(Algorithm 10) returns a set $\hat{S}$ such that (1) $|\hat{S}| \leq k$ and (2) LP$(\hat{S}, R)$ is feasible.*

Proof. We first show that for any $R \geq R^*$ the number of centers in $\hat{S}$ returned by Algorithm 10 is at most k. Each center $i^* \in S^*$ has at most one $i' \in \hat{S}$ from its optimal cluster. This is because any two centers in $\hat{S}$ are separated by a distance strictly greater than $2R$ and therefore no two centers in $\hat{S}$ are selected from the same optimal cluster. Therefore, we have $|\hat{S}| \leq |S^*| \leq k$.

To prove the second part of the lemma, it suffices to show that for any $R \geq R^*$, (i) there exists an assignment $\phi' : \mathcal{P} \to \hat{S}$ that assigns points in $\mathcal{P}$ to the centers in $\hat{S}$ and (ii) the cluster $\phi'^{-1}(i)$ corresponding to each $i \in \hat{S}$ is *robust fair*. In Lemma 7.5.2, we show using a non-constructive proof that for any $R \geq R^*$ there exists a feasible solution $(\hat{S}, \phi')$ to our problem with a cost of at most $3R$ where each of the centers $i \in \hat{S}$ has a cluster that is *robust fair*.

If such a solution $(\hat{S}, \phi')$ exists then it immediately corresponds to feasible solution to the LP, this can be shown as follows: for each point $j \in \mathcal{P}$, set $x_{i,j} = 1$ if $\phi'(j) = i$ and $x_{i,j} = 0$ otherwise. Clearly, this x satisfies the constraints in (7.17). Moreover, each cluster $C_i = \phi'^{-1}(i)$ satisfies the

constraints in (7.10b) and (7.10a) since it is robust fair, i.e.,

$$\forall i \in \hat{S}, g \in \mathcal{G} : |C_i \cap \mathcal{P}_g| - m_g^- \geq \alpha_g |C_i|, \quad \text{and} \quad |C_i \cap \mathcal{P}_g| + m_g^+ \leq \beta_g |C_i|. \tag{7.21}$$

Furthermore, since $|C_i| = \sum_{j \in \mathcal{P}} x_{i,j}$ and $|C_i \cap \mathcal{P}_g| = \sum_{j \in \mathcal{P}_g} x_{i,j}$, the assignment x satisfies the constraints in (7.18) and (7.19) by substituting $|C_i| = \sum_{j \in \mathcal{P}} x_{i,j}$ and $|C_i \cap \mathcal{P}_g| = \sum_{j \in \mathcal{P}_g} x_{i,j}$ in the (7.21) we have

$$\forall i \in \hat{S}, g \in \mathcal{G} : \sum_{j \in \mathcal{P}_g} x_{i,j} - m_g^- \geq \alpha_g \sum_{j \in \mathcal{P}} x_{i,j} \quad \text{and} \quad \sum_{j \in \mathcal{P}_g} x_{i,j} + m_g^+ \leq \beta_g \sum_{j \in \mathcal{P}} x_{i,j}.$$

Further note that by Lemma 7.5.2 that each point is assigned to a center in $\hat{S}$ that is at most at a distance of $3R$ therefore (7.20) is satisfied as well. Therefore, we conclude that for any $R \geq R^*$, there exists a non-empty feasible solution to LP $(\hat{S}, R)$. □

The above suggests that we may run GETCENTERS at different radius values R, by Lemma 7.5.3 once $R \geq R^*$ the returned centers $\hat{S}$ would have at most k centers and since LP$(\hat{S}, R)$ would be feasible by running it we would obtain a feasible assignment. In fact, our algorithm ROBUSTALG (Algorithm 11) does that and uses binary search over the set of pairwise distances to find the smallest value of R where the conditions of Lemma 7.5.3 are satisfied. The issue is that the feasible solution that we would obtain x^{LP} can be fractional, i.e., $\mathrm{x}_{i,j}^{\mathrm{LP}} \in [0, 1]$ and not necessarily $\in \{0, 1\}$. Therefore, we would have to round these fractional values into valid integral ones $\mathrm{x}_{i,j}^{\mathrm{Integ}} \in \{0, 1\}$. The following lemma shows that using the MAXFLOW rounding scheme [5] [12, 13] (see Section 7.6 for more details) we can obtain an integral assignment at no increase to the clustering cost and only for a slight change

[5] In short, in MAXFLOW rounding we solve a network flow instance corresponding to a given clustering instance and fractional LP assignment x^{LP}. Then an integral flow is found and used to construct the rounded integral assignment $\mathrm{x}^{\mathrm{Integ}}$.

in the cluster sizes as shown in Lemma 7.5.4.

Lemma 7.5.4. *Let* x^{Integ} *be the integral assignment that results from running* MAXFLOW *rounding over a fractional assignment* x^{LP}*, then (1) if* $\mathrm{x}^{LP}_{i,j} = 0$ *then* $\mathrm{x}^{Integ}_{i,j} = 0$ *and (2) for any center* $i \in \hat{S}$ *and group* $g \in \mathcal{G}$*,*

$$\left\lfloor |C_i^{LP}| \right\rfloor \leq |C_i^{Integ}| \leq \left\lceil |C_i^{LP}| \right\rceil,$$

$$\left\lfloor |C_{i,g}^{LP}| \right\rfloor \leq |C_{i,g}^{Integ}| \leq \left\lceil |C_{i,g}^{LP}| \right\rceil$$

where $|C_i^{LP}| = \sum_{j \in \mathcal{P}} \mathrm{x}^{LP}_{i,j}$, $|C_i^{Integ}| = \sum_{j \in \mathcal{P}} \mathrm{x}^{Integ}_{i,j}$, $|C_{i,g}^{LP}| = \sum_{j \in \mathcal{P}_g} \mathrm{x}^{LP}_{i,j}$, *and* $|C_{i,g}^{Integ}| = \sum_{j \in \mathcal{P}_g} \mathrm{x}^{Integ}_{i,j}$

The fact that the clustering cost would not increase should be clear from guarantee (1) of the above lemma as it implies that $\mathrm{x}^{\text{Integ}}$ will only assign points j to centers i where they already had a non-zero assignment in the fractional solution, i.e., $\mathrm{x}^{\text{LP}}_{i,j} > 0$. The integral assignment $\mathrm{x}^{\text{Integ}}$ can immediately be used to construct the assignment function $\hat{\phi} : \mathcal{P} \to \hat{S}$. Therefore, our final solution is $(\hat{S}, \hat{\phi})$.

All that remains is the final guarantee on the solution $(\hat{S}, \hat{\phi})$. While it is clear that the radius is at most $3R^*$, the theorem below also shows that the violation in the fairness constraints is also bounded by a small value. Formally, we have the following.

Lemma 7.5.5. $(\hat{S}, \hat{\phi})$ *returned by Algorithm 11 is a* λ*-violating solution where* λ *is at most* $\frac{2}{\sum_{g \in \mathcal{G}} m_g^-}$.

Proof. According to Equation (7.9) a λ-violating solution is only required to satisfy the following reduced constraints.

$$\forall i \in S, g \in \mathcal{G} : \frac{|C_{i,g}(\chi)| + m_g^+}{|C_i|} \leq \beta_g + \lambda \text{ and } \frac{|C_{i,g}(\chi)| - m_g^-}{|C_i|} \geq \alpha_g - \lambda$$

Therefore, for any given clustering $\{C_i\}_{i\in S}$, we have the fairness violation λ as follows,

$$\lambda \leq \max_{g\in\mathcal{G}, i\in S} \left\{ \frac{\alpha_g|C_i| - |C_{i,g}(\chi)| + m_g^-}{|C_i|}, \frac{|C_{i,g}(\chi)| + m_g^+ - \beta_g|C_i|}{|C_i|} \right\}$$

Before we delve into the proof we note that any LP solution x^{LP} that satisfies constraints Equation (7.17)-Equation (7.20), the following holds:

$$\alpha_g|C_i^{\mathrm{LP}}| - |C_{i,g}^{\mathrm{LP}}| + m_g^- \leq 0 \tag{7.22}$$

$$|C_{i,g}^{\mathrm{LP}}| - \beta_g|C_i^{\mathrm{LP}}| + m_g^+ \leq 0 \tag{7.23}$$

Where Inequalities (7.22) and (7.23) follow from constraints (7.19) and (7.18), respectively, by simple algebraic manipulation.

We note further by Observation 7.5.1 that $|C_i^{\mathrm{LP}}|$ satisfies:

$$|C_i^{\mathrm{LP}}| \geq \sum_{g\in\mathcal{G}} m_g^- \tag{7.24}$$

Now, let C_i^{Integ} denote the cluster corresponding to the set of points that are assigned to center

$i \in S$. The fairness violation from the lower bound can be bounded as follows

$$\begin{aligned}
\frac{\alpha_g|C_i^{\text{Integ}}| - |C_{i,g}^{\text{Integ}}| + m_g^-}{|C_i^{\text{Integ}}|} &\leq \frac{\alpha_g(|C_i^{\text{LP}}| + 1) - (|C_{i,g}^{\text{LP}}| - 1) + m_g^-}{\lfloor |C_i^{\text{LP}}| \rfloor} && \text{(by Lemma 7.5.4)} \\
&= \frac{\alpha_g|C_i^{\text{LP}}| - |C_{i,g}^{\text{LP}}| + m_g^-}{\lfloor |C_i^{\text{LP}}| \rfloor} + \frac{\alpha_g + 1}{\lfloor |C_i^{\text{LP}}| \rfloor} \\
&\leq 0 + \frac{\alpha_g + 1}{\lfloor |C_i^{\text{LP}}| \rfloor} && \text{(by Inequality (7.22))} \\
&\leq \frac{\alpha_g + 1}{\sum_{g \in \mathcal{G}} m_g^-} && \text{(by Inequality(7.24) since } \sum_{g \in \mathcal{G}} m_g^- \text{ is an integer)}
\end{aligned}$$

Similarly, we can bound the violation from the upper bound as follows

$$\begin{aligned}
\frac{|C_{i,g}^{\text{Integ}}| + m_g^+ - \beta_g|C_i^{\text{Integ}}|}{|C_i^{\text{Integ}}|} &\leq \frac{(|C_{i,g}^{\text{LP}}| + 1) + m_g^+ - \beta_g(|C_i^{\text{LP}}| - 1)}{\lfloor |C_i^{\text{LP}}| \rfloor} && \text{(by the Lemma 7.5.4)} \\
&= \frac{|C_{i,g}^{\text{LP}}| - \beta_g|C_i^{\text{LP}}| + m_g^+}{\lfloor |C_i^{\text{LP}}| \rfloor} + \frac{1 + \beta_g}{\lfloor |C_i^{\text{LP}}| \rfloor} \\
&\leq 0 + \frac{\beta_g + 1}{\lfloor |C_i^{\text{LP}}| \rfloor} && \text{(by Inequality (7.23))} \\
&\leq \frac{\beta_g + 1}{\sum_{g \in \mathcal{G}} m_g^-} && \text{(by Inequality(7.24) since } \sum_{g \in \mathcal{G}} m_g^- \text{ is an integer)}
\end{aligned}$$

Finally, we have the following bound on λ,

$$\lambda \leq \max_{g \in \mathcal{G}} \left\{ \frac{\alpha_g + 1}{\sum_{g \in \mathcal{G}} m_g^-}, \frac{1 + \beta_g}{\sum_{g \in \mathcal{G}} m_g^-} \right\} < \frac{2}{\sum_{g \in \mathcal{G}} m_g^-}$$

The last inequality is due to the fact that $\alpha_g \leq \beta_g < 1$. □

Theorem 7.5.6. ROBUSTALG *(Algorithm 11) is a* 3*-approximation algorithm for the* ROBUSTFAIR-k-CENTER *with fairness violation* $\lambda = \frac{2}{\sum_{g \in \mathcal{G}} m_g^-}$.

Proof. For any given instance of ROBUSTFAIR-k-CENTER, any non-empty solution $(\hat{S}, \hat{\phi})$ returned

by ROBUSTALG (Algorithm 11) has a cost of at most $3R^*$ guaranteed by the fractional assignment returned by LP $(\hat{S}, \hat{\phi})$ as no point is fractionally assigned to any center at a distance more than $3R^*$. Further, part (1) of Lemma 7.5.4 guarantees that the radius would not increase, therefore the cost would still be at most $3R^*$. Moreover, from Lemma 7.5.5 it follows that $(\hat{S}, \hat{\phi})$ is a λ-violating solution with $\lambda \leq \frac{2}{\sum_{g \in \mathcal{G}} m_g^-}$. This concludes the proof of the theorem. □

From the above theorem it is clear that for large values of $\sum_{g \in \mathcal{G}} m_g^-$ the violations would become smaller. In fact, if $\sum_{g \in \mathcal{G}} m_g^- = \omega(1)$ then $\lambda \to 0$ as $n \to \infty$.

Algorithm 11 ROBUSTALG

Input: An instance of FAIR-k-CENTER, $m_g^+, m_g^-{}_{g \in \mathcal{G}}$.

Output: Clustering of points $(\hat{S}, \hat{\phi})$

1: Perform binary search to find the smallest radius R for which the set S returned by GETCENTERS$(\mathcal{P}, R)$ has at most k centers and LP(S, R) is feasible.
2: We call the set of centers returned at the smallest radius $\hat{S}$ and we call its associated fractional assignment x^{LP}.
3: Round x^{LP} to $\mathrm{x}^{\mathrm{Integ}}$ using MAXFLOW rounding (see full details in Section 7.6).
3: Construct the assignment $\hat{\phi} : \mathcal{P} \to \hat{S}$ as follows, for each $j \in \mathcal{P}$ set $\hat{\phi}(j) = i$ if $\mathrm{x}_{i,j}^{\mathrm{Integ}} = 1$.
4: **return** $(\hat{S}, \hat{\phi})$

7.5.2 Failure When Using a Vanilla Clustering Algorithm

Here we show that LP(S, R) would not be feasible using centers selected by a vanilla clustering algorithm S_{vll}[6]. This is shown in the theorem below. The main idea behind this is that a vanilla clustering algorithm lacks adaptivity and therefore may select too many centers and since the constraints in LP(S, R) (specifically (7.18) and (7.19)) implicitly impose a lower bound on the cluster size there would not be enough points to assign to each center. While closing a subset of centers in S_{vll} might lead to a feasible LP(S_{vll}, R), knowing which ones to close without degrading the clustering cost is not

[6]By a vanilla clustering algorithm we mean one which has some α approximation ratio for the ordinary clustering objective.

straightforward to do. Our algorithm GETCENTERS avoids all of this and gives a simple way to find the set of centers and construct the final clustering solution.

Theorem 7.5.7. *Given a set of centers selected by a vanilla k-center algorithm S_{vll} then for any arbitrarily large R, there may not exist a feasible solution to LP(S_{vll}, R).*

Proof. We prove this theorem using a counter example. Specifically, consider the instance in Figure 7.3 with $n = 4$ points belonging to red and blue groups. Let $k = 2$ and $\alpha_{\text{red}} = \alpha_{\text{blue}} = 1/4$ and $\beta_{\text{red}} = \beta_{\text{blue}} = 3/4$. The noise parameters are set to $m^{+}_{\text{red}} = m^{-}_{\text{red}} = 1$. The vanilla k-center algorithm selects the centers $S_{\text{vll}} = \{s_1, s_2\}$. Consider the lower bound constraints (7.19) in the corresponding LP with $S = S_{\text{vll}}$. This constraint requires that each center s_1 and s_2 must have a fractional assignment strictly greater than 1 from each color. This follows from the lower bound constraints in Equation (7.19) where each center in $s \in S_{\text{vll}}$ and $h \in \{\text{red,blue}\}$ must satisfy

$$|C^{\text{LP}}_{s,g}| \geq m^{-}_{g} + \alpha_g |C^{\text{LP}}_{s}| \geq 1 + \alpha_g |C^{\text{LP}}_{s}| > 1$$

where $|C^{\text{LP}}_{s,g}|$ denotes the fractional assignment of strictly greater than one point of color g to center s and $|C^{\text{LP}}_{s}|$ denotes the fractional assignment of strictly greater than two points to center s. However, both s_1 and s_2 cannot simultaneously be fractionally assigned strictly greater than 2 points each since there are only 4 points in total. Therefore, this shows that there exists no feasible solution to LP (7.17)-(7.20) for any arbitrarily large R.

□

Figure 7.3: Toy example showing that for the set of centers S_{vll} from vanilla k-center algorithm, there does not exist a feasible solution to LP (7.17)-(7.20) for any arbitrarily large R.

7.6 MAXFLOW Rounding

MAXFLOW rounding is given an LP solution satisfying (7.17),(7.18),(7.19), and (7.20). The MAXFLOW rounding we use is identical to the one used in [12, 13, 141]. Our explanation here closely follows that in [13] with some modifications for our setting. Formally, given an LP solution $\mathrm{x}^{\text{LP}} = \{x^{\text{LP}}_{i,j}\}_{i\in\hat{S},j\in\mathcal{P}}$, the network flow diagram is constructed as follows:

1. $V = \{s,t\} \cup \mathcal{P} \cup \{i^h | i \in \hat{S}, g \in \mathcal{G}\} \cup \{i \in \hat{S}\}$.

2. $A = A_1 \cup A_2 \cup A_3 \cup A_4$ where $A_1 = \{(s,j) | j \in \mathcal{P}\}$ with upper bound of 1. $A_2 = \{(j, i^h) | x^{\text{LP}}_{i,j} > 0\}$ with upper bound of 1. The arc set $A_3 = \{(i^h, i) | i \in \hat{S}, g \in \mathcal{G}\}$ with lower bound $\left\lfloor \sum_{j\in\mathcal{P}_g} x^{\text{LP}}_{i,j} \right\rfloor$ and upper bound of $\left\lceil \sum_{j\in\mathcal{P}_g} x^{\text{LP}}_{i,j} \right\rceil$. As for $A_4 = \{(i,t) | i \in \hat{S}\}$ the lower and upper bounds are $\left\lfloor \sum_{j\in\mathcal{P}} x^{\text{LP}}_{i,j} \right\rfloor$ and $\left\lceil \sum_{j\in\mathcal{P}} x^{\text{LP}}_{i,j} \right\rceil$.

By construction of the network flow diagram the maximum flow that can be achieved at the sink t is n (the number of points). Further, the given LP assignment $\mathrm{x}^{\text{LP}} = \{x^{\text{LP}}_{i,j}\}_{i\in\hat{S},j\in\mathcal{P}}$ is a valid fractional flow that achieves a maximum flow of n. Since the upper and lower bound on the arcs are integral, it follows by standard result of the max flow problem that we can find an integral maximum flow

$\mathrm{x}^{\text{Integ}} = \{x_{i,j}^{\text{Integ}}\}_{i \in \hat{S}, j \in \mathcal{P}}$. Moreover, from the set upper and lower bounds the following is immediate:

$$\left\lfloor \sum_{j \in \mathcal{P}} \mathrm{x}_{i,j}^{\text{LP}} \right\rfloor \leq \sum_{j \in \mathcal{P}} \mathrm{x}_{i,j}^{\text{Integ}} \leq \left\lceil \sum_{j \in \mathcal{P}} \mathrm{x}_{i,j}^{\text{LP}} \right\rceil$$
$$\left\lfloor \sum_{j \in \mathcal{P}_g} \mathrm{x}_{i,j}^{\text{LP}} \right\rfloor \leq \sum_{j \in \mathcal{P}_g} \mathrm{x}_{i,j}^{\text{Integ}} \leq \left\lceil \sum_{j \in \mathcal{P}_g} \mathrm{x}_{i,j}^{\text{LP}} \right\rceil$$

Further, recall from Lemma 7.5.4 that $|C_i^{\text{LP}}| = \sum_{j \in \mathcal{P}} \mathrm{x}_{i,j}^{\text{LP}}$, $|C_i^{\text{Integ}}| = \sum_{j \in \mathcal{P}} \mathrm{x}_{i,j}^{\text{Integ}}$, $|C_{i,g}^{\text{LP}}| = \sum_{j \in \mathcal{P}_g} \mathrm{x}_{i,j}^{\text{LP}}$, and $|C_{i,g}^{\text{Integ}}| = \sum_{j \in \mathcal{P}_g} \mathrm{x}_{i,j}^{\text{Integ}}$. Therefore, it follows that we have

$$\left\lfloor |C_i^{\text{LP}}| \right\rfloor \leq |C_i^{\text{Integ}}| \leq \left\lceil |C_i^{\text{LP}}| \right\rceil,$$
$$\left\lfloor |C_{i,g}^{\text{LP}}| \right\rfloor \leq |C_{i,g}^{\text{Integ}}| \leq \left\lceil |C_{i,g}^{\text{LP}}| \right\rceil$$

This proves part (2) of Lemma 7.5.4. Part (1) of Lemma 7.5.4 simply follows from that fact that if $\mathrm{x}_{i,j}^{\text{LP}} = 0$ then there would not be an arc connecting point (vertex) j to vertex $i^h, \forall g \in \mathcal{G}$ and that immediately implies that $\mathrm{x}_{i,j}^{\text{Integ}} = 0$.

7.7 Experiments

7.7.1 Experimental Setup

The experiments are run on Python 3.6.15 on a commodity laptop with a Ryzen 7 5800U and 16GB of RAM. The linear programs (LP) in the algorithms are solved using CPLEX 12.8.0.0 [189] and flow problems are solved using `NetworkX` 2.5.1 [190]. In total, the experiments in Figure 7.4 solve 53 fair clustering instances (30 robust fair instances, 20 probabilistic fair, and 3 deterministic fair). PROBALG is not run on **Census1990** because its theoretical guarantees hold for the two-color

setting only. DETALG has a single fair clustering instance per dataset because DETALG does not depend on m. Our code implementation forks the code of [13].

Datasets. We experiment on three datasets from the UCI repository [168]: **Adult**, **Bank**, and **Census1990**. The datasets have $32k, 32k$ and $4.5k$ points with 5, 3, and 66 numerical features, respectively. Further, the colors in each dataset, i.e., sensitive attributes, are respectively a binary sex, a binary marital status, and a membership in one of three age buckets. The distances between any pair of points is set to the Euclidean distance between their normalized numerical features.

Parameters. The experiments use noise parameters $m_g^+ = m_g^- = m, \forall g \in \mathcal{G}$, where m ranges from $\frac{n}{100}$ to $\frac{n}{10}$. We set the proportions α_g and β_g to the respectively greatest and least values leading to feasible ROBUSTFAIR-k-CENTER instances, in accordance with Observation 7.5.2. Therefore, α_g and β_g are the same across instances in the same dataset. The number of centers is fixed at 10, i.e., $k = 10$.

We benchmark our proposed algorithm ROBUSTALG against two relevant baselines. Namely, *probabilistic* fair clustering [58] and *deterministic* fair clustering [8]. These three fair clustering algorithms are denoted as ROBUSTALG, PROBALG, and DETALG in the plots. Note that probabilistic fair clustering is an algorithm for two colors only so it is tested solely on **Adult** and **Bank**.

We evaluate the algorithms on their attained k-center objectives and fairness violation given by (7.9). For deterministic (and robust) fair clustering we obtain fairness violations by directly finding the corruption of point colors leading to the greatest fairness violation as described in Inequalities (7.11a) and (7.11b). Specifically, for each color, up to m points can essentially be added or subtracted. Over half of ROBUSTALG's pictured fairness violations are exactly zero; the rest fall between 10^{-5} and 10^{-3}. The 95% confidence interval around PROBALG is shaded; however, it is faint because the fairness violations are very sharply concentrated around their plotted mean.

In the probabilistic fair clustering work of Esmaeili et al. [58], the noise model is different. Specifically, each point's color is corrupted with some probability. To ensure that our evaluation is fair we evaluate the probabilistic instances under their assumed noise model. Instances are set up so that each point's color is corrupted with probability m/n, leading to an *expected* m corruptions dataset-wide. In contrast, robust and deterministic fair clustering allow for up to m corruptions in *each color*. Finally, we sample 200 realizations of colors and directly report the mean fairness violations. However, probabilistic fair clustering says nothing about point correlations. We exploit this fact as follows. First, sample $S_{i,g} \sim \text{Bernoulli}(m/n)$ for each cluster i and color g, and then, if $S_{i,g} = 1$, corrupt the colors of all points of color g assigned cluster i. Note that this is a generous evaluation, since an adversarial color assignment (as done in deterministic and robust fair clustering) can only lead to a higher violation.

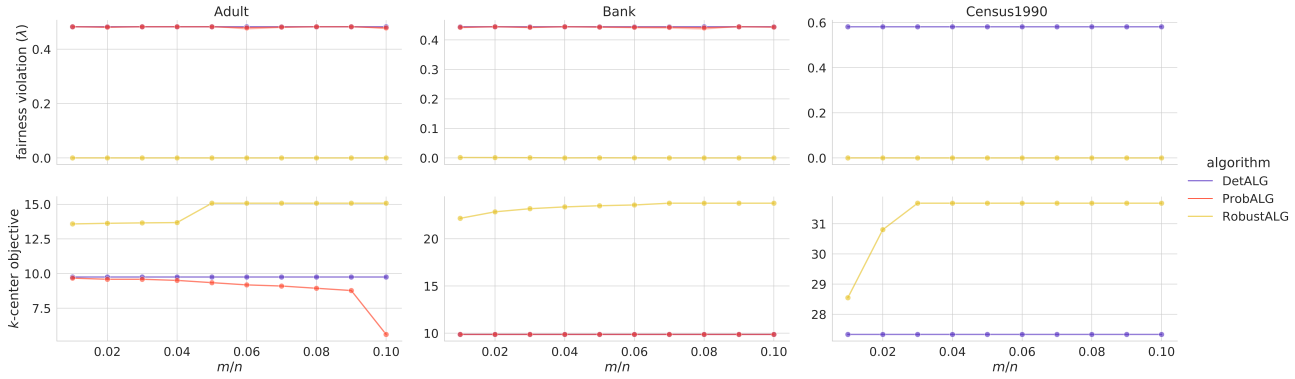


Figure 7.4: Plots for k-center objective and fairness violation as m/n increases.

Figure 7.4 shows the results of our experiments. RobustAlg *maintains fairness violations of zero and near-zero* across the board; unlike deterministic and probabilistic baselines which have fairness violations as large as $0.6 = 60\%$. In fact, in DetAlg the post-corruption ratio of some colors becomes 0 in **Adult** and in **Bank** (i.e., a color is completely absent from the cluster) or 1 in **Census1990** (i.e., a color is fully dominating the cluster). Note that these are the worst-possible fairness violations

in all three datasets. PROBALG nearly hits these worst-case violations as well.

The objective (clustering cost) of ROBUSTALG is greater and increases with m/n, as expected when targeting a more stringent notion of robustness. The break in the objective plot of Figure 7.4 occurs when ROBUSTALG opens fewer centers. As m increases, centers must have a greater number of points and thus fewer centers can receive points; otherwise applying all m corruptions to the smallest center results in (nearly) absent colors or (nearly) monochromatic centers, which produce large fairness violations. Figure 7.5 recreates the experiments in Figure 7.4 but with values of m one order of magnitude smaller and $k = 5$. These plots show greater variance in the fairness violations of PROBALG, which makes sense given the smaller m and that, in probabilistic fair clustering, the probability a point's label is correct is $p_{\text{acc}} = 1 - m/n$.

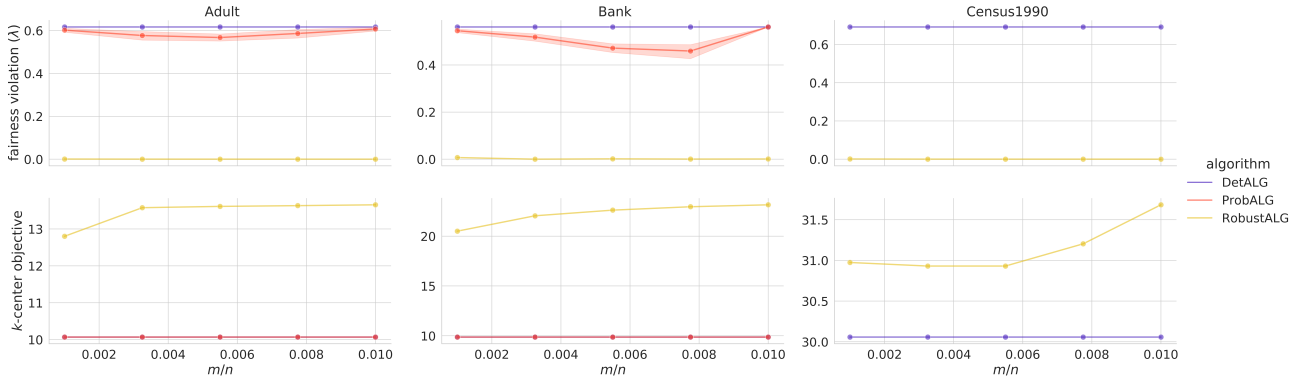


Figure 7.5: A repeat of Figure 7.4 but for smaller values of m and $k = 5$ instead of $k = 10$.

Finally, Figure 7.6 concludes with experiments on **Bank** using three settings of noise parameters. We fix the α_g and β_g for all plots as described before. We denote the two colors in bank by 0 and 1. Further, we take m from $\frac{1}{1000}n$ to $\frac{1}{100}n$ but consider the (m_0^+, m_1^+) choices of $(m, m/2)$, $(m/2, m)$, and (m, m). The experiments validate our intuitions. First, (m, m) achieves the highest objective as it has the most corruptions (largest uncertainty set). Moreover, $(m/2, m)$ and $(m, m/2)$ both have the same number of total corruptions and their objectives are indeed comparable. In all cases, the fairness

violations border on zero, agreeing with Theorem 7.5.6.

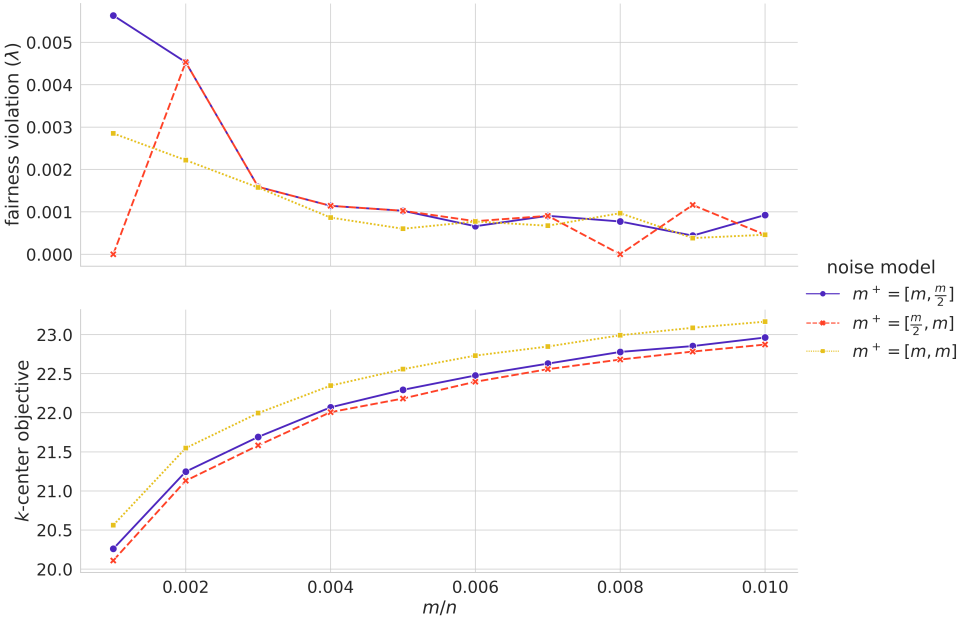


Figure 7.6: Comparison of the effect of increasing noise, given different noise model configurations.

7.7.2 Running Times of the Experiments

Figure 7.7 shows RobustAlg outperforms the baselines in both of the larger datasets (**Adult** and **Census1990**); this is not the case in **Bank**, but the difference is at most 20 seconds there. In each step of the binary search, RobustAlg has an additional step of GetCenters which contributes to the running time. However, the running time of these algorithms is largely dominated by solving the associated LPs. However, the LPs for the baselines always use k centers and thus have nk variables. However, RobustAlg uses the $k' \leq k$ centers selected by GetCenters. In practice, this can be a substantial speedup as the difference between k' and k increases and as n increases. This behavior is also observed in **Diabetes** dataset.

Related to this phenomenon, as m grows, the binary search in RobustAlg discards smaller

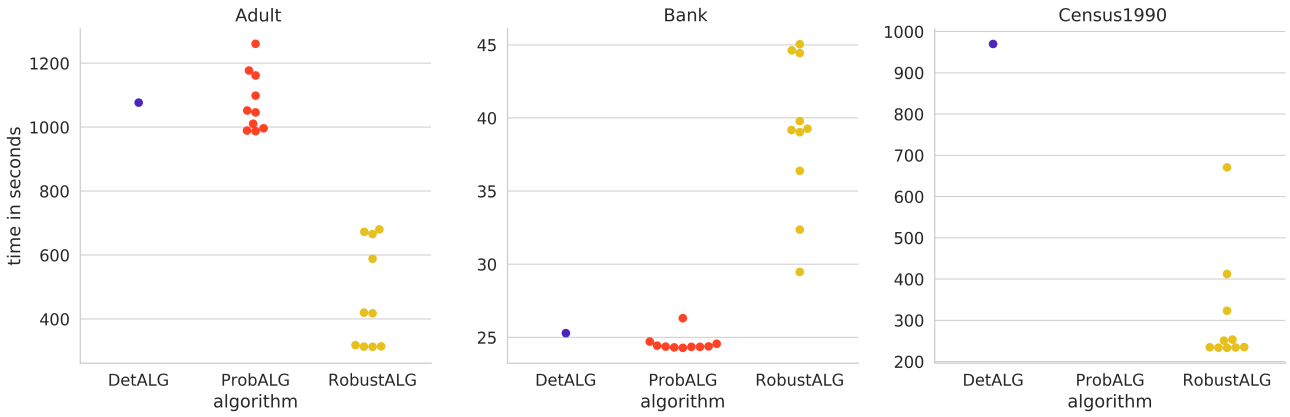


Figure 7.7: Wall clock time each algorithm took to solve the associated {deterministic, probabilistic, robust} fair clustering instances in Figure 7.4.

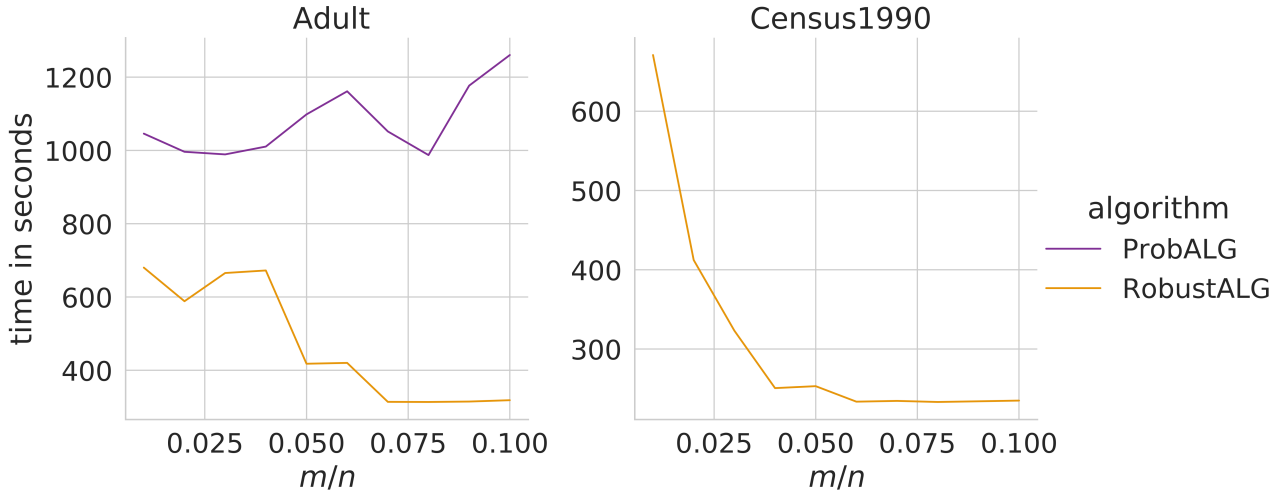


Figure 7.8: Running times of PROBALG and ROBUSTALG from Figure 7.7 but plotted against m.

values of R because more LPs become infeasible. Therefore, the binary search moves onto greater values of R, for which GETCENTERS selects fewer centers. Indeed, in Figure 7.8 we can observe ROBUSTALG speeding up as m increases. On the other hand, and as expected, the running time of PROBALG does not exhibit this interaction with m.

7.7.3 Additional Experiments on the **Diabetes** dataset

We use an additional dataset **Diabetes** to supplement our running time experiments. Like our other three datasets, **Diabetes** is also from the UCI repository. It has 49 features and we set up two colors corresponding to whether the patient (i.e., point) was or was not female. Figure 7.9 shows the dependence of all three algorithms on k, and we can also see that the baselines are more sensitive to larger k. These plots use the first n' points of **Diabetes** for n' in $\{2000, 4000, 6000, \ldots, 23000\}$. For each instance we take $m = 0.005n'$. The proportionality constants α_g and β_g are set to be feasible for RobustAlg exactly in the same way as for other datasets

We also recreate the fairness violation and objective plots from Figure 7.4 on **Diabetes** as shown in Figure 7.10. For these experiments we fix $k = 5$ and $\forall g$, $\alpha_g = 0.3$ and $\beta_g = 0.75$. A moment's reflection shows the worst fairness violation possible is 0.3. Indeed, DetAlg and ProbAlg, respectively, exactly reach and get very close to this violation. The objectives also only differ by at most a meager 4 units of distance.

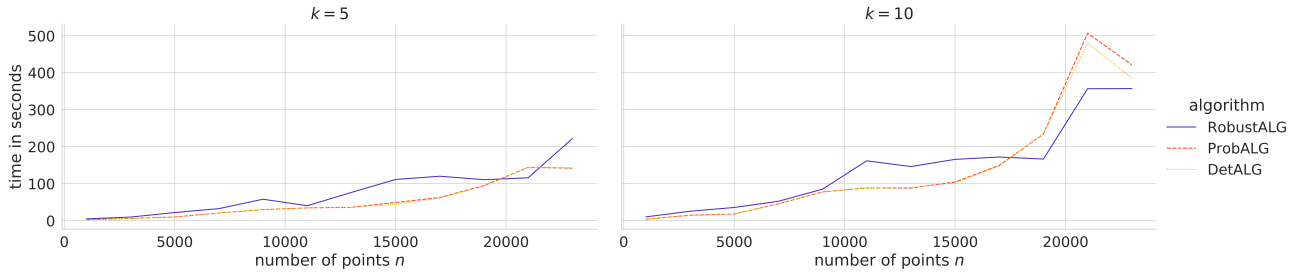


Figure 7.9: Running times of RobustAlg, ProbAlg, and DetAlg on dataset **Diabetes** as the number of points n increases. We run the experiments for number of clusters k of 5 and 10.

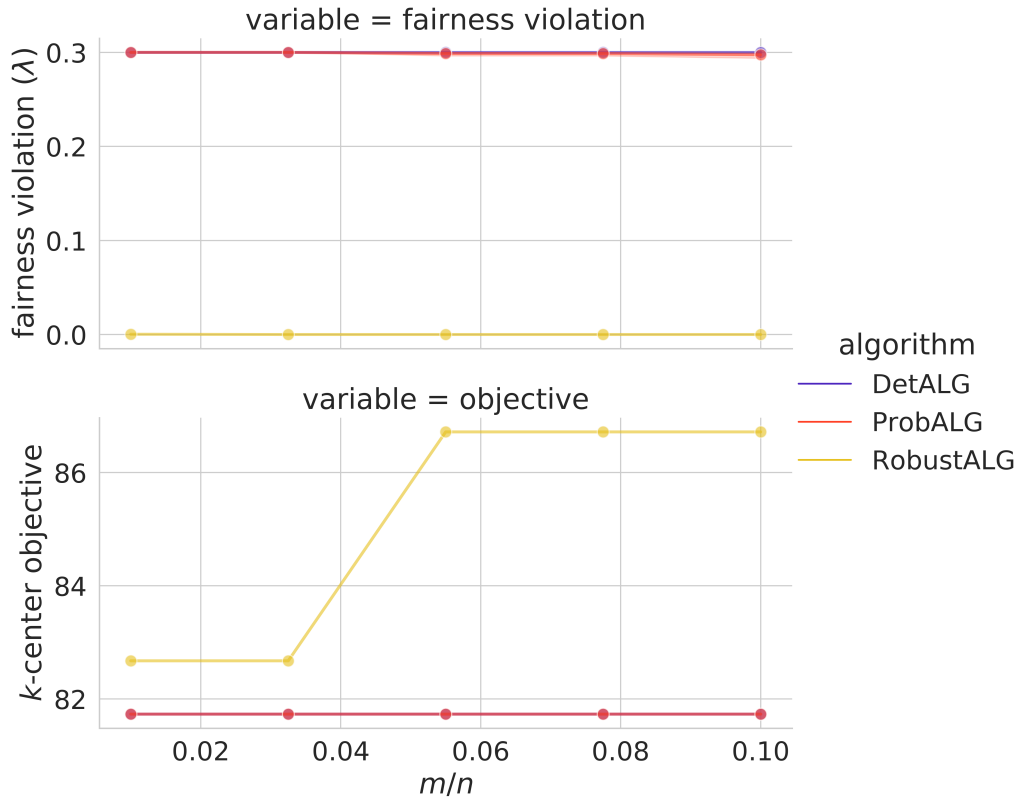


Figure 7.10: Plots of k-center objective and fairness violations on **Diabetes**.

7.8 More Discussion About Previous Noise Models in Fair Clustering

7.8.1 Drawbacks of the Noise Model of Probabilistic Fair Clustering

We provide an example to show the drawbacks of the probabilistic model introduced in Esmaeili et al. [58]. Their theoretical guarantees for robust fair clustering only guarantee to satisfy the fairness constraints in expectation. However, the realizations can be arbitrarily unfair. We illustrate this using an example. Consider a set of 14 points as shown in the Figure 7.11 where each point is assigned to either the red or blue group, each with a probability of $1/2$. Any clustering of these points is fair in expectation assuming the lower and upper bounds for both the blue and red group are close to $\frac{1}{2}$. However individual realizations can be unfair. Specifically, since the joint probability distribution

is not known (in fact, it is not incorporated at all in the probabilistic model of Esmaeili et al. [58]) the realizations could be as shown in the figure where all points in a cluster take on the same color simultaneously in a realization. This shows that the probabilistic model may return clusters that can be fair (proportional) in expectation but completely unfair (unproportional) in realization.

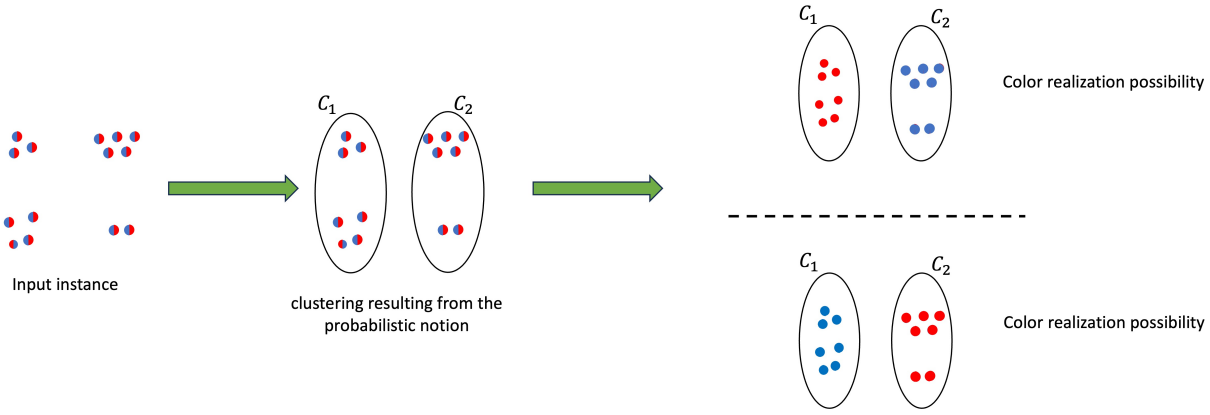


Figure 7.11: An instance of FAIR-k-CENTER with 14 points where each point has the probabilities $p_j^{\text{red}} = p_j^{\text{blue}} = \frac{1}{2}$. Here p_j^{red} and $p_j^{\text{blue}} \forall j \in \mathcal{P}$ denote the probability that point j belong to red and blue groups respectively. Note how the probabilistic fair clustering satisfies fairness in expectation. However, the two realizations shown demonstrate that color proportionality can be completely violated.

7.8.2 More Discussion About the Noise Model Chhabra et al.

We provide an example to show the drawbacks of the noise model introduced by Chhabra et al. [59]. Their noise model assumes that only a subset of the points are affected by the adversary but they do not specify how one can access this subset. In their experiments, they generate this subset using random sampling. Specifically, they independently sample each point with probability 0.15 to obtain a subset of points $\mathcal{P}' \subseteq \mathcal{P}$. However, we can easily construct examples where their random sampling method with probability ≈ 0.99 can never return some subsets. We illustrate this using an example. Consider an instance of FAIR-k-CENTER as shown in Figure 7.12 where only a subset of points have incorrect memberships according to Chhabra et al. [59]. Their sampling process selects a

subset (comprising 10% of the points) by randomly sampling each point independently with a probability of 0.1. As a result, out of 160 points 16 points are perturbed in expectation. However, this does not capture scenarios where all perturbations occur within a subset of points (as shown in the left side of Figure 7.12) as the probability of such an event is close to 0, precisely 0.0002. On the other hand, our model considers for all possible subsets with 16 points. As a result, we can model the scenarios where all the incorrect memberships occur in a single group or more generally all possible combinations across the two groups.

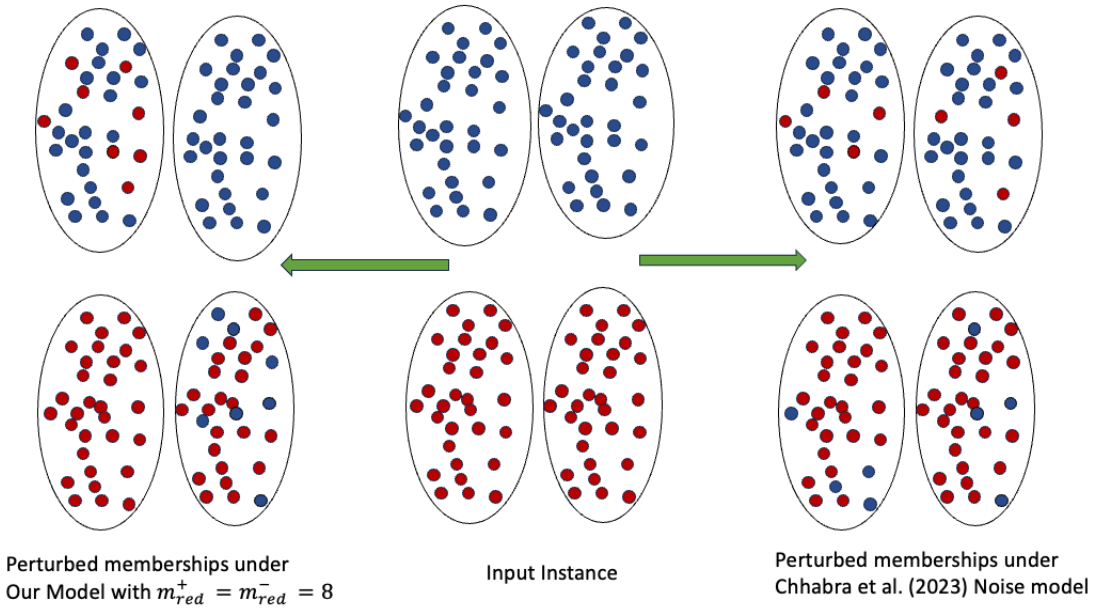


Figure 7.12: Inst

Consider an instance of FAIR-k-CENTER with $n = 160$ points, where 10% of the points have incorrect memberships. Chhabra et al. [59] perturbs a subset (comprising 10% of the points) by randomly sampling each point independently with a probability of 0.1. As a result 16 points are perturbed in expectation. However, this does not capture scenarios where all perturbations occur within a subset of points as shown on the left side. This is because the probability of such an event is ≈ 0. On the other hand, our model considers for all possible subsets with 16 points, thereby covering all possible

scenarios.

Furthermore, in terms of algorithms [59] also provide a defense algorithm by a Consensus Clustering method via k-means (Lloyd's algorithm) combined with fair constraints to achieve robustness against their proposed attack. Their algorithm trains a neural network using a loss function based on pair-wise similarity information obtained by running Consensus k-means clustering, which is different from our k-center objective. Moreover, their *fair clustering loss* fails to include the fractional proportional bounds present in our fair clustering instances. Therefore, their algorithm is inapplicable to our problem. Further, their algorithm provide no theoretical or empirical guarantees on the distance-based clustering cost. Finally, their fairness loss does not model the fairness constraints based on proportional lower and upper bounds or any other parameters provided by the stakeholder. More importantly, their robust algorithms have no theoretical guarantees on ex-post fairness violations, whereas ours do.

Chapter 8: Conclusion and Future Work

In Chapter 3, we present a randomized algorithm for offline bipartite graphs where the input graph and group assignments are fully available. There are two exciting directions for future work. First, in contrast to our current algorithm which processes vertices in a fixed or arbitrary order, Random Order Contention Resolution Scheme (RCRS) from [191] provides better selection guarantees where vertices are processed in a random order. More precisely, RCRS achieves an approximation ratio of $1 - \frac{1}{e}$. However, it appears to be much more challenging to compute the one-step variance of the martingale at each time t, making it difficult to derive a *useful bound* for the sum of variances $\sum_{t\in[n]} \mathrm{Var}(|\Delta M_t| \mid \mathcal{H}_{t-1})$. Therefore, it remains an interesting open problem to determine if we can overcome this obstacle to improve the approximation factor from $1/2$ to $1 - \frac{1}{e}$. Moreover, adapting the analysis to general graphs is also an interesting future direction.

Second, building upon the foundations of this work, an interesting next step is to extend our analysis to online vertex arrival models also discussed in Chapter 3. In these settings, the offline set of vertices (e.g., resources) is known a priori, but the online vertices (e.g., users) arrive one by one, where it is required to make immediate and irrevocable matching decisions. Given that our existing algorithm (Algorithm 4 presented in Chapter 3) is inherently online[1], we are working on extending the algorithm to handle online models such as the KnownIID and KnownAD settings discussed in

[1] processes the vertices in a sequential manner

Chapter 2.

In Chapter 7, we introduced a novel noise model to capture uncertainty in group memberships within fair clustering. This model can be generalized to other problems. For example, it can be readily extended to the problems addressed in this thesis, such as bipartite matching and set packing as well to a broader class of constrained optimization problems. Moving forward, we present two primary research directions for future work. First, from a statistical perspective, the estimation of noise parameters remains a critical challenge. While we assume that these are given, the overall quality of the clustering and the robustness guarantees are dependent on the group-level noise parameters (e.g., m_g^+ and m_g^- as shown in Figure 7.1). Therefore, to make this this framework viable, it is important to develop efficient methods to estimate these noise parameters (μ_{hg} and μ_{gh}) and integrate them with our algorithms to provide an end-to-end solution for maintaining group fairness in the presence of noisy group memberships.

Second, understanding the applicability of our noise model to other fundamental problems such as bipartite matching, set packing, and set covering is another interesting open direction. Expanding into these areas is essential for understanding the individual challenges that arise when group membership is uncertain, especially, the trade-off between robust-fairness and underlying objective.

Appendix A: Useful Facts and Inequalities

Fact A.0.1. *For any positive real numbers* $a_1, a_2, \ldots, a_n$ *and* $b_1, b_2, \ldots, b_n$*, the following holds*

$$\min_{i\in[n]} \frac{a_i}{b_i} \leq \frac{a_1 + a_2 + \cdots + a_n}{b_1 + b_2 + \cdots + b_n} \leq \max_{i\in[n]} \frac{a_i}{b_i} \tag{A.1}$$

Proof. Let $\tau_{\max} = \max_{i\in[n]} \frac{a_i}{b_i}$, therefore we have

$$\frac{a_1 + a_2 + \cdots + a_n}{b_1 + b_2 + \cdots + b_n} \leq \frac{\tau_{\max}(b_1 + b_2 + \cdots + b_n)}{b_1 + b_2 + \cdots + b_n} = \tau_{\max}$$

The lower bound can be proved similarly. □

Bibliography

[1] Emilio Ferrara. Fairness and bias in artificial intelligence: A brief survey of sources, impacts, and mitigation strategies. *Sci*, 6(1):3, 2024.

[2] Jennifer L Skeem and Christopher T Lowenkamp. Risk, race, and recidivism: Predictive bias and disparate impact. *Criminology*, 54(4):680–712, 2016.

[3] Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464):447–453, 2019.

[4] David García-Soriano and Francesco Bonchi. Maxmin-fair ranking: individual fairness under group-fairness constraints. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 436–446, 2021.

[5] Vedant Nanda, Pan Xu, Karthik Abinav Sankararaman, John P Dickerson, and Aravind Srinivasan. Balancing the tradeoff between profit and fairness in rideshare platforms during high-demand hours. In *AAAI*, 2020.

[6] Jacob Abernethy, Pranjal Awasthi, Matthäus Kleindessner, Jamie Morgenstern, Chris Russell, and Jie Zhang. Active sampling for min-max fairness. *arXiv preprint arXiv:2006.06879*, 2020.

[7] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair clustering through fairlets. *NeurIPS*, 30, 2017.

[8] Suman Bera, Deeparnab Chakrabarty, Nicolas Flores, and Maryam Negahbani. Fair algorithms for clustering. *Advances in Neural Information Processing Systems*, 32, 2019.

[9] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvtiskii. Matroids, matchings, and fairness. In *The 22nd international conference on artificial intelligence and statistics*, pages 2212–2220. PMLR, 2019.

[10] 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.

[11] Jad Salem, Reuben Tate, and Stephan Eidenbenz. Expected maximin fairness in max-cut and other combinatorial optimization problems. *arXiv preprint arXiv:2410.02589*, 2024.

[12] Ioana O Bercea, Martin Groß, Samir Khuller, Aounon Kumar, Clemens Rösner, Daniel R Schmidt, and Melanie Schmidt. On the cost of essentially fair clusterings. *arXiv preprint arXiv:1811.10319*, 2018.

[13] John Dickerson, Seyed A Esmaeili, Jamie Morgenstern, and Claire Jie Zhang. Doubly constrained fair clustering. *NeurIPS*, 2023.

[14] Ji Wang, Ding Lu, Ian Davidson, and Zhaojun Bai. Scalable spectral clustering with group fairness constraints. In *International conference on artificial intelligence and statistics*, pages 6613–6629. PMLR, 2023.

[15] Pengxin Zeng, Yunfan Li, Peng Hu, Dezhong Peng, Jiancheng Lv, and Xi Peng. Deep fair clustering via maximizing and minimizing mutual information: Theory, algorithm and metric. In *Computer Vision and Pattern Recognition Conference (CVPR)*, pages 23986–23995, June 2023.

[16] Sayan Bandyapadhyay, Fedor V Fomin, Tanmay Inamdar, and Kirill Simonov. Proportionally fair matching with multiple groups. *arXiv preprint arXiv:2301.03862*, 2023.

[17] Chien-Ju Ho and Jennifer Vaughan. Online task assignment in crowdsourcing markets. In *AAAI*, 2012.

[18] Yongxin Tong, Jieying She, Bolin Ding, Lei Chen, Tianyu Wo, and Ke Xu. Online minimum matching in real-time spatial data: experiments and analysis. *Proceedings of the VLDB Endowment*, 9(12):1053–1064, 2016.

[19] John P Dickerson, Karthik Abinav Sankararaman, Aravind Srinivasan, and Pan Xu. Balancing relevance and diversity in online bipartite matching via submodularity. In *AAAI*, 2019.

[20] Meghna Lowalekar, Pradeep Varakantham, and Patrick Jaillet. Online spatio-temporal matching in stochastic and dynamic domains. *Artificial Intelligence (AIJ)*, 261:71–112, 2018.

[21] John P Dickerson, Karthik A Sankararaman, Aravind Srinivasan, and Pan Xu. Allocation problems in ride-sharing platforms: Online matching with offline reusable resources. *ACM Transactions on Economics and Computation (TEAC)*, 9(3):1–17, 2021.

[22] Will Ma, Pan Xu, and Yifan Xu. Fairness maximization among offline agents in online-matching markets. In *WINE*, 2021.

[23] Gagan Goel and Aranyak Mehta. Online budgeted matching in random input models with applications to adwords. In *SODA*, 2008.

[24] Aranyak Mehta. Online matching and ad allocation. *Foundations and Trends in Theoretical Computer Science*, 8(4):265–368, 2013.

[25] Seyed Esmaeili, Sharmila Duppala, Davidson Cheng, Vedant Nanda, Aravind Srinivasan, and John P Dickerson. Rawlsian fairness in online bipartite matching: Two-sided, group, and individual. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 5624–5632, 2023.

[26] Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. Adwords and generalized online matching. *Journal of the ACM*, 54, no. 5, 2007. URL `http://doi.acm.org/10.1145/1284320.1284321`.

[27] Aranyak Mehta. Online matching and ad allocation. *Foundations and Trends in Theoretical Computer Science*, 8 (4):265–368, 2013. URL `http://dx.doi.org/10.1561/0400000057`.

[28] Chien-Ju Ho and Jennifer Vaughan. Online task assignment in crowdsourcing markets. *Proceedings of the AAAI Conference on Artificial Intelligence*, 26(1):45–51, Sep. 2021. doi: 10.1609/aaai.v26i1.8120. URL https://ojs.aaai.org/index.php/AAAI/article/view/8120.

[29] Yuya Hikima, Yasunori Akagi, Hideaki Kim, Masahiro Kohjima, Takeshi Kurashima, and Hiroyuki Toda. Integrated optimization of bipartite matching and its stochastic behavior: New formulation and approximation algorithm via min-cost flow optimization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(5):3796–3805, May 2021. doi: 10.1609/aaai.v35i5.16497. URL https://ojs.aaai.org/index.php/AAAI/article/view/16497.

[30] Manish Purohit, Sreenivas Gollapudi, and Manish Raghavan. Hiring under uncertainty. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5181–5189. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/purohit19a.html.

[31] John P Dickerson, Ariel D Procaccia, and Tuomas Sandholm. Failure-aware kidney exchange. In *Proceedings of the fourteenth ACM conference on Electronic commerce*, pages 323–340, 2013.

[32] Duncan C McElfresh, Hoda Bidkhori, and John P Dickerson. Scalable robust kidney exchange. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):1077–1084, Jul. 2019. doi: 10.1609/aaai.v33i01.33011077. URL https://ojs.aaai.org/index.php/AAAI/article/view/3899.

[33] Golnoosh Farnadi, William St-Arnaud, Behrouz Babaki, and Margarida Carvalho. Individual fairness in kidney exchange programs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(13):11496–11505, May 2021. doi: 10.1609/aaai.v35i13.17369. URL https://ojs.aaai.org/index.php/AAAI/article/view/17369.

[34] Alireza Farhadi, Jacob Gilbert, and MohammadTaghi Hajiaghayi. Generalized stochastic matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10008–10015, 2022.

[35] Vedant Nanda, Pan Xu, Karthik Abhinav Sankararaman, John Dickerson, and Aravind Srinivasan. Balancing the tradeoff between profit and fairness in rideshare platforms during high-demand hours. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(02):2210–2217, Apr. 2020. doi: 10.1609/aaai.v34i02.5597. URL https://ojs.aaai.org/index.php/AAAI/article/view/5597.

[36] John Dickerson, Karthik Sankararaman, Aravind Srinivasan, and Pan Xu. Allocation problems in ride-sharing platforms: Online matching with offline reusable resources. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018. doi: 10.1609/aaai.v32i1.11477. URL https://ojs.aaai.org/index.php/AAAI/article/view/11477.

[37] Sharmila Duppala, Nathaniel Grammel, Juan Luque, Calum MacRury, and Aravind Srinivasan. Proportionally fair matching via randomized rounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 16435–16443, 2025.

[38] John P. Dickerson, Ariel D. Procaccia, and Tuomas Sandholm. Price of fairness in kidney exchange. In *Int. Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2014.

[39] Jian Li, Yicheng Liu, Lingxiao Huang, and Pingzhong Tang. Egalitarian pairwise kidney exchange: fast algorithms vialinear programming and parametric flow. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems (AAMAS)*, pages 445–452, 2014.

[40] Golnoosh Farnadi, William St-Arnaud, Behrouz Babaki, and Margarida Carvalho. Individual fairness in kidney exchange programs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11496–11505, 2021.

[41] Zhaohong Sun, Taiki Todo, and Toby Walsh. Fair pairwise exchange among groups. In *Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 419–425, 2021.

[42] Alvin E Roth, Tayfun Sönmez, and M Utku Ünver. Pairwise kidney exchange. *Journal of Economic Theory (JET)*, 125(2):151–188, 2005.

[43] Itai Ashlagi, Maximilien Burq, Patrick Jaillet, and Vahideh Manshadi. On matching and thickness in heterogeneous dynamic markets. *Operations Research*, 67(4):927–949, 2019.

[44] Itai Ashlagi, Duncan S Gilchrist, Alvin E Roth, and Michael A Rees. Nonsimultaneous chains and dominos in kidney-paired donation—revisited. *American Journal of Transplantation (AJT)*, 11(5):984–994, 2011.

[45] Duncan C McElfresh and John P Dickerson. Balancing lexicographic fairness and a utilitarian objective with application to kidney exchange. In *AAAI*, 2018.

[46] Péter Biro, David F Manlove, and Romeo Rizzi. Maximum weight cycle packing in directed graphs, with application to kidney exchange programs. *Discrete Mathematics, Algorithms and Applications*, 1(04):499–517, 2009.

[47] Mugang Lin, Jianxin Wang, Qilong Feng, and Bin Fu. Randomized parameterized algorithms for the kidney exchange problem. *Algorithms*, 12(2):50, 2019.

[48] Sharmila Duppala, Juan Luque, John Dickerson, and Aravind Srinivasan. Group fairness in set packing problems. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, IJCAI '23, 2023. ISBN 978-1-956792-03-4. doi: 10.24963/ijcai.2023/44. URL https://doi.org/10.24963/ijcai.2023/44.

[49] Mark S Manasse, Lyle A McGeoch, and Daniel D Sleator. Competitive algorithms for server problems. *Journal of Algorithms*, 11(2):208–230, 1990.

[50] Sharmila VS Duppala, Karthik A Sankararaman, and Pan Xu. Online minimum matching with uniform metric and random arrivals. *Operations Research Letters*, 50(1):45–49, 2022.

[51] Seyed A Esmaeili, Sharmila Duppala, John P Dickerson, and Brian Brubach. Fair labeled clustering. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 327–335, 2022.

[52] Pranjal Awasthi, Matthäus Kleindessner, and Jamie Morgenstern. Equalized odds postprocessing under imperfect group information. In *International conference on artificial intelligence and statistics*, pages 1770–1780. PMLR, 2020.

[53] Pranjal Awasthi, Alex Beutel, Matthäus Kleindessner, Jamie Morgenstern, and Xuezhi Wang. Evaluating fairness of machine learning models under uncertain and incomplete information. In *Conference on Fairness, Accountability, and Transparency (FAccT)*, pages 206–214, 2021.

[54] Serena Wang, Wenshuo Guo, Harikrishna Narasimhan, Andrew Cotter, Maya Gupta, and Michael Jordan. Robust optimization for fairness with noisy protected groups. *NeurIPS*, 33: 5190–5203, 2020.

[55] Tatsunori Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. Fairness without demographics in repeated loss minimization. In *ICML*, pages 1929–1938. PMLR, 2018.

[56] Nathan Kallus, Xiaojie Mao, and Angela Zhou. Assessing algorithmic fairness with unobserved protected class using data combination. *Management Science*, 68(3):1959–1981, 2022.

[57] Alex Lamy, Ziyuan Zhong, Aditya K Menon, and Nakul Verma. Noise-tolerant fair classification. *NeurIPS*, 32, 2019.

[58] Seyed Esmaeili, Brian Brubach, Leonidas Tsepenekas, and John Dickerson. Probabilistic fair clustering. *NeurIPS*, 33:12743–12755, 2020.

[59] Anshuman Chhabra, Peizhao Li, Prasant Mohapatra, and Hongfu Liu. Robust fair clustering: A novel fairness attack and defense framework. *International Conference on Learning Representations (ICLR)*, 2023.

[60] Sharmila Duppala, Juan Luque, John P. Dickerson, and Seyed A. Esmaeili. Robust fair clustering with group membership uncertainty sets. In *International Conference on Artificial Intelligence and Statistics*, 2024. URL `https://api.semanticscholar.org/CorpusID:270218660`.

[61] Gina Cook. *Woman Says Uber Driver Denied Her Ride Because of Her Wheelchair*. NBC4-Washington, 2018. Available at https://www.nbcwashington.com/news/local/woman-says-uber-driver-denied-her-ride-because-of-her-wheelchair/2029780/, Accessed: 2023-04-23.

[62] Gillan B. White. *Uber and Lyft Are Failing Black Riders*. The Atlantic, 2016. Available at https://www.theatlantic.com/business/archive/2016/10/uber-lyft-and-the-false-promise-of-fair-rides/506000/, Accessed: 2023-04-22.

[63] Eli Wirtschafter. *Driver discrimination still a problem as Uber and Lyft prepare to go public*. KALW, 2019. Available at https://www.kalw.org/post/driver-discrimination-still-problem-uber-and-lyft-prepare-go-public, Accessed: 2023-04-23.

[64] Alex Rosenblat, Karen EC Levy, Solon Barocas, and Tim Hwang. Discriminating tastes: Customer ratings as vehicles for bias. *Available at SSRN 2858946*, 2016.

[65] Hernan Galperin and Catrihel Greppi. Geographical discrimination in the gig economy. *Available at SSRN 2922874*, 2017.

[66] Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning: Limitations and Opportunities*. MIT Press, 2023.

[67] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *ITCS*, 2012.

[68] Nixie S Lesmana, Xuan Zhang, and Xiaohui Bei. Balancing efficiency and fairness in on-demand ridesourcing. In *NeurIPS*, 2019.

[69] Will Ma and Pan Xu. Group-level fairness maximization in online bipartite matching. In *AAMAS*, 2022.

[70] Yifan Xu and Pan Xu. Trade the system efficiency for the income equality of drivers in rideshare. In *IJCAI*, 2020.

[71] Tom Sühr, Asia J Biega, Meike Zehlike, Krishna P Gummadi, and Abhijnan Chakraborty. Two-sided fairness for repeated matchings in two-sided markets: A case study of a ride-hailing platform. In *KDD*, 2019.

[72] Jon Feldman, Aranyak Mehta, Vahab Mirrokni, and Shan Muthukrishnan. Online stochastic matching: Beating 1-1/e. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 117–126. IEEE, 2009.

[73] Nikhil Bansal, Anupam Gupta, Jian Li, Julián Mestre, Viswanath Nagarajan, and Atri Rudra. When lp is the cure for your matching woes: Improved bounds for stochastic matchings. In *European Symposium on Algorithms*, pages 218–229. Springer, 2010.

[74] Saeed Alaei, MohammadTaghi Hajiaghayi, and Vahid Liaghat. The online stochastic generalized assignment problem. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 11–25. Springer, 2013.

[75] Richard M Karp, Umesh V Vazirani, and Vijay V Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 352–358. ACM, 1990.

[76] Bahman Bahmani and Michael Kapralov. Improved bounds for online stochastic matching. In *European Symposium on Algorithms*, pages 170–181. Springer, 2010.

[77] Vahideh H Manshadi, Shayan Oveis Gharan, and Amin Saberi. Online stochastic matching: Online actions based on offline statistics. *Mathematics of Operations Research*, 37(4):559–573, 2012.

[78] Saeed Alaei, MohammadTaghi Hajiaghayi, and Vahid Liaghat. Online prophet-inequality matching with applications to ad allocation. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 18–35, 2012.

[79] Brian Brubach, Karthik Abinav Sankararaman, Aravind Srinivasan, and Pan Xu. New algorithms, better bounds, and a novel model for online stochastic matching. In *24th Annual European Symposium on Algorithms (ESA 2016)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

[80] John Dickerson, Karthik Sankararaman, Kanthi Sarpatwar, Aravind Srinivasan, Kung-Lu Wu, and Pan Xu. Online resource allocation with matching constraints. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2019.

[81] Marek Adamczyk, Fabrizio Grandoni, and Joydeep Mukherjee. Improved approximation algorithms for stochastic matching. In *Algorithms-ESA 2015*, pages 1–12. Springer, 2015.

[82] Brian Brubach, Karthik Abinav Sankararaman, Aravind Srinivasan, and Pan Xu. Online stochastic matching: New algorithms and bounds, 2016.

[83] Aravind Srinivasan. Distributions on level-sets with applications to approximation algorithms. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 588–597. IEEE, 2001.

[84] Michael Mitzenmacher and Eli Upfal. *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis*. Cambridge university press, 2017.

[85] Benjamin Barann, Daniel Beverungen, and Oliver Müller. An open-data approach for quantifying the potential of taxi ridesharing. *Decision Support Systems*, 99:86–95, 2017.

[86] Sebastijan Sekulić, Jed Long, and Urška Demšar. A spatially aware method for mapping movement-based and place-based regions from spatial flow networks. *Transactions in GIS*, 25(4):2104–2124, 2021.

[87] Javier Alonso-Mora, Alex Wallar, and Daniela Rus. Predictive routing for autonomous mobility-on-demand systems with ride-sharing. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3583–3590, 2017. doi: 10.1109/IROS.2017.8206203.

[88] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE transactions on pattern analysis and machine intelligence*, 24(4):509–522, 2002.

[89] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. Text matching as image recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[90] Bert Huang and Tony Jebara. Loopy belief propagation for bipartite maximum weight b-matching. In *Artificial Intelligence and Statistics*, pages 195–202. PMLR, 2007.

[91] Tony Jebara, Jun Wang, and Shih-Fu Chang. Graph construction and b-matching for semi-supervised learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 441–448, 2009.

[92] Bert Huang and Tony Jebara. Fast b-matching via sufficient selection belief propagation. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 361–369. JMLR Workshop and Conference Proceedings, 2011.

[93] Krzysztof M Choromanski, Tony Jebara, and Kui Tang. Adaptive anonymity via b-matching. *Advances in Neural Information Processing Systems*, 26, 2013.

[94] Mikhail Zaslavskiy, Francis Bach, and Jean-Philippe Vert. Global alignment of protein–protein interaction networks by graph matching methods. *Bioinformatics*, 25(12):i259–1267, 2009.

[95] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, page 259–268, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450336642. doi: 10.1145/2783258.2783311. URL https://doi.org/10.1145/2783258.2783311.

[96] Govind S Sankar, Anand Louis, Meghana Nasre, and Prajakta Nimbhorkar. Matchings with group fairness constraints: Online and offline algorithms. *arXiv preprint arXiv:2105.09522*, 2021.

[97] Matthäus Kleindessner, Samira Samadi, Pranjal Awasthi, and Jamie Morgenstern. Guarantees for spectral clustering with fairness constraints. In *International Conference on Machine Learning*, pages 3458–3467. PMLR, 2019.

[98] Paul Lewis and Paul Hilder. Leaked: Cambridge Analytica's blueprint for Trump victory. *The Guardian*, 2018. URL https://www.theguardian.com/uk-news/2018/mar/23/leaked-cambridge-analyticas-blueprint-for-trump-victory.

[99] Nicholas Confessore. Cambridge Analytica and Facebook: The Scandal and the Fallout So Far. *The New York Times*, 2018. URL https://www.nytimes.com/2018/04/04/us/politics/cambridge-analytica-scandal-fallout.html.

[100] David García-Soriano and Francesco Bonchi. Fair-by-design matching. *Data Mining and Knowledge Discovery*, 34:1291–1335, 2020.

[101] R. Impagliazzo and R. Paturi. Complexity of k-sat. In *Proceedings. Fourteenth Annual IEEE Conference on Computational Complexity (Formerly: Structure in Complexity Theory Conference) (Cat.No.99CB36317)*, pages 237–240, 1999. doi: 10.1109/CCC.1999.766282.

[102] Devdatt P. Dubhashi and Desh Ranjan. Balls and bins: A study in negative dependence. *BRICS Report Series*, 3(25), Jan. 1996. doi: 10.7146/brics.v3i25.20006. URL https://tidsskrift.dk/brics/article/view/20006.

[103] Rajiv Gandhi, Samir Khuller, Srinivasan Parthasarathy, and Aravind Srinivasan. Dependent rounding and its applications to approximation algorithms. *Journal of the ACM (JACM)*, 53(3): 324–360, 2006.

[104] Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. *SIAM Journal on Computing*, 43 (6):1831–1879, 2014.

[105] Moran Feldman, Ola Svensson, and Rico Zenklusen. Online contention resolution schemes with applications to bayesian selection problems. *SIAM Journal on Computing*, 50(2):255–300, 2021.

[106] Tomer Ezra, Michal Feldman, Nick Gravin, and Zhihao Gavin Tang. Prophet matching with general arrivals. *Mathematics of Operations Research*, 47(2):878–898, 2022.

[107] David A Freedman. On tail probabilities for martingales. *the Annals of Probability*, pages 100–118, 1975.

[108] Alessandro Panconesi and Aravind Srinivasan. Randomized distributed edge coloring via an extension of the chernoff–hoeffding bounds. *SIAM Journal on Computing*, 26(2):350–368, 1997.

[109] Tom Bohman and Peter Keevash. Dynamic concentration of the triangle-free process, 2019. URL `https://arxiv.org/abs/1302.5963`.

[110] David J Abraham, Avrim Blum, and Tuomas Sandholm. Clearing algorithms for barter exchange markets: Enabling nationwide kidney exchanges. In *ACM Conference on Electronic Commerce (EC)*, pages 295–304, 2007.

[111] Mingyu Xiao and Xuanbei Wang. Exact algorithms and complexity of kidney exchange. In *Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 555–561, 2018.

[112] Richard M Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Springer, 1972.

[113] Avrim Blum, John P Dickerson, Nika Haghtalab, Ariel D Procaccia, Tuomas Sandholm, and Ankit Sharma. Ignorance is almost bliss: Near-optimal stochastic matching with few queries. In *Conference on Economics and Computation (EC)*, pages 325–342, 2015.

[114] Abolfazl Asudeh, Tanya Berger-Wolf, Bhaskar DasGupta, and Anastasios Sidiropoulos. Maximizing coverage while ensuring fairness: A tale of conflicting objectives. *Algorithmica*, pages 1–45, 2022.

[115] Osbert Bastani, Xin Zhang, and Armando Solar-Lezama. Probabilistic verification of fairness properties via concentration. *Proceedings of the ACM on Programming Languages*, 3 (OOPSLA):1–27, 2019.

[116] Péter Biró, Bernadette Haase-Kromwijk, et al. Building kidney exchange programmes in Europe: an overview of exchange practice and activities. *Transpl.*, 103(7), 2019.

[117] Georg Anegg, Haris Angelidakis, and Rico Zenklusen. Simpler and stronger approaches for non-uniform hypergraph matching and the furedi, kahn, and seymour conjecture. In *Symposium on Simplicity in Algorithms (SOSA)*, pages 196–203. SIAM, 2021.

[118] Nikhil Bansal, Nitish Korula, Viswanath Nagarajan, and Aravind Srinivasan. On k-column sparse packing programs. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 369–382. Springer, 2010.

[119] Brian Brubach, Karthik A. Sankararaman, Aravind Srinivasan, and Pan Xu. Algorithms to approximate column-sparse packing problems. *ACM Trans. Algorithms*, 16(1), November 2019. ISSN 1549-6325. doi: 10.1145/3355400. URL `https://doi.org/10.1145/3355400`.

[120] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair clustering through fairlets. In *Neural Information Processing Systems (NeurIPS)*, 2017.

[121] Xiaohui Bei, Shengxin Liu, Chung Keung Poon, and Hongao Wang. Candidate selections with proportional fairness constraints. *Autonomous Agents and Multi-Agent Systems*, 36, 2022.

[122] Alvin E Roth. *Who gets what–and why: the new economics of matchmaking and market design*. Houghton Mifflin Harcourt, 2015.

[123] Zoltan Furedi, Jeff Kahn, and Paul D. Seymour. On the fractional matching polytope of a hypergraph. *Combinatorica*, 13(2):167–180, 1993.

[124] Svante Janson and Andrzej Ruciński. The infamous upper tail. *Random Structures & Algorithms*, 20(3):317–342, 2002.

[125] Will Ma. Improvements and generalizations of stochastic knapsack and multi-armed bandit approximation algorithms. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 1154–1163. SIAM, 2014.

[126] Jeong Han Kim and Van H Vu. Concentration of multivariate polynomials and its applications. *Combinatorica*, 20(3):417–434, 2000.

[127] Warren Schudy and Maxim Sviridenko. Bernstein-like concentration and moment inequalities for polynomials of independent random variables: multilinear case. *arXiv preprint arXiv:1109.5193*, 2011.

[128] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. In *The collected works of Wassily Hoeffding*, pages 409–426. Springer, 1994.

[129] Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013.

[130] Charles Harris, K Jarrod Millman, et al. Array programming with NumPy. *Nature*, 585(7825): 357–362, 2020.

[131] John P. Dickerson, Ariel D. Procaccia, and Tuomas Sandholm. Failure-aware kidney exchange. *Management Science*, 65(4):1768–1791, 2019.

[132] OPTN/UNOS. OPTN policy notice: Revising priority points in kidney paired donation program, 2019. Technical report.

[133] National Kidney Registry. National Kidney Registry quarterly report: Q1 2022. https://www.kidneyregistry.org/wp-content/uploads/2022/05/nkr-report-2022-Q1_v16.pdf, 2022. [Online; accessed 19 May 2022].

[134] Samir Khuller, Stephen G Mitchell, and Vijay V Vazirani. On-line algorithms for weighted matching and stable marriages. Technical report, Cornell University, 1990.

[135] Bala Kalyanasundaram and Kirk Pruhs. Online weighted matching. *Journal of Algorithms*, 14 (3):478–488, 1993.

[136] Adam Meyerson, Akash Nanavati, and Laura Poplawski. Randomized online algorithms for minimum metric bipartite matching. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 954–959. Society for Industrial and Applied Mathematics, 2006.

[137] Nikhil Bansal, Niv Buchbinder, Anupam Gupta, and Joseph Naor. A randomized o(log2 k)-competitive algorithm for metric bipartite matching. *Algorithmica*, 68(2):390–403, 2014. URL http://dblp.uni-trier.de/db/journals/algorithmica/algorithmica68.html#BansalBGN14.

[138] Martin Gairing and Max Klimm. Greedy metric minimum online matchings with random arrivals. *Operations Research Letters*, 47(2):88–91, 2019.

[139] Sharath Raghvendra. A robust and optimal online algorithm for minimum metric bipartite matching. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2016)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

[140] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. propublica. *See https://www. propublica. org/article/machine-bias-risk-assessments-in-criminal-sentencing*, 2016.

[141] Sara Ahmadian, Alessandro Epasto, Ravi Kumar, and Mohammad Mahdian. Clustering without over-representation. In *KDD*, 2019.

[142] Arturs Backurs, Piotr Indyk, Krzysztof Onak, Baruch Schieber, Ali Vakilian, and Tal Wagner. Scalable fair clustering. In *International conference on machine learning*, pages 405–413. PMLR, 2019.

[143] Lingxiao Huang, Shaofeng Jiang, and Nisheeth Vishnoi. Coresets for clustering with fairness constraints. In *NeurIPS*, 2019.

[144] Matthäus Kleindessner, Pranjal Awasthi, and Jamie Morgenstern. Fair k-center clustering for data summarization. In *International Conference on Machine Learning*, pages 3448–3457. PMLR, 2019.

[145] Ian Davidson and SS Ravi. Making existing clusterings fairer: Algorithms, complexity results and insights. In *AAAI*, 2020.

[146] Daqing Chen, Sai Laing Sain, and Kun Guo. Data mining for the online retail industry: A case study of rfm model-based customer segmentation using data mining. *Journal of Database Marketing & Customer Strategy Management*, 19(3):197–208, 2012.

[147] Charu Chandra Aggarwal, Joel Leonard Wolf, and Philip Shi-lung Yu. Method for targeted advertising on the web based on accumulated self-learning data, clustering users and semantic node graph techniques, March 30 2004. US Patent 6,714,975.

[148] Pang-Ning Tan, Michael Steinbach, DA Karpatne, and DV Kumar. Introduction to data mining , 2nd editio, 2018.

[149] Jiawei Han, Micheline Kamber, and Jian Pei. Data mining concepts and techniques third edition. *The Morgan Kaufmann Series in Data Management Systems*, 5(4):83–124, 2011.

[150] Ava Kofman and Ariana Tobin. Facebook ads can still discriminate against women and older workers, despite a civil rights settlement. *ProPublica. Retrieved December*, 12:2020, 2019.

[151] Till Speicher, Muhammad Ali, Giridhari Venkatadri, Filipe Nunes Ribeiro, George Arvanitakis, Fabrício Benevenuto, Krishna P Gummadi, Patrick Loiseau, and Alan Mislove. Potential for discrimination in online targeted advertising. In *Conference on Fairness, Accountability, and Transparency (FAccT)*, 2018.

[152] Amit Datta, Anupam Datta, Jael Makagon, Deirdre K Mulligan, and Michael Carl Tschantz. Discrimination in online advertising: A multidisciplinary inquiry. In *Conference on Fairness, Accountability, and Transparency (FAccT)*, 2018.

[153] P Deepak. Whither fair clustering. In *AI for Social Good Workshop. Harvard CRCS*, 2020.

[154] David B Shmoys, Chaitanya Swamy, and Retsef Levi. Facility location with service installation costs. In *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1088–1097, 2004.

[155] Dachuan Xu and Shuzhong Zhang. Approximation algorithm for facility location with service installation costs. *Operations Research Letters*, 36(1):46–50, 2008.

[156] United States Senate. S. 1745 – 102nd Congress: Civil Rights Act of 199, 1991. `https://www.govtrack.us/congress/bills/102/s1745`.

[157] Seyed A Esmaeili, Brian Brubach, Aravind Srinivasan, and John P Dickerson. Fair clustering under a bounded cost. *arXiv preprint arXiv:2106.07239*, 2021.

[158] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *ICML*, 2013.

[159] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *KDD*, 2015.

[160] David Harris, Shi Li, Aravind Srinivasan, Khoa Trinh, and Thomas Pensyl. Approximation algorithms for stochastic clustering. In *NeurIPS*, 2018.

[161] Brian Brubach, Darshan Chakrabarti, John Dickerson, Samir Khuller, Aravind Srinivasan, and Leonidas Tsepenekas. A pairwise fair and community-preserving approach to k-center clustering. In *International conference on machine learning*, pages 1178–1189. PMLR, 2020.

[162] Mohsen Abbasi, Aditya Bhaskara, and Suresh Venkatasubramanian. Fair clustering via equitable group representations, 2020.

[163] Mehrdad Ghadiri, Samira Samadi, and Santosh Vempala. Socially fair k-means clustering. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 438–448, 2021.

[164] Xiaohui Bei, Shengxin Liu, Chung Keung Poon, and Hongao Wang. Candidate selections with proportional fairness constraints. In *AAMAS*, 2020.

[165] Michael R Garey and David S Johnson. *Computers and intractability*, volume 174. freeman San Francisco, 1979.

[166] Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized algorithms*, volume 5. Springer, 2015.

[167] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[168] Arthur Asuncion, David Newman, et al. Uci machine learning repository, 2007.

[169] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006.

[170] Jack Nicholls, Aditya Kuppa, and Nhien-An Le-Khac. Financial cybercrime: A comprehensive survey of deep learning approaches to tackle the evolving financial crime landscape. *IEEE Access*, 9:163965–163986, 2021.

[171] Sanjay Kumar, Rafeeq Ahmed, Salil Bharany, Mohammed Shuaib, Tauseef Ahmad, Elsayed Tag Eldin, Ateeq Ur Rehman, and Muhammad Shafiq. Exploitation of machine learning algorithms for detecting financial crimes based on customers' behavior. *Sustainability*, 14(21): 13875, 2022.

[172] Mohammad Ahmad Sheikh, Amit Kumar Goel, and Tapas Kumar. An approach for prediction of loan approval using machine learning algorithm. In *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pages 490–494. IEEE, 2020.

[173] Kumar Arun, Garg Ishan, and Kaur Sanmeet. Loan approval prediction based on machine learning approach. *IOSR Journal of Computer Engineering*, 18(3):18–21, 2016.

[174] Ali A Mahmoud, Tahani AL Shawabkeh, Walid A Salameh, and Ibrahim Al Amro. Performance predicting in hiring process and performance appraisals using machine learning. In *International Conference on Information and Communication Systems (ICICS)*, pages 110–115. IEEE, 2019.

[175] Elmira Van den Broek, Anastasia Sergeeva, and Marleen Huysman. When the machine meets the expert: An ethnography of developing AI for hiring. *MIS Quarterly*, 45(3), 2021.

[176] Guido Vittorio Travaini, Federico Pacchioni, Silvia Bellumore, Marta Bosia, and Francesco De Micco. Machine learning and criminal justice: A systematic review of advanced methodology for recidivism risk prediction. *International Journal of Environmental Research and Public Health*, 19(17):10594, 2022.

[177] Mehdi Ghasemi, Daniel Anvari, Mahshid Atapour, J Stephen Wormith, Keira C Stockdale, and Raymond J Spiteri. The application of machine learning to a general risk–need assessment instrument in the prediction of criminal recidivism. *Criminal Justice and Behavior*, 48(4):518–538, 2021.

[178] David Danks and Alex John London. Algorithmic bias in autonomous systems. In *Int. Joint Conf. on Artificial Intelligence (IJCAI)*, volume 17, pages 4691–4697, 2017.

[179] Trishan Panch, Heather Mattie, and Rifat Atun. Artificial intelligence and algorithmic bias: implications for health systems. *Journal of Global Health*, 9(2), 2019.

[180] Pranjal Awasthi, Brian Brubach, Deeparnab Chakrabarty, John P Dickerson, Seyed A. Esmaeili, Matthäus Kleindessner, Marina Knittel, Jamie Morgenstern, Samira Samadi, Aravind Srinivasan, and Leonidas Tsepenekas. Fairness in clustering. In *AAAI*, 2022.

[181] Sara Ahmadian, Alessandro Epasto, Marina Knittel, Ravi Kumar, Mohammad Mahdian, Benjamin Moseley, Philip Pham, Sergei Vassilvitskii, and Yuyan Wang. Fair hierarchical clustering. *Advances in Neural Information Processing Systems*, 33:21050–21060, 2020.

[182] Marina Knittel, Max Springer, John P Dickerson, and MohammadTaghi Hajiaghayi. Generalized reductions: making any hierarchical clustering fair and balanced with low cost. In *ICML*, pages 17218–17242. PMLR, 2023.

[183] Marina Knittel, Max Springer, John Dickerson, and MohammadTaghi Hajiaghayi. Fair polylog-approximate low-cost hierarchical clustering. *NeurIPS*, 2023.

[184] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *NeurIPS*, 29, 2016.

[185] Ivar Krumpal. Determinants of social desirability bias in sensitive surveys: a literature review. *Quality & quantity*, 47(4):2025–2047, 2013.

[186] Anay Mehrotra and Nisheeth Vishnoi. Fair ranking with noisy protected attributes. *Advances in Neural Information Processing Systems*, 35:31711–31725, 2022.

[187] Anay Mehrotra and Nisheeth K. Vishnoi. Fair ranking with noisy protected attributes. *NeurIPS*, 2022.

[188] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust Optimization*, volume 28. Princeton University Press, 2009.

[189] Stefan Nickel, Claudius Steinhardt, Hans Schlenker, and Wolfgang Burkart. Ibm ilog cplex optimization studio—a primer. In *Decision optimization with IBM ILOG CPLEX Optimization Studio: A hands-on introduction to modeling with the Optimization Programming Language (OPL)*, pages 9–21. Springer, 2022.

[190] Aric Hagberg, Dan Schult, Pieter Swart, D Conway, L Séguin-Charbonneau, C Ellison, B Edwards, and J Torrents. Networkx. *URL http://networkx. github. io/index. html*, 2013.

[191] Marek Adamczyk and Michał Włodarczyk. Random order contention resolution schemes. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 790–801. IEEE, 2018.